Normalizing Flows for Knockoff-free Controlled Feature Selection

Derek Hansen
Department of Statistics
University of Michigan
dereklh@umich.edu

Brian Manzo
Department of Statistics
University of Michigan
bmanzo@umich.edu

Jeffrey Regier
Department of Statistics
University of Michigan
regier@umich.edu

Abstract

Controlled feature selection aims to discover the features a response depends on while limiting the false discovery rate (FDR) to a predefined level. Recently, multiple deep-learning-based methods have been proposed to perform controlled feature selection through the Model-X knockoff framework. We demonstrate, however, that these methods often fail to control the FDR for two reasons. First, these methods often learn inaccurate models of features. Second, the "swap" property, which is required for knockoffs to be valid, is often not well enforced. We propose a new procedure called FLOWSELECT to perform controlled feature selection that does not suffer from either of these two problems. To more accurately model the features, FLOWSELECT uses normalizing flows, the state-of-the-art method for density estimation. Instead of enforcing the "swap" property, FLOWSELECT uses a novel MCMC-based procedure to calculate p-values for each feature directly. Asymptotically, FLOWSELECT computes valid p-values. Empirically, FLOWSELECT consistently controls the FDR on both synthetic and semi-synthetic benchmarks, whereas competing knockoff-based approaches do not. FLOWSELECT also demonstrates greater power on these benchmarks. Additionally, FLOWSELECT correctly infers the genetic variants associated with specific soybean traits from GWAS data.

1 Introduction

Researchers in machine learning have made much progress in developing regression and classification models that can predict a response based on features. In many application areas, however, practitioners need to know *which* features drive variation in the response, and they need to do so in a way that limits the number of false discoveries. For example, in genome-wide association studies (GWAS), scientists must consider hundreds of thousands of genetic markers to identify variants associated with a particular trait or disease. The cost of false discoveries (i.e., selecting variants that are not associated with the disease) is high, as a costly follow-up experiment is often conducted for each selected variant. Another example where controlled feature selection matters is analyzing observational data about the effectiveness of educational interventions. In this case, researchers may want to select certain educational programs to implement on a larger scale and require confidence that their selection does not include unacceptably many ineffective programs. As a result, researchers are interested in methods that model the dependence structure of the data while providing an upper bound on the false discovery rate (FDR).

Model-X knockoffs (Candès et al., 2018) is a popular method for controlled variable selection, offering theoretical guarantees of FDR control and the flexibility to use arbitrary predictive models. However, even with knowledge of the underlying feature distribution, the Model-X knockoffs method is not feasible unless the feature distribution is either a finite mixture of Gaussians (Gimenez et al., 2019) or has a known Markov structure (Bates et al., 2020). Hence, a body of research explores the use of empirical approaches that use deep generative models to estimate the distribution of X and

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

sample knockoff features (Jordon et al., 2019; Liu & Zheng, 2018; Romano et al., 2020; Sudarshan et al., 2020).

The ability of these methods to control the FDR is contingent on their ability to correctly model the distribution of the features. By itself, learning a sufficiently expressive feature model can be challenging. However, the knockoff procedure requires learning a knockoff distribution that satisfies the *swap property*, which is a much stronger requirement. Formally, let $X \in \mathbb{R}^D$ be a sample from the feature distribution and $\tilde{X} \in \mathbb{R}^D$ be a sample from the knockoff distribution conditioned on X. The swap property stipulates that the joint distribution $(X, \tilde{X}) \in \mathbb{R}^{2D}$ must be invariant to swapping the positions of any subset of features $S \in \{1, \ldots, D\}$:

$$(X, \tilde{X})_{\text{swap}(S)} \stackrel{D}{=} (X, \tilde{X}) \tag{1}$$

Here, swap(S) means exchanging the positions of X_j and $\tilde{X}_j$ for all $j \in S$. For example, in the case $D = 3$ and $S = \{1, 3\}$, the joint distribution is $(X, \tilde{X}) = (X_1, X_2, X_3, \tilde{X}_1, \tilde{X}_2, \tilde{X}_3)$, and the swapped joint distribution is $(X, \tilde{X})_{\text{swap}(S)} = (\tilde{X}_1, X_2, \tilde{X}_3, X_1, \tilde{X}_2, X_3)$. Note that, for $S = \{1, \ldots, D\}$, the swap property implies that $\tilde{X} \stackrel{D}{=} X$. See Candès et al. (2018) for a more detailed description of the swap property.

Even if a distribution were found satisfying the swap property, it may not provide enough power to make discoveries. For example, both properties are trivially satisfied by constructing exact copies of the features as knockoffs, but the resulting procedure has no power.

In situations where a valid knockoff distribution is available to sample from, knockoffs are computationally appealing because they require only one sample from a knockoff distribution to assess the relevance of all p features. However, in situations where the joint density of the features is unknown, we show that empirical approaches to knockoff generation (Jordon et al., 2019; Liu & Zheng, 2018; Romano et al., 2020; Sudarshan et al., 2020) fail to characterize a valid knockoff distribution and therefore do not control the FDR. We further show that even with a known covariate model, it is not straightforward to construct a valid knockoff distribution unless a specific model structure is known.

We propose a new feature selection method called FLOWSELECT (Section 3), which does not suffer from these problems. FLOWSELECT uses normalizing flows to learn the joint density of the covariates. Normalizing flows is a state-of-the-art method for density estimation; asymptotically, it can approximate any distribution arbitrarily well (Papamakarios et al., 2021; Kobyzev et al., 2020; Huang et al., 2018). Additionally, FLOWSELECT circumvents the need to sample a knockoff distribution by instead applying a fast variant of the conditional randomization test (CRT) introduced in Candès et al. (2018). Samples from the complete conditionals are drawn using MCMC, ensuring they are unbiased with respect to the learned data distribution.

Asymptotically, FLOWSELECT computes correct p-values to use for feature selection (Section 4). Our proof assumes the universal approximation property of normalizing flows and the convergence of MCMC samples to the Markov chain's stationary distribution. Under the same assumptions as the CRT, which includes a multiple-testing correction as in Benjamini & Hochberg (1995), a selection threshold can be picked which controls the FDR at a pre-defined level. Empirically, on both synthetic (Gaussian) data and semi-synthetic data (real predictors and a synthetic response), FLOWSELECT controls the FDR where other deep-learning-based knockoff methods do not. In cases in which competing methods do control the FDR, FLOWSELECT shows higher power (Section 5). Finally, in a challenging real-world problem with soybean genome-wide association study (GWAS) data, FLOWSELECT successfully harnesses normalizing flows for modeling discrete and sequential GWAS data, and for selecting genetic variants the traits depend on (Section 5.4).

2 Background

FLOWSELECT brings together four existing lines of research, which we briefly introduce below.

Normalizing flows Normalizing flows is a general framework for density estimation of a multi-dimensional distribution with arbitrary dependencies (Papamakarios et al., 2021). A normalizing flow starts with a simple probability distribution (e.g., Gaussian or uniform), which is called the *base distribution* and denoted Z, and transforms samples from this base distribution through a

series of invertible and differentiable transformations, denoted G, to define the joint distribution of $X \in \mathbb{R}^D \sim \mathcal{P}_X$. A normalizing flow with enough transformations can approximate any multivariate density, subject to regularity conditions detailed by Kobyzev et al. (2020). Compared to other density-estimation methods, normalizing flows are computationally efficient. Details about the specific normalizing flow architecture used in FLOWSELECT are provided in Appendix A.

Controlled feature selection Consider a response Y which depends on a vector of features $X \in \mathbb{R}^D$. Depending on how the features are chosen, it is plausible that only a subset of the features contains all relevant information about Y. Specifically, conditioned on the relevant features in X, Y is independent of the remaining features in X (i.e. the null features). The goal of the controlled feature selection procedure is to maximize the number of relevant features selected while limiting the number of null features selected to a predefined level. If we denote the total number of selected features R, then we can decompose R into V, the number of relevant features selected, and S, the number of null features selected.

Conditional randomization test Controlled feature selection can be seen as a multiple hypothesis testing problem where there are p null hypotheses, each of which says that feature X_j is conditionally independent of the response Y given all the other features X_{-j}. Explicitly, the test of the following hypothesis is conducted for each feature $j = \{1, \ldots, D\}$:

$$H_0 : X_j \perp Y | X_{-j} \quad \text{versus} \quad H_1 : X_j \not\perp Y | X_{-j}. \tag{2}$$

To test these hypotheses, one can use a conditional randomization test (CRT) (Candès et al., 2018). For each feature tested in a conditional randomization test, a test statistic T_j (e.g., the LASSO coefficient or another measure of feature importance) is first computed on the data. Then, the null distribution of T_j is estimated by computing its value $\tilde{T}_j$ based on samples $\tilde{X}_j$ drawn from the conditional distribution of X_j given X_{-j}. Finally, the p-value is calculated based on the empirical CDF of the null test statistics, and features whose p-values fall below the threshold set by the Benjamini-Hochberg procedure (Benjamini & Hochberg, 1995) are selected. Though the CRT is introduced as a computationally inefficient alternative to knockoffs, the CRT nonetheless has appeal because it requires only knowledge of the feature distribution, which can be learned empirically by maximum likelihood.

Holdout randomization test The holdout randomization test (HRT) (Tansey et al., 2021) is a fast variant of the CRT; it uses a test statistic that requires fitting the model only once. Let θ represent the parameters of the chosen model, and let $T(X, Y, \theta)$ be an importance statistic calculated from the model with input data. For example, T, could be the predictive likelihood $\mathcal{P}_\theta(Y^{\text{test}}|X^{\text{test}})$ or the predictive score R^2. To use the HRT, first fit model parameters $\hat{\theta}$ based on the training data. Next, for each covariate j, calculate the test statistic $T_j^* \leftarrow T(X^{\text{test}}, Y^{\text{test}}, \hat{\theta})$. Then, generate k null samples and compute $T_{j,k} \leftarrow T(X^{\text{test}}_{(j \leftarrow j_k)}, Y^{\text{test}}, \hat{\theta})$, where $X^{\text{test}}_{(j \leftarrow j_k)}$ replaces the j-th covariate with the k-th generated null sample. Finally, calculate the p-value as in the CRT, based on the empirical CDF of the null test statistics.

3 Methodology

FLOWSELECT implements the CRT for arbitrary feature distributions by using a normalizing flow to fit the feature distribution and Markov chain Monte Carlo (MCMC) to sample from each complete conditional distribution. Performing controlled feature selection with FLOWSELECT consists of the three steps below.

Step 1: Model the predictors with a normalizing flow

Starting with the observed samples of the features $X_1, \ldots, X_N \sim \mathcal{P}_X$, we fit the parameters of a normalizing flow G_θ to maximize the log likelihood of the data with respect to a base distribution p_Z:

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{N} \log p_\theta(X_i)$$
$$\text{where } p_\theta(X_i) = p_Z(G_\theta(X)) \left| \det \left(\frac{\partial G_\theta(X)}{\partial X} \right) \right|. \tag{3}$$

The resulting density $p_{\hat{\theta}}$ is a fitted approximation to the true density $\mathcal{P}_X$. The specific normalizing flow architecture we use in our first two experiments consists of a single Gaussianization layer (Meng et al., 2020) followed by a masked autoregressive flow (MAF) (Papamakarios et al., 2017). The first layer can learn complex marginal distributions for each covariate, while the MAF learns the dependencies between them. More detail on normalizing flows and on this particular architecture can be found in Appendix A.

Step 2: Sample from the complete conditionals with MCMC

For each feature j, we aim to sample corresponding null features $\tilde{X}_{i,j,k}$ for all $k \in \{1, \ldots, K\}$ that are equal in distribution to $p_{\hat{\theta}}(X_{i,j}|X_{i,-j})$, but independent of Y_i. However, directly sampling from this conditional distribution is intractable. Instead, we implement an MCMC algorithm that admits it as a stationary distribution. The samples drawn from MCMC are autocorrelated, but any statistic calculated over these samples will converge almost surely to the correct value. The choice of the MCMC proposal distribution q_j is flexible. Because each Markov chain is only one-dimensional, a Metropolis-Hastings Gaussian random walk with the standard deviation set based on the covariance can be expected to mix rapidly. Alternatively, information from $p_{\hat{\theta}}$, such as higher-order derivatives, could be used to construct a more efficient proposal. Algorithm 1 details how to implement step 2.

Step 3: Test for significance with the HRT

As in the CRT, feature j has high evidence of being significant if, under the assumption that j is a null feature, the probability of realizing a test statistic greater than the observed $T_j(X)$ is low. Formally, letting $[\tilde{X}_j, X_{-j}]$ be the observed feature matrix with the observed feature X_j swapped out with the null feature $\tilde{X}_j$, we can write this as a p-value α_j:

$$\alpha_j \equiv \mathcal{P}_{\tilde{X}_j|X_{-j}} \left(T_j(X) < T_j([\tilde{X}_j, X_{-j}]) \right). \tag{4}$$

However, the above p-value α_j is not tractable. For each sample $\tilde{X}_{\cdot,j,k}$ drawn using MCMC, we calculate the corresponding feature statistic and compare it to the real feature statistic, leading to an

Algorithm 1 Step 2 of the FLOWSELECT procedure for drawing K null features $\tilde{X}_{i,j}|X_{i,-j}$ for feature j at observation i.

Input: Feature matrix $X \in \mathbb{R}^{N \times D}$, observation index i, feature index j, number of samples K, fitted normalizing flow $p_{\hat{\theta}}$, MCMC proposal q_j
Output: Null features $\tilde{X}_{i,j,k}$ for $k = 1, \ldots, K$
for $k = 1, \ldots, K$ **do**
  Propose: $X^\star_{i,j,k} \sim q_j(\cdot|\tilde{X}_{i,j,k-1}, X_{i,-j})$
  $r_{i,j,k} \leftarrow \frac{p_{\hat{\theta}}(X^\star_{i,j,k}, X_{i,-j}) q_j(\tilde{X}_{i,j,k-1}|X^\star_{i,j,k}, X_{i,-j})}{p_{\hat{\theta}}(\tilde{X}_{i,j,k-1}, X_{i,-j}) q_j(X^\star_{i,j,k}|\tilde{X}_{i,j,k-1}, X_{i,-j})}$
  Sample: $U_{i,j,k} \sim \text{Bernoulli}(r_{i,j,k} \wedge 1)$
  if $U_{i,j,k} = 1$ **then**
    $\tilde{X}_{i,j,k} \leftarrow X^\star_{i,j,k}$
  else
    $\tilde{X}_{i,j,k} \leftarrow \tilde{X}_{i,j,k-1}$
  end if
end for

approximated p-value $\hat{\alpha}_j$:

$$\hat{\alpha}_j \equiv \frac{1}{K+1}(1 + \sum_{k=1}^{K} \mathbf{1}[T_j(X) < T_j([\tilde{X}_{j,k}, X_{-j}])). \tag{5}$$

To control the FDR, we use the Benjamini-Hochberg procedure to establish a threshold for the observed p-values. Specifically, we set the threshold to $s(\gamma) \triangleq \max_j\{\hat{\alpha}_j : \hat{\alpha}_j \leq \frac{j}{D}\gamma\}$, and select all features j such that $\alpha_j \leq s(\gamma)$.

The Benjamini-Hochberg correction only guarantees FDR control provided that the p-values have either positive or zero correlation. Thus, the FDR control of FLOWSELECT depends on these assumptions being met. A more conservative correction from Benjamini & Yekutieli (2001) allows for arbitrary dependencies in p-values, but it suffers from low power. The Benjamini-Hochberg correction is widely used and empirically robust (Tansey et al., 2021), so we report results using it. Across our synthetic and semi-synthetic benchmarks in Section 5, we also find that FLOWSELECT maintains empirical FDR control.

Provided that the Benjamini-Hochberg assumptions are met, the FDR will be controlled, but the power of the test depends on T_j being higher when j is a significant feature. For example, if Y is expected to vary approximately linearly with respect to X, $T_j(X)$ could be the absolute estimated regression coefficient $|\hat{\beta}_j|$ for the linear model $Y = X\beta + \epsilon$. Another choice is the HRT feature statistic described earlier.

4 Asymptotic results

The ability of FLOWSELECT to control the FDR relies on its ability to produce estimated p-values that converge to the correct p-values for the hypothesis test in Equation (2).

Theorem 1. *Let $X \in \mathbb{R}_{N\times D}$ be a random feature matrix, where each row $X_{i,\cdot}$ is independent and identically distributed; $x \in \mathbb{R}_{N\times D}$ be the observed feature matrix; and α_j be the p-value as defined in Equation* (4) *with test statistic $T_j(X)$. Suppose there exists a sequence of functions $(G^n)_{n=1}^{\infty}$ and a base random variable Z satisfying the following conditions:*

1. *Each G^n is continuously differentiable and invertible.*
2. *$G^n \to G$ pointwise for some map G that is triangular, increasing, continuously differentiable, and satisfies $G(X_{i,\cdot}) \overset{D}{=} Z$.*

For $n = 1, 2, \ldots,$ let X^n be the random feature matrix where each row i is independent and has distribution $X^n_{i,\cdot} = (G^n)^{-1}(Z)$. Then, the p-value in Equation (5) *calculated using K MCMC samples targeting $X^n_{\cdot,j} \mid X^n_{\cdot,-j} = x_{\cdot,-j}$ converges to the correct p-value α_j with probability* 1.

Here we sketch the proof. A full proof can be found in Appendix B. First, by construction each G^n defines a distribution $X^n_{i,\cdot} \overset{D}{=} (G^n)^{-1}(Z)$ that in turn implies a conditional distribution $X^n_{\cdot,j}|X^n_{\cdot,-j} = x_{\cdot,-j}$. We show these conditional distributions converge to the true conditional distribution of $X_{\cdot,j}$ given $X_{\cdot,-j} = x_{\cdot,-j}$. Consequently, the probability of observing a higher test statistic under the approximated null distribution $\tilde{X}^n_{\cdot,j} \overset{D}{=} X^n_{\cdot,j}$, written α^n_j, will converge to the probability under the true null distribution $\tilde{X}_{\cdot,j}|X_{\cdot,-j} = x_{\cdot,-j}$, i.e. α_j. Next, the Cesaro average of K samples from an MCMC algorithm targeting $\tilde{X}^n_{\cdot,j}|X_{\cdot,-j} = x_{\cdot,-j}$, written $\hat{\alpha}_{j,K,n}$ will converge to α^n_j with probability 1 as $K \to \infty$. Combining these two convergences leads to the stated result.

Assuming the limiting p-values $\{\alpha_j\}$ satisfy the chosen multiple-hypothesis-testing assumptions, Theorem 1 specifies additional conditions that are sufficient for FDR control. These conditions are not strictly fewer than those required for empirical model-X knockoff-based methods to control FDR, but they may be easier to satisfy adequately in practice. For example, the condition that there exists a sequence $(G_n)_{n=1}^{\infty}$ converging to the true mapping G is satisfied asymptotically by many flow architectures that are universal distribution approximators, including the Gaussianization Flows and Masked Autoregressive Flows used in our experiments (Huang et al., 2018; Meng et al., 2020; Kobyzev et al., 2020). In practice, it is unlikely that an exact mapping G will be learned, as doing so

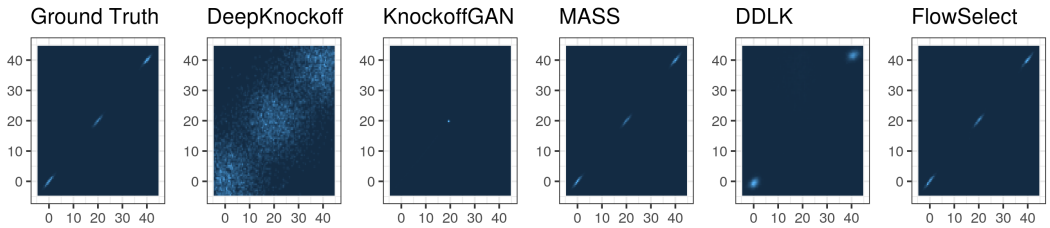

Figure 1: A density plot of the feature distribution with coordinate $j = 1$ on the x-axis and coordinate $j = 2$ on the y-axis. The ground truth density is compared to the normalizing flow fitted within FLOWSELECT and the distribution of each knockoff method (DeepKnockoff, KnockoffGAN, MASS, and DDLK). To have FDR control, each distribution should match the distribution of the features.

could require infinite training data, infinitely deep transformations, and exact nonconvex optimization. Nonetheless, normalizing flows work extremely well in practice; Theorem 1 gives intuition for the good performance of FLOWSELECT that we observe empirically.

5 Experiments

5.1 Synthetic experiment with a mixture of highly correlated Gaussians

We compare FLOWSELECT to the aforementioned knockoff methods with synthetic data drawn from a mixture of three highly correlated Gaussian distributions with dimension $D = 100$.[1] For each knockoff method, we use the exact implementation described in their respective papers, and we utilize the code made publicly available by the authors (c.f. Appendix D.3 for further details). For further comparison, we also implement the MASS knockoff procedure from Gimenez et al. (2019) and the RANK knockoff procedure from Fan et al. (2020). These methods estimate the unknown feature distribution using either a mixture of Gaussians (MASS) or a sparse precision matrix (RANK), and then sample the knockoffs directly as in Candès et al. (2018).

To generate the data, we draw $N = 100,000$ highly correlated samples. For $i = 1, \ldots, N$, we sample

$$X_i \overset{\text{i.i.d}}{\sim} \sum\nolimits_{m=1}^{3} \pi_m p_{\mathcal{N}}(X_i; \mu_m, \Sigma_m), \tag{6}$$

with mixing weights $\pi = (0.371, 0.258, 0.371)$, mean vector $\mu = (0, 20, 40)$, and covariance matrices Σ_m. Each covariance Σ_m follows an AR(1) pattern such that $(\Sigma_m)_{i,j} = \rho_m^{|i-j|}$ where $\rho = (0.982, 0.976, 0.970)$. The response Y_i is linear in $f_i(X_i)$ for some function f_i and coefficient vector β i.e., $Y_i = f_i(X_i)\beta + \epsilon_i$. Each coefficient β_j equals $\frac{100}{\sqrt{N}} B_j$, where $B_j = 0$ with probability 0.8, $B_j = 1$ with probability 0.1, and $B_j = -1$ with probability 0.1. We consider two different schemes for the f_i that connect the features to the response. In our linear setting, f_i is equal to the identity function. In our nonlinear setting, $f_i(x)$ is set equal to $\sin(5x)$ for odd i and $f_i(x) = \cos(5x)$ for even i.

The experimental setting we have described so far is adapted from Sudarshan et al. (2020). However, we found that the $N = 2000$ they used was too few observations for any of the methods to do well in a general non-linear setting. Moreover, in many situations where controlled feature selection is deployed, neighboring features will be highly correlated. To reflect this, we also increased the base correlation between features within each mixture to create a more challenging example. We show results under the original settings of Sudarshan et al. (2020) in Appendix K.

For each model, we use 90% of the data for training to generate null features and the remaining 10% for calculating the feature statistics. To define the feature statistics, we use the holdout randomization test (HRT) described at the end of Section 2. For the HRT, we employ different predictive models for each response type ("linear" and "nonlinear"). Specifically, for the linear response, we use the predictive log-likelihood from the LASSO (Tibshirani, 1996), and for the nonlinear response, we use the predictive negative mean-squared error from a random forest regressor (Breiman, 2001).

[1] Software to reproduce our experiments is available at https://github.com/dereklhansen/flowselect.

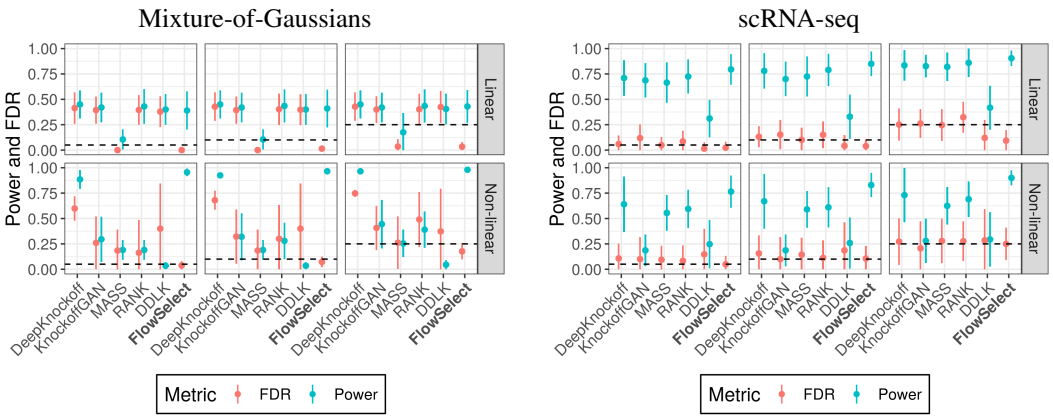

Figure 2: Comparison of power and false discovery rate (FDR) control of FLOWSELECT to knockoff methods on the Mixture-of-Gaussians dataset (left) and the scRNA-seq dataset (right) at targeted FDRs of 0.05, 0.1, and 0.25 (indicated by the dashed lines). Each point indicates the mean power and FDR across 20 replications and the error bars span one standard deviation either direction. In the top row, the response depends linearly on the features, and the feature statistics are calculated using the HRT with the LASSO. In the bottom row, the response depends non-linearly on the features, and the feature statistics are calculated using the HRT with random forest regression.

First, we look at how each procedure models the covariate distribution in Figure 1. In order to be valid knockoffs, the distribution of two knockoff features needs to be equal to that of the covariates. In this challenging example, each of the empirical knockoff methods fails to match the ground truth. In particular, DDLK and DeepKnockoffs are over-dispersed, while KnockoffGAN suffers from mode collapse. These findings for DeepKnockoffs and KnockoffGAN are similar to those reported by Sudarshan et al. (2020). Other than MASS, which directly fits a mixture of Gaussians, FLOWSELECT is the only method that matches the basic structure of the ground truth.

Figure 2 shows that the empirical knockoff procedures fail to control the FDR for both linear and nonlinear responses. One explanation for this lack of FDR control is the inability of the deep-learning-based methods to accurately model a knockoff distribution (c.f., Figure 1). As a result, the assumptions for the knockoff procedure will not hold, and FDR control is not guaranteed.

The effects of misspecification are clearly visible in the case of RANK, which approximates the mixture-of-Gaussians data with a multivariate Gaussian. However, even MASS, when given access to the correct data distribution, does not achieve across-the-board FDR control. This highlights the potential sensitivity of knockoffs to parameter misfit even when the underlying distributional family of the features is known. This is confirmed by the fact that, when provided with the true parameters, the oracle Model-X maintains FDR control, though with significantly less power than FLOWSELECT. (c.f. Appendix H).

5.2 Semi-synthetic experiment with scRNA-seq data

In this experiment, we use single-cell RNA sequencing (scRNA-seq) data from 10x Genomics (10x Genomics, 2017). Each variable $X_{n,g}$ is the observed gene expression of gene g in cell n. These data provide an experimental setting that is both realistic and, because gene expressions are often highly correlated, challenging. More background information about scRNA-seq data can be found in Agarwal et al. (2020).

We normalize the gene expression measurements to lie in $[0, 1]$, and we add a small amount of Gaussian noise so that the data is not zero-inflated. As in the semi-synthetic experiment from Sudarshan et al. (2020), we pick the 100 most correlated genes to provide a challenging, yet realistic example. We simulate responses that are both linear and nonlinear in the features. Figure 2 shows that FLOWSELECT maintains FDR control across multiple FDR target levels, feature statistics, and generated responses. In cases in which the knockoff methods control FDR successfully, FLOWSELECT has higher power in discovering the features the response depends on.

An advantage of knockoffs over CRT-based methods like FLOWSELECT is that the predictive model only needs to be evaluated once. Hence, while FLOWSELECT has a faster runtime than DDLK for this experiment, it is slower than DeepKnockoff and KnockoffGAN. However, Figure 2 shows that these two models fail to reliably control FDR and have much less power than FLOWSELECT; it is not clear how additional computational resources could be leveraged to improve the performance of these competing methods. A full table of runtimes on the scRNA-seq dataset can be found in Appendix F.

The need to compute a different predictive model for each feature within the CRT is mitigated by using efficient feature statistics such as the HRT (Tansey et al., 2021) and the distilled CRT (Liu et al., 2020). These methods fit a larger predictive model once, then evaluate either the residuals or test mean-squared-error for each feature individually. Moreover, the ability to scale to large feature dimensions D is more limited by fitting the feature distribution than computational burden, a trait shared by both knockoff- and CRT-based methods.

FLOWSELECT provides asymptotic guarantees of FDR control assuming sufficient MCMC samples have been drawn for the p-values to converge. In this experiment, the consequence of terminating MCMC sampling before convergence is low power, rather than loss of FDR control (see Figure 7 in Appendix J). Even for small numbers of MCMC samples, the FDR stabilizes below the target rate, while the power steadily increases with the number of samples. Because the MCMC run is initialized at the true features, we speculate that the sampled features will be highly correlated with the true features in the beginning of the run, making it harder to reject the null hypothesis that a feature is unimportant.

5.3 Ablation Study

FLOWSELECT differs from the competing knockoff-based approaches in two ways: using normalizing flows with MCMC to model the feature distribution for sampling null features and using the CRT for feature selection. To illustrate the impact of each of these components separately, we compare to the procedure used in Tansey et al. (2021), which uses mixture density networks (MDNs) to model the complete conditional distribution of each feature $\mathcal{P}(X_j|X_{-j})$ separately. They then sample null features from these learned distributions directly and use the HRT for feature selection. Since both FLOWSELECT and this procedure utilize the HRT, this allows us to evaluate whether the performance improvement of FLOWSELECT over empirical knockoffs is solely due to use of the HRT.

We compare the MDN-based approach to FLOWSELECT on the mixture-of-Gaussians (Section 5.1) and scRNA-seq (Section 5.2) datasets. A plot of this comparison can be found in Appendix G. While the MDN-based approach was able to match the performance of FLOWSELECT on the scRNA-seq dataset, it failed to control FDR at any level on the Mixture-of-Gaussians dataset, indicating that MDNs are less flexible than normalizing flows. In aggregate, these results show that both the normalizing flows paired with MCMC and the use of the HRT for significance testing are key to the performance of FLOWSELECT.

5.4 Real data experiment: soybean GWAS

Genome-wide association studies are a way for scientists to identify genetic variants (single-nucleotide polymorphisms, or SNPs) that are associated with a particular trait (phenotype). We tested FLOWSELECT on a dataset from the SoyNAM project (Song et al., 2017), which is used to conduct GWAS for soybeans. Each feature X_j takes on one of four discrete values, indicating whether a particular SNP is homozygous in the non-reference allele, heterozygous, homozygous in the reference allele, or missing. A number of traits are included in the SoyNAM data; we considered oil content (percentage in the seed) as the phenotype of interest in our analysis. There are 5,128 samples and 4,236 SNPs in total.

To estimate the joint density of the genotypes, we used a discrete flow (Tran et al., 2019). Modeling of genomic data is typically done with a hidden Markov model (Xavier et al., 2016); however, such a model may fail to account for long range dependence between SNPs, which a normalizing flow is better suited to handle. Having a more flexible model of the genome enables FLOWSELECT to provide better FDR control for assessing genotype/phenotype relationships. For the predictive model, we used a feed-forward neural network with three hidden layers. Additional details of training and architecture are presented in Appendix E.

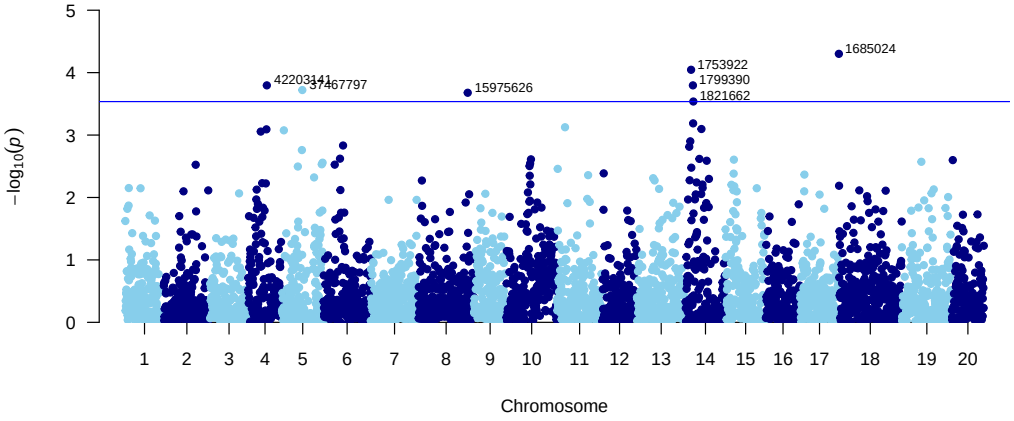

Figure 3: Manhattan plot for oil content in soybean GWAS experiment (Turner, 2018). p is the estimated p-value from the FLOWSELECT procedure, and the blue line indicates the rejection threshold for a nominal FDR of 20%.

A graphical representation of our results is shown as a Manhattan plot in Figure 3, which plots the negative logarithm of the estimated p-values for each SNP. At a nominal FDR of 20%, we identified seven SNPs that are associated with oil content in soybeans. We cross-referenced our discoveries with other publications to identify SNPs that have been previously shown to be associated with oil content in soybeans. For example, FLOWSELECT identifies one SNP on the 18th chromosome, `Gm18_1685024`, which is also selected in Liu et al. (2019). FLOWSELECT also selects a SNP on the 5th chromosome, `Gm05_37467797`, which is near two SNPs (`Gm05_38473956` and `Gm05_38506373`) identified in Cao et al. (2017) but which are not in the SoyNAM dataset. Sonah et al. (2014) identifies eight SNPs near the start of the 14th chromosome, and we select multiple SNPs in a nearby region on the 14th chromosome (seen in the peak of dots on chromosome 14 in Figure 3). However, the dataset in Sonah et al. (2014) is much larger ($\approx 47,000$ SNPs), which prevents an exact comparison. A list of all SNPs selected by our method is provided in Appendix E. For this experiment, FLOWSELECT tests over 4000 features in 10 hours using a single GPU. None of the empirical knockoff procedures (Sudarshan et al., 2020; Jordon et al., 2019; Romano et al., 2020) tested more than 387 features. This shows the potential for FLOWSELECT for high-dimensional feature selection with FDR control in a reasonable amount of time. Additional details about this experiment are available in Appendix E.

6 Discussion

FLOWSELECT enables scientists and other practitioners to discover features a response depends on while controlling false discovery rate, using an arbitrary predictive model; even large-scale nonlinear machine learning models can be utilized. By making fewer false discoveries for a fixed sensitivity level, FLOWSELECT can reduce the cost of follow-up experiments by limiting the number of irrelevant features considered. In contrast to the original model-X knockoffs method, FLOWSELECT does not require the feature distribution to be known a priori, nor does it require the feature distribution to have a particular form (e.g., Gaussian). Neither of these conditions are often satisfied in practice.

One limitation shared by both the conditional randomization test (CRT) and knockoffs is low power in cases in which important features are highly correlated with other important features. To mitigate this limitation, the CRT can be applied to test the significance of groups of correlated features rather than individual features. Within the FLOWSELECT framework, this entails modifying the MCMC step to draw null samples of groups of features conditioned on the others. The group's p-value can then be calculated with the same holdout randomization test (HRT) statistic used for testing individual

features. Group feature selection has also been explored for knockoffs (Dai & Barber, 2016; Liu et al., 2020).

Another limitation of FLOWSELECT stems from its reliance on normalizing flows. The flexibility of normalizing flows, though often beneficial, comes at a cost: sufficient training examples are needed to learn the feature distribution, limiting applicability in data-starved regimes. Fortunately, as we show in Appendix K, FLOWSELECT fares no worse than competing methods in low-data settings. In these regimes, FLOWSELECT could also use other density estimation techniques such as autoregressive models.

Furthermore, learning the feature distribution (potentially from limited data) is not the sole difficulty that the deep-learning-based knockoff methods face. To demonstrate that there are additional sources of difficult for knockoff-based methods, we gave DDLK, which typically fits the data distribution as part of its training procedure, access to the *the exact joint density*; neither the empirical FDR nor the power improved significantly (c.f. Appendix I). This result points to a failure of DDLK to enforce the swap property, which is a challenging task as the number of swaps grows exponentially with the number of features. FLOWSELECT, on the other hand, achieves FDR control under a different set of conditions that often are simpler to satisfy adequately in practice.

Acknowledgments and Disclosure of Funding

Derek Hansen acknowledges support from the National Science Foundation Graduate Research Fellowship Program under grant no. 1256260. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

10x Genomics. Our 1.3 million single cell dataset is ready to download. https://www.10xgenomics.com/blog/our-13-million-single-cell-dataset-is-ready-to-download, 2017.

Agarwal, D., Wang, J., and Zhang, N. Data denoising and post-denoising corrections in single cell rna sequencing. *Statistical Science*, 35(1):112–128, 2020.

Bates, S., Candès, E., Janson, L., and Wang, W. Metropolized knockoff sampling. *Journal of the American Statistical Association*, pp. 1–15, March 2020. ISSN 0162-1459, 1537-274X. doi: 10.1080/01621459.2020.1729163.

Benjamini, Y. and Hochberg, Y. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B*, 57(1):289–300, 1995. doi: 10.1111/j.2517-6161.1995.tb02031.x. URL `https://doi.org/10.1111/j.2517-6161.1995.tb02031.x`.

Benjamini, Y. and Yekutieli, D. The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, 29(4):1165–1188, August 2001. ISSN 0090-5364, 2168-8966. doi: 10.1214/aos/1013699998.

Bogachev, V. I., Kolesnikov, A. V., and Medvedev, K. V. Triangular transformations of measures. *Sbornik: Mathematics*, 196(3):309, April 2005. ISSN 1064-5616. doi: 10.1070/SM2005v196n03ABEH000882.

Breiman, L. Random forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324. URL `http://dx.doi.org/10.1023/A%3A1010933404324`.

Candès, E., Fan, Y., Janson, L., and Lv, J. Panning for gold: 'model-X' knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B*, 80(3):551–577, 2018. ISSN 1467-9868. doi: 10.1111/rssb.12265.

Cao, Y., Li, S., Wang, Z., Chang, F., Kong, J., Gai, J., and Zhao, T. Identification of major quantitative trait loci for seed oil content in soybeans by combining linkage and genome-wide association mapping. *Frontiers in Plant Science*, 8, July 2017. doi: 10.3389/fpls.2017.01222. URL `https://doi.org/10.3389/fpls.2017.01222`.

Dai, R. and Barber, R. The knockoff filter for FDR control in group-sparse and multitask regression. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 1851–1859. PMLR, June 2016.

Fan, Y., Demirkaya, E., Li, G., and Lv, J. RANK: Large-Scale Inference With Graphical Nonlinear Knockoffs. *Journal of the American Statistical Association*, 115(529):362–379, January 2020. ISSN 0162-1459. doi: 10.1080/01621459.2018.1546589.

Friedman, J., Hastie, T., and Tibshirani, R. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, July 2008. ISSN 1465-4644. doi: 10.1093/biostatistics/kxm045.

Gayoso, A., Lopez, R., Xing, G., Boyeau, P., Wu, K., Jayasuriya, M., Mehlman, E., Langevin, M., Liu, Y., Samaran, J., Misrachi, G., Nazaret, A., Clivio, O., Xu, C., Ashuach, T., Lotfollahi, M., Svensson, V., da Veiga Beltrame, E., Talavera-Lopez, C., Pachter, L., Theis, F. J., Streets, A., Jordan, M. I., Regier, J., and Yosef, N. scvi-tools: a library for deep probabilistic analysis of single-cell omics data. *bioRxiv*, 2021. doi: 10.1101/2021.04.28.441833. URL https://www.biorxiv.org/content/early/2021/04/29/2021.04.28.441833.

Gimenez, J. R., Ghorbani, A., and Zou, J. Knockoffs for the mass: New feature importance statistics with false discovery guarantees. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, 16–18 Apr 2019. URL http://proceedings.mlr.press/v89/gimenez19a.html.

Huang, C.-W., Krueger, D., Lacoste, A., and Courville, A. Neural autoregressive flows. In *International Conference on Machine Learning*, July 2018.

Jordon, J., Yoon, J., and van der Schaar, M. KnockoffGAN: Generating knockoffs for feature selection using generative adversarial networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ByeZ5jC5YQ.

Kobyzev, I., Prince, S., and Brubaker, M. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Liu, M., Katsevich, E., Janson, L., and Ramdas, A. Fast and Powerful Conditional Randomization Testing via Distillation. *arXiv:2006.03980*, July 2020.

Liu, Y. and Zheng, C. Auto-Encoding Knockoff Generator for FDR Controlled Variable Selection. *arXiv:1809.10765*, September 2018.

Liu, Y., Wang, D., He, F., Wang, J., Joshi, T., and Xu, D. Phenotype prediction and genome-wide association study using deep convolutional neural network of soybean. *Frontiers in Genetics*, 10, 2019. ISSN 1664-8021. doi: 10.3389/fgene.2019.01091. URL https://www.frontiersin.org/articles/10.3389/fgene.2019.01091/full.

Meng, C., Song, Y., Song, J., and Ermon, S. Gaussianization flows. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, 26–28 Aug 2020. URL http://proceedings.mlr.press/v108/meng20b.html.

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/file/6c1da886822c67822bcf3679d04369fa-Paper.pdf.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22 (57):1–64, 2021.

Romano, Y., Sesia, M., and Candès, E. Deep knockoffs. *Journal of the American Statistical Association*, 115(532):1861–1872, 2020. doi: 10.1080/01621459.2019.1660174. URL https://doi.org/10.1080/01621459.2019.1660174.

Smith, A. F. M. and Roberts, G. O. Bayesian Computation Via the Gibbs Sampler and Related Markov Chain Monte Carlo Methods. *Journal of the Royal Statistical Society: Series B (Methodological)*, 55(1):3–23, 1993. ISSN 2517-6161. doi: 10.1111/j.2517-6161.1993.tb01466.x.

Sonah, H., O'Donoughue, L., Cober, E., Rajcan, I., and Belzile, F. Identification of loci governing eight agronomic traits using a GBS-GWAS approach and validation by QTL mapping in soya bean. *Plant Biotechnology Journal*, 13(2):211–221, September 2014. doi: 10.1111/pbi.12249. URL https://doi.org/10.1111/pbi.12249.

Song, Q., Yan, L., Quigley, C., Jordan, B. D., Fickus, E., Schroeder, S., Song, B.-H., Charles An, Y.-Q., Hyten, D., Nelson, R., Rainey, K., Beavis, W. D., Specht, J., Diers, B., and Cregan, P. Genetic characterization of the soybean nested association mapping population. *The Plant Genome*, 10(2), 2017. doi: https://doi.org/10.3835/plantgenome2016.10.0109. URL https://acsess.onlinelibrary.wiley.com/doi/abs/10.3835/plantgenome2016.10.0109.

Sudarshan, M., Tansey, W., and Ranganath, R. Deep direct likelihood knockoffs. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Tansey, W., Veitch, V., Zhang, H., Rabadan, R., and Blei, D. M. The holdout randomization test for feature selection in black box models. *Journal of Computational and Graphical Statistics*, 2021. [In press; available on arXiv].

Tibshirani, R. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B*, 58:267–288, 1996.

Tran, D., Vafa, K., Agrawal, K., Dinh, L., and Poole, B. Discrete flows: Invertible generative models of discrete data. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Turner, S. qqman: an R package for visualizing GWAS results using Q-Q and Manhattan plots. *The Journal of Open Source Software*, 2018. doi: 10.21105/joss.00731.

Xavier, A., Muir, W., and Rainey, K. Impact of imputation methods on the amount of genetic variation captured by a single-nucleotide polymorphism panel in soybeans. *BMC Bioinformatics*, 17(1), February 2016. doi: 10.1186/s12859-016-0899-7. URL https://doi.org/10.1186/s12859-016-0899-7.

Xavier, A., Beavis, W., Specht, J., Diers, B., Mian, R., Howard, R., Graef, G., Nelson, R., Schapaugh, W., Wang, D., Shannon, G., McHale, L., Cregan, P., Song, Q., Lopez, M., Muir, W., and Rainey., K. *SoyNAM: Soybean Nested Association Mapping Dataset*, 2019. URL https://CRAN.R-project.org/package=SoyNAM. R package version 1.6.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Runtime discussion in section 5.2 and limitations of normalizing flows in section 6
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Theorem 1 statement lists all assumptions.
   (b) Did you include complete proofs of all theoretical results? [Yes] Yes; Appendix B contains the proof to Theorem 1.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code included as supplement, and specific details were described in the Appendix.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In Appendices D and E.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] All figures comparing FDR and Power results (Figure 2 in main text, Figures 4-6, 8 in the Appendix) contain error bars indicating results across runs.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We specify hardware used in Appendix F.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

   (a) If your work uses existing assets, did you cite the creators? [Yes] See Appendix C

   (b) Did you mention the license of the assets? [Yes] See Appendix C

   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] Data are publicly available

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Normalizing flows

Normalizing flows (Papamakarios et al., 2021) represent a general framework for density estimation of a multi-dimensional distribution with arbitrary dependencies. Briefly, suppose $X \sim \mathcal{P}_X$ is a random variable in $\mathbb{R}^d$. Now, let $Z \sim \mathcal{N}(0, I_d)$ be a multivariate standard normal distribution. We assume there exists a mapping G that is triangular, increasing, and differentiable such that

$$G(X) = Z.$$

A formal treatment of when such a G exists can be found in Bogachev et al. (2005). However, a sufficient condition is that the density of X is greater than 0 on $\mathbb{R}^d$ and the cumulative density function of X_j, conditional on the previous components $X_{\leq j}$, is differentiable with respect to $X_j, X_{\leq j}$ (Papamakarios et al., 2021):

$$U_i = G_i(X) \equiv F_i(X_i | X_{\leq i})$$

From this construction, each U_i is independent of all previous U_i and has distribution $\mathrm{Unif}[0, 1]$. From there, we simply set $Z_i = \Phi^{-1}(U_i)$, where Φ is the CDF of the standard normal.

Since $G_i(X)$ depends only on the elements in X up to i, it is triangular. Because $p_X > 0$, the conditional cdfs are strictly increasing, so G is an increasing map. Finally, since each cdf is differentiable, the entire map G is differentiable, and its Jacobian is non-zero.

Because of the inverse mapping theorem, G is invertible and we can write

$$X = G(Z).$$

Normalizing flows are a collection of distributions that parameterize a family of invertible, differentiable transformations G_θ from a fixed base distribution Z to an unknown distribution X. Using the change-of-variables theorem, we can express the distribution of X in terms of the base distribution density p_Z and the transformation G_θ:

$$p_\theta(X) = p(G_\theta(X)) \left| \det \left(\frac{\partial G_\theta(X)}{\partial X} \right) \right|$$

where $\frac{\partial G_\theta(X)}{\partial X}$ is the Jacobian of G. The goal is to find a parameter value $\hat{\theta}$ that maximizes the likelihood of the observed X:

$$\hat{\theta} = \arg\max_\theta p_\theta(X).$$

A key feature of normalizing flows is that they are composable.

A.1 Flow Architecture

In experiments, the first layer G is a Gaussianization flow (Meng et al., 2020) applied elementwise:

$$G_j(X_j) = \Phi^{-1} \left(\sum_{m=1}^{M} \sigma \left(\frac{X_j - \mu_{j,m}}{s_{j,m}} \right) \right),$$

where Φ^{-1} is the standard normal inverse CDF. With sufficiently large M, this Gaussianization layer can approximate any univariate distribution. This is composed with a Masked Autoregressive Flow (MAF) F (Papamakarios et al., 2017), which consists of MADE layers interspersed with batch normalization and reverse permutation layers:

$$\begin{aligned}
\mathrm{MADE}_{j,k} &= (X_j - \mu_{j,k}) \exp(-\alpha_{j,k}) \\
\text{where } \mu_j &= f_{\mu_{j,k}}(X_{<j}) \\
\alpha_j &= f_{\alpha_{j,k}}(X_{<j}) \\
F &= \mathrm{MADE}_{j,K} \circ \mathrm{BatchNorm} \circ \mathrm{Reverse} \circ \mathrm{MADE}_{j,K-1} \circ \cdots \mathrm{BatchNorm} \circ \mathrm{Reverse} \circ \mathrm{MADE}_{j,1}
\end{aligned}$$

Here, f_{μ_j} and f_{α_j} are fully connected neural networks.

B Proof of convergence

Theorem 1. *Let $X \in \mathbb{R}_{N\times D}$ be a random feature matrix, where each row $X_{i,\cdot}$ is independent and identically distributed; $x \in \mathbb{R}_{N\times D}$ be the observed feature matrix; and α_j be the p-value as defined in Equation* (4) *with test statistic $T_j(X)$. Suppose there exists a sequence of functions $(G^n)_{n=1}^{\infty}$ and a base random variable Z satisfying the following conditions:*

1. *Each G^n is continuously differentiable and invertible.*
2. *$G^n \to G$ pointwise for some map G that is triangular, increasing, continuously differentiable, and satisfies $G(X_{i,\cdot}) \overset{D}{=} Z$.*

For $n = 1, 2, \ldots,$ let X^n be the random feature matrix where each row i is independent and has distribution $X^n_{i,\cdot} = (G^n)^{-1}(Z)$. Then, the p-value in Equation (5) *calculated using K MCMC samples targeting $X^n_{\cdot,j} \mid X^n_{\cdot,-j} = x_{\cdot,-j}$ converges to the correct p-value α_j with probability* 1.

Proof of Theorem 1. Without loss of generality, we consider the first feature, which is indexed by $j = 1$. Let p_X be the density of each row of the matrix $X_{i,\cdot}$ and p_Z the density of the base variable Z. For each i.i.d observation at $i = 1, \ldots, N$, we define F to be the cumulative distribution function of $X_{i,1}$ conditional on the other features $X_{i,-1} = x_{i,-1}$:

$$\begin{aligned} F(x_1) \triangleq \mathcal{P}(X_{i,1} \le x_1 | X_{i,-1} = x_{i,-1}) &= \frac{\int_{-\infty}^{x_1} p_X(x_1', x_{i,-1}) dx_1'}{\int_{-\infty}^{\infty} p_X(x_1', x_{i,-1}) dx_1'} \\ &= \frac{\int_{-\infty}^{x_1} p_Z(G(x_1', x_{i,-1})) |\partial G(x_1', x_{i,-1})| dx_1'}{\int_{-\infty}^{\infty} p_Z(G(x_1', x_{i,-1})) |\partial G(x_1', x_{i,-1})| dx_1'}. \end{aligned} \tag{7}$$

For a particular mapping G^n, we define F^n analogously:

$$F^n(x_1) \triangleq \frac{\int_{-\infty}^{x_1} p_Z(G^n(x_1', x_{i,-1})) |\partial G^n(x_1', x_{i,-1})| dx_1'}{\int_{-\infty}^{\infty} p_Z(G^n(x_1', x_{i,-1})) |\partial G^n(x_1', x_{i,-1})| dx_1'}. \tag{8}$$

Since G^n and G are continuously differentiable,

$$p_Z(G^n(x_1', x_{i,-1})) |\partial G^n(x_1', x_{i,-1})| \to p_Z(G(x_1', x_{i,-1})) |\partial G(x_1', x_{i,-1})| \text{ as } n \to \infty. \tag{9}$$

Then, by the dominated convergence theorem, $F^n \to F$ pointwise.

Let $X^n_{i,1} \sim F^n$. Since $F^n \to F$ pointwise, and F is a distribution function, $X^n_{i,1}$ converges in distribution to $X_{i,1} \mid X_{i,-1} = x_{i,-1}$. Likewise, the joint distribution across all independent observations, written $X^n_{\cdot,1}$, converges in distribution to $X_{\cdot,1} \mid X_{\cdot,-1} = x_{\cdot,-1}$.

Now, let $\tilde{X}^n_{\cdot,1}$ be equal in distribution to $X^n_{\cdot,1}$, but sampled such that it is independent of the outcome Y. It follows from the reasoning above that $\tilde{X}^n_{\cdot,1}$ converges to the desired null distribution $\tilde{X}_{\cdot,1} | X_{\cdot,-1}$ as $n \to \infty$. Define $g_1(\tilde{x}_{\cdot,1}) \triangleq 1[T_1(X) < T_1([\tilde{x}_{\cdot,1}, X_{\cdot,-1}])]$. With the regularity condition that T_1 is discontinuous on a set of measure zero, the expectation converges:

$$\lim_{n\to\infty} \mathbb{E}_{\tilde{X}^n_{\cdot,1}}(g_1) \to \mathbb{E}_{\tilde{X}_{\cdot,1}|X_{\cdot,-1}=x_{\cdot,-1}}(g_1) = \alpha_1. \tag{10}$$

The Cesaro average of g calculated over MCMC samples that target the distribution of $\tilde{X}^n_{\cdot,1}$ under the probability law of G_n converges almost surely to $\mathbb{E}_{\tilde{X}^n_{\cdot,1}}(g_1)$ (Smith & Roberts, 1993). That is,

$$\lim_{K\to\infty} \hat{\alpha}_{j,K,n} = \lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} g_1(\tilde{X}_{\cdot,1,k}) = \mathbb{E}_{\tilde{X}^n_{\cdot,1}}(g_1) \; w.p.1. \tag{11}$$

Combining Equation (10) and Equation (11) gives the desired result. □

C Feature datasets

Name	Covariate	Response	N	D	# Relevant	Source
Gaussian Mixture	Synthetic	Synthetic	$100,000$	100	20	-
scRNA-seq	Real	Synthetic	$100,000$	100	10	10x Genomics (2017)
Soybean	Real	Real	$5,128$	$4,236$	-	Xavier et al. (2019)

Licensing All of the data used is available for personal use. Terms for the scRNA-seq data can be found here: `https://www.10xgenomics.com/terms-of-use`. The scRNA-seq data was accessed using scvi-tools (Gayoso et al., 2021), distributed under the BSD 3-Clause license. The soybean data is part of the SoyNAM R package (Xavier et al., 2019), distributed under the GPL-3 license.

D Architecture and training details for synthetic experiments

D.1 FlowSelect

For FLOWSELECT, the joint distribution was fitted with a GaussMAF normalizing flow as described in Appendix A. The first Gaussianization layer consisted of $M = 6$ clusters, followed by 5 layers of MAF. Within each MAF layer, the neural network consisted of three masked fully connected residual layers with 100 hidden units, followed by a BatchNorm layer.

We trained the Gaussianization layer first with 100 epochs and learning rate 1×10^{-3} within the ADAM optimizer. This allowed the Gaussianization layer to learn the marginal distribution of each feature. Then, we jointly trained the whole architecture with 100 epochs and learning rate 1×10^{-3} using ADAM.

MCMC We draw 1000 samples using a Metropolis-Hastings procedure. The proposal distribution is a random walk:

$$X^*_{i,j,k} \sim \mathcal{N}(\tilde{X}_{i,j,k-1}, \hat{\sigma}^2_j),$$

where $\hat{\sigma}^2_j$ is the sample conditional variance:

$$\hat{\sigma}^2_j = \hat{\Sigma}_{j,j} - \hat{\Sigma}_{j,-j}\hat{\Sigma}^{-1}_{-j,-j}\hat{\Sigma}^T_{j,-j}$$
$$\text{where } \hat{\Sigma}_j = \widehat{\text{Var}}(X)$$

D.2 Variable selection methods

Linear For the linear response, we estimate a linear model with an L1 penalty (aka the LASSO) on training data:

$$\hat{\beta} = \arg\min_{\beta} \frac{1}{N}\|X\beta - Y\|_2^2 + \lambda \sum_{j=1}^{D} |\beta_j| \tag{12}$$

The penalization term λ is selected via 5-fold cross-validation.

Nonlinear For the nonlinear response, we fit a random forest on the training data. The hyperparameters are the defaults in the scikit-learn implementation.

Feature statistic If $\hat{f}(X)$ is the fitted regression function, then the feature statistic is the negative mean-squared error:

$$T(X, Y) = -\frac{1}{N}\|\hat{f}(X) - Y\|_2^2.$$

D.3 Competing methods

For DDLK (Sudarshan et al., 2020), KnockoffGAN (Jordon et al., 2019), and DeepKnockoffs, (Romano et al., 2020), we used the exact architecture and hyperparameter settings from their respective papers. For the ablation study in Section 5.3, we use the exact implementation in Tansey et al. (2021). For these methods, we used the code that the researchers graciously made publicly available:

Method	Link
DDLK	`https://github.com/rajesh-lab/ddlk/`
DeepKnockoffs	`https://github.com/msesia/deepknockoffs/`
HRT (MDN)	`https://github.com/tansey/hrt/`
KnockoffGAN	`https://github.com/firmai/tsgan/tree/master/alg/knockoffgan`

For MASS (Gimenez et al., 2019), we followed their described procedure and fit a mixture of Gaussians to the feature distribution using scikit-learn, selecting the number of components via the Akiake Information Criterion (AIC). We then used the `knockoffs` R package, available on CRAN, to sample knockoffs using the estimated parameters for each component.

For RANK (Fan et al., 2020), we estimate the sparse precision matrix using the Graphical LASSO (Friedman et al., 2008) implemented in sci-kit learn, using cross-validation to tune the regularization parameter. We then use the `knockoffs` R package to sample the knockoffs with this covariance.

E Architecture and training details for soybean GWAS

Discrete flows For the discrete flows in the soybean example, we use a single layer of MADE which outputs a dimension of size 4. μ is then set equal to the argmax of this output.

For training the flows, we use a relaxation of argmax with temperature equal to 0.1.

Discrete MCMC Each feature has $K = 4$ values, so we can enumerate all four possible states for each proposal and sample in proportional to these probabilities via a Gibbs Sampling procedure. Setting the probabilities leads to an acceptance rate of 1, and the samples are uncorrelated since the previous sample doesn't enter into the proposal distribution

Predictive model For the predictive model of each trait conditional on the SNPs, we use a fully connected neural network. This network has three hidden layers of size 128, 256, and 128. ReLU activations are used between each fully connected layer. Dropout is used on both the input layer and after each hidden layer with $p = 0.2$. The learning rate in ADAM was set to 1×10^{-5}, with early stopping implemented using a held-out validation set.

The feature statistic for each sample is the negative mean-squared error (MSE) for each observation.

Runtime To obtain sufficient resolution on roughly 4200 simultaneous tests, we drew 100,000 samples from our model. The runtime was 10 hours using a single NVIDIA 2080 Ti.

Selected SNPs Table 1 shows the SNPs selected by FLOWSELECT that are associated with oil content in soybeans.

Chromosome	SNP	p-value
4	Gm04_42203141	1.60e-04
5	Gm05_37467797	1.90e-04
8	Gm08_15975626	2.10e-04
14	Gm14_1753922	9.00e-05
14	Gm14_1799390	1.60e-04
14	Gm14_1821662	2.90e-04
18	Gm18_1685024	5.00e-05

Table 1: Selected SNPs for soybean GWAS experiment.

F Runtime comparison of each controlled feature selection method

Method	Runtime (min)
DeepKnockoff	3.0
KnockoffGAN	3.73
MASS	12.6
DDLK	91.9
FLOWSELECT	59.4

Table 2: The median runtime for each method on the scRNA-seq data with $D = 100$ features and $N = 100,000$ observations. All experiments were implemented using PyTorch, except for KnockoffGAN, which was implemented in Tensorflow, and MASS, which we implemented using scikit-learn and the knockoffs R package. The experiments were conducted using an Intel Xeon Gold 6130 CPU and an NVIDIA GeForce RTX 2080 Ti GPU.

G Comparison to Holdout Randomization Test of Tansey et al. (2021)

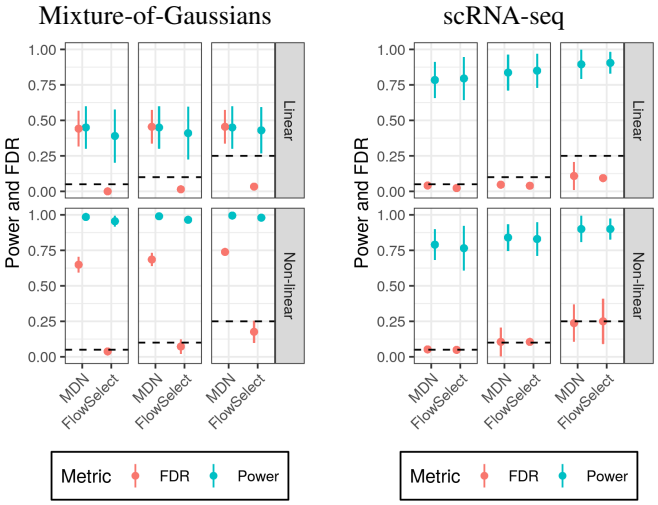

Figure 4: Comparison of FLOWSELECT to the HRT procedure in Tansey et al. (2021) which samples the complete conditionals using multiple mixture-density-networks (MDNs). Each column shows the power and observed false discovery rate (FDR) at targeted FDRs of 0.05, 0.1, and 0.25 (indicated by the dashed lines). The experimental settings for each dataset are the same as in Figure 2.

H Oracle Model-X

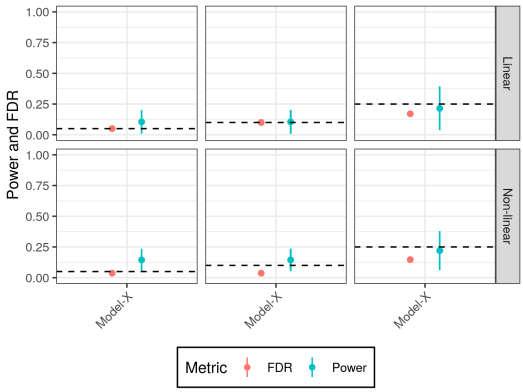

Figure 5: FDR control and power of Oracle Model-X knockoffs on the mixture-of-Gaussians dataset (compare to Figure 2).

I DDLK with true joint distribution

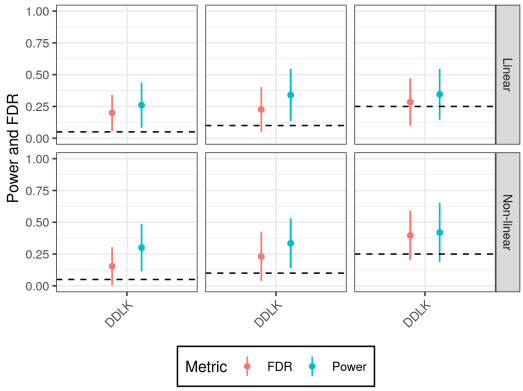

Figure 6: FDR control and power of DDLK on the mixture-of-Gaussians dataset using the ground truth feature density in training (compare to Figure 2).

J Observed Power and FDR control for given number of MCMC samples

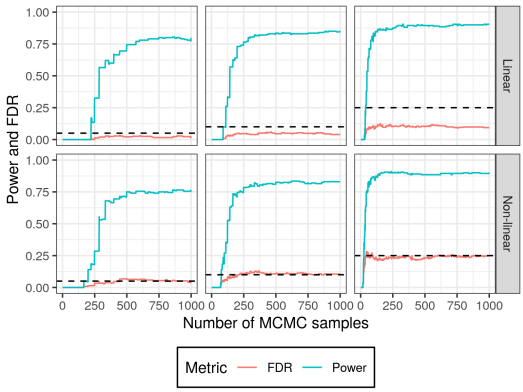

Figure 7: Power and FDR control of FLOWSELECT on the scRNA-seq dataset as a function of the number of MCMC samples at targeted FDRs of 0.05, 0.1, and 0.25 (indicated by the dashed lines). This suggests that the consequence of terminating the MCMC chain prematurely leads to a drop in power but FDR control is still maintained.

K Mixture-of-Gaussians results for FDR and Power under Sudarshan et al. (2020) settings

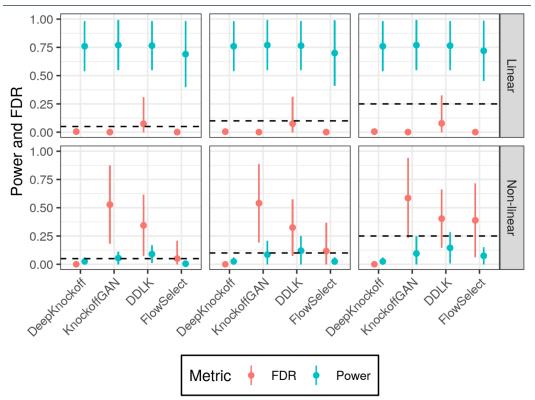

Figure 8: Mixture-of-gaussians setup with $\rho = (0.6, 0.4, 0.2)$ and $N = 2000$ to match the settings in Sudarshan et al. (2020). In the linear response setting, which matches the data-generating process of Sudarshan et al. (2020), all competing knockoff-based methods (i.e., DDLK, KnockoffGAN, and DeepKnockoff) as well as FlowSelect control the FDR at 5%, 10% and 25% levels and achieve a power of about 0.75. In the non-linear response setting, none of the methods control FDR, except for DeepKnockoffs which had nearly zero power. The good performance in the linear setting can be explained by the LASSO feature statistic shrinking most null features to zero since they have relatively low correlation. Since FDR control should hold for any response setting, these findings suggest that none of the methods do well in modeling the underlying distribution with $N = 2000$ observations.

L Learned normalizing flow mapping on mixture-of-Gaussians and scRNA-seq datasets

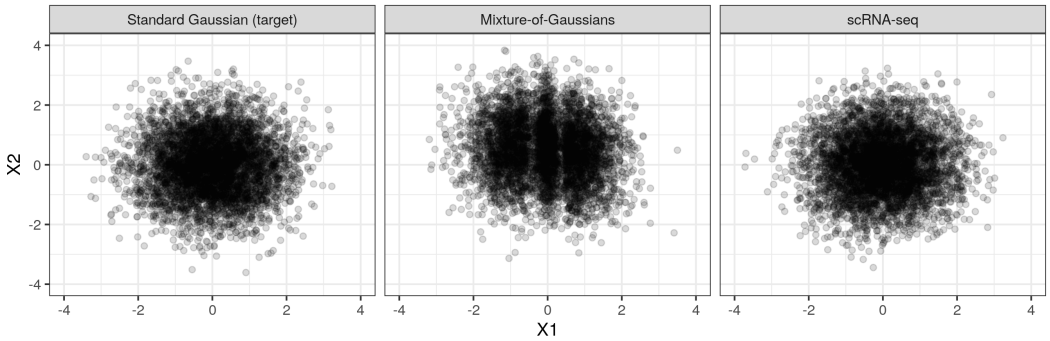

Figure 9: Plot of features mapped to flow space by the learned normalizing flow within FLOWSELECT with $j = 1$ on the x-axis and $j = 2$ on the y-axis. Mapped features are shown for the mixture-of-Gaussians and scRNA-seq datasets, and they are compared to samples from a true standard Gaussian distribution.