Metropolized Knockoff Sampling

Stephen Bates*[1], Emmanuel Candès[1,2], Lucas Janson[3], and Wenshuo Wang[3]

[1]Department of Statistics, Stanford University, Stanford, CA 94305
[2]Department of Mathematics, Stanford University, Stanford, CA 94305
[3]Department of Statistics, Harvard University, Cambridge, MA 02138

March 1, 2019

Abstract

Model-X knockoffs is a wrapper that transforms essentially any feature importance measure into a variable selection algorithm, which discovers true effects while rigorously controlling the expected fraction of false positives. A frequently discussed challenge to apply this method is to construct knockoff variables, which are synthetic variables obeying a crucial exchangeability property with the explanatory variables under study. This paper introduces techniques for knockoff generation in great generality: we provide a sequential characterization of all possible knockoff distributions, which leads to a Metropolis–Hastings formulation of an *exact* knockoff sampler. We further show how to use conditional independence structure to speed up computations. Combining these two threads, we introduce an explicit set of sequential algorithms and empirically demonstrate their effectiveness. Our theoretical analysis proves that our algorithms achieve near-optimal computational complexity in certain cases. The techniques we develop are sufficiently rich to enable knockoff sampling in challenging models including cases where the covariates are continuous and heavy-tailed, and follow a graphical model such as the Ising model.

Keywords. False discovery rate (FDR), Metropolis–Hastings, Markov chain, graphical models, Ising model, junction tree, treewidth

1 Introduction

In modern science, researchers often have access to large data sets featuring comprehensive measurements about some phenomenon of interest. The question is then to discover meaningful relationships between an outcome and all the measured covariates. While it is often expected that only a small fraction of the covariates may be associated with the outcome, the relevance of any particular variable is unknown a priori. For instance, a researcher may be interested in understanding which of the thousands of gene-expression profiles may help determine the severity of a tumor. In such circumstances, the researcher often relies on statistical algorithms to sift through large data sets and find those promising candidates, making variable selection a topic of central importance in contemporary statistical research.

The knockoff filter (Barber and Candès 2015; Candès et al. 2018) has recently emerged as a useful framework for performing controlled variable selection, allowing the user to convert any black-box

*Authors are listed in alphabetical order.

feature importance measure into a variable selection procedure while rigorously controlling the expected fraction of false positives. This means that the statistician can use essentially any black-box importance measure to return a list of variables with the guarantee that, on the average, the ratio between the number of false positives—loosely speaking, a false positive is a variable that does not influence the response, see Candès et al. (2018)—and the total number of reported variables is below a user-specified threshold. The strength of this method is that the guarantees hold in finite samples and in situations where nothing can be assumed about the dependence between the response and the explanatory variables. Instead, the statistician must have knowledge of the distribution of the explanatory variables. When this happens to be the case, a remaining challenge is the ability to generate the *knockoffs*, a set of synthetic variables, which can essentially be used as negative controls; these fake variables must mimic the original variables in a particular way without having any additional predictive power. In sum, constructing valid knockoff distributions and sampling mechanisms across a wide range of covariate models is critical to deploying the knockoff filter in a number of applications.

1.1 Our contribution

This paper describes a theory for sampling knockoff variables and introduces a general and efficient sampler inspired by ideas from Markov chain Monte Carlo (MCMC). Before moving on, we pause to explicitly mention the two main considerations one should keep in mind when constructing knockoffs:

Computation. How can we *efficiently* sample nontrivial knockoffs?

Statistical power. How can we generate knockoffs that will ultimately lead to *powerful* variable selection procedures? On this note, it has been observed that knockoffs that are less correlated with the original variables lead to higher power (Barber and Candès 2015; Candès et al. 2018) and, therefore, low correlation must be a design objective.

Having said that, our work makes several specific contributions.

1. **Characterization of all knockoff distributions**. We provide a sequential characterization of *every* valid knockoff distribution. Furthermore, we introduce a connection linking pairwise exchangeability between original and knockoff variables to reversible Markov chains, enabling the use of powerful sampling tools from computational statistics.

2. **Complexity of knockoff sampling procedures**. We introduce a class of algorithms which use conditional independence information to efficiently generate knockoffs. The computational complexity of such procedures is shown to be determined by the complexity of the dependence structure in a precise way. Furthermore, we present a lower bound on complexity showing that structural assumptions are necessary for efficient computation, and that our procedure achieves the lower bound in certain cases.

3. **Practical sampling algorithms**. We develop a concrete knockoff sampler for a large number of distributions. This is achieved by constructing a family of MCMC tools—designed to have good performance—which only require the numerical evaluation of an *unnormalized* density. We identify a default parameter setting for the sampler that performs well across a variety of situations, producing a general and easy-to-use tool for practitioners.

We shall see that our ideas enable knockoff sampling in challenging models including situations where the covariates are continuous and heavy-tailed and where they follow an Ising model.

1.2 Related literature

This paper draws most heavily on Candès et al. (2018), which builds on Barber and Candès (2015) to introduce the model-X knockoff framework. In particular, the former reference proposes the *Sequential Conditional Independent Pairs* (SCIP) procedure for knockoff generation; this is the only known generic knockoff sampler to date, which shall serve as our starting point. The SCIP procedure, however, is only abstractly specified and prior to this paper, implementations were only available for Gaussian distributions and discrete Markov chains. Briefly, Sesia et al. (2018) developed a concrete SCIP algorithm for discrete Markov chains, and then leveraged this construction to sample knockoffs for covariates following hidden Markov models widely used in genome-wide association studies. Similarly relevant is the work of Gimenez et al. (2018), which developed a sampling strategy for a restricted class of Bayesian networks, most notably Gaussian mixture models. In contrast, we address here knockoff sampling for a much larger class of distributions, namely, arbitrary graphical models. We also describe the form of all valid knockoff sampling strategies, thereby providing a framework possibly enabling the construction of future knockoff sampling algorithms. Hence, our work may be of value to the increasing number of researchers deploying the knockoff framework for feature selection in a variety of applications including neural networks (Lu et al. 2018), time-series modeling (Fan et al. 2018), Gaussian graphical model structure learning (Zheng et al. 2018), and biology (Xiao et al. 2017; Gao et al. 2018). Lastly, we close by emphasizing that our contribution is very different from a new strand of research introducing approximate knockoffs generated with techniques from deep learning (Romano et al. 2018; Jordon et al. 2019; Liu and Zheng 2018). While these approaches are tantalizing and demonstrate promising empirical performance in low-dimensional situations, they currently lack formal guarantees about their validity.

2 Characterizing knockoff distributions

2.1 Knockoff variables

Consider random covariates $X = (X_1, X_2, \dots, X_p)$. We say that the random variables $\tilde{X} = (\tilde{X}_1, \tilde{X}_2, \dots, \tilde{X}_p)$ are *knockoffs* for X if for each $j = 1, \dots, p$,

$$(X, \tilde{X})_{\text{swap}(j)} \overset{d}{=} (X, \tilde{X}). \tag{1}$$

Here, the notation swap(j) means permuting X_j and $\tilde{X}_j$; for instance, $(X_1, X_2, X_3, \tilde{X}_1, \tilde{X}_2, \tilde{X}_3)_{\text{swap}(2)}$ is the vector $(X_1, \tilde{X}_2, X_3, \tilde{X}_1, X_2, \tilde{X}_3)$.[a] Property (1) is known as the *pairwise exchangeability* property, and it is in general challenging to define joint distributions $(X, \tilde{X})$ satisfying this condition. Before continuing, we briefly pause to understand the meaning of pairwise exchangeability. A consequence of (1) is that for all sets $A \subseteq \{1, \dots, p\}$,

$$(X, \tilde{X})_{\text{swap}(A)} \overset{d}{=} (X, \tilde{X}),$$

where $(X, \tilde{X})_{\text{swap}(A)}$ denotes the swapping of X_j and $\tilde{X}_j$ for all $j \in A$. Taking $A = \{1, \dots, p\}$ and marginalizing, we immediately see that $\tilde{X} \overset{d}{=} X$; that is, X and $\tilde{X}$ are distributed in the same way. Also changing any subset of entries of X with their knockoff counterparts does not change the distribution either. Another consequence of the exchangeability property (1) is that the

[a] In the presence of a response Y, we also require $\tilde{X} \perp\!\!\!\perp Y \mid X$, which is easily satisfied by procedures that generate $\tilde{X}$ from X without looking at Y.

mixed second moments of $(X, \tilde{X})$ must match. Assume the second moments of X exist and write $\mathbf{\Sigma} = \mathrm{Cov}(X)$. Then the covariance of the vector $(X, \tilde{X})$ must take the form

$$\mathrm{Cov}(X, \tilde{X}) = \mathbf{\Gamma}(s) := \begin{bmatrix} \mathbf{\Sigma} & \mathbf{\Sigma} - \mathrm{diag}(s) \\ \mathbf{\Sigma} - \mathrm{diag}(s) & \mathbf{\Sigma} \end{bmatrix}, \tag{2}$$

where $s \in \mathbb{R}^p$ is any vector such that the right-hand side is positive semi-definite. In other words, for each pair (i, j) with $i \neq j$, we have $\mathrm{Cov}(X_i, \tilde{X}_j) = \mathrm{Cov}(X_i, X_j)$.

We are interested in constructing knockoff variables and below we call a *knockoff sampler* a procedure that takes as inputs a distribution $\mathbb{P}$ and a sample $X \sim \mathbb{P}$ and returns $\tilde{X}$ such that (1) holds. Nontrivial samplers have been demonstrated in a few cases, for instance, when $X \sim \mathcal{N}(0, \mathbf{\Sigma})$ is multivariate Gaussian. In this case, Candès et al. (2018) show that if $(X, \tilde{X})$ is jointly Gaussian with mean zero and covariance $\mathbf{\Gamma}(s)$, then the entries of $\tilde{X}$ are knockoffs for X. One can say that appropriately matching the first two moments is sufficient to generate knockoffs in the special case of the multivariate normal distribution. However, this does not extend and matching the first two moments is in general not sufficient; to be sure, (1) requires that all moments match appropriately.

Gibbs measures. As a motivating example, consider the Ising model, a frequently discussed family of Gibbs measures first introduced in the statistical physics literature (Ising 1925). In this model, the random vector $X \in \{-1, 1\}^{d_1 \times d_2}$ defined over a $d_1 \times d_2$ $X \in \{-1, 1\}^{d_1 \times d_2}$ grid has a probability mass function (PMF) of the form

$$\mathbb{P}(X) = \frac{1}{Z(\beta, \alpha)} \exp\left(\sum_{\substack{s,t \in \mathcal{I} \\ \|s-t\|_1 = 1}} \beta_{st} X_s X_t + \sum_{s \in \mathcal{I}} \alpha_s X_s \right); \tag{3}$$

here, $\mathcal{I} = \{(i_1, i_2) : 1 \leq i_1 \leq d_1, 1 \leq i_2 \leq d_2\}$ is the grid and α and β are parameters. As we have seen, knockoffs $\tilde{X}$ for X must marginally follow the Ising distribution (3). Furthermore, $\tilde{X}$ must be dependent on X in such a way that any vector of the form $\{(Z_1, \ldots, Z_p) : Z_j = X_j \text{ or } Z_j = \tilde{X}_j, 1 \leq j \leq p\}$ has PMF given by (3). It is tempting to naïvely define a joint PMF for $(X, \tilde{X})$ as

$$\mathbb{P}(X, \tilde{X}) \propto \exp\left(\sum_{\substack{s,t \in \mathcal{I} \\ \|s-t\|_1 = 1}} \beta_{st} (X_s X_t + \tilde{X}_s \tilde{X}_t + X_s \tilde{X}_t + \tilde{X}_s X_t) + \sum_{s \in \mathcal{I}} \alpha_s (X_s + \tilde{X}_s) \right).$$

Although the joint distribution is symmetric in X_s and $\tilde{X}_s$, the marginal distribution of X is not an Ising model! Hence, this is not a valid joint distribution. Other than the trivial construction $\tilde{X} = X$, it is a priori unclear how one would construct knockoffs. Any distribution continuous or discrete factoring over a grid poses a similar challenge.

2.2 SCIP and its limitations

The only generic knockoff sampler one can find in the literature is SCIP from Candès et al. (2018), given in Procedure 1. While this procedure provably generates valid knockoffs for any input distribution, there are two substantial limitations. The first is that SCIP is only given abstractly; it is challenging to specify $\mathcal{L}(X_j \mid X_{-j}, \tilde{X}_{1:(j-1)})$,[b] let alone to sample from it. As a result, it is

[b] We use $\mathcal{L}(W_1 \mid W_2)$ to denote the conditional distribution of W_1 given W_2. We use the subscript $1:0$ to mean an empty vector.

only known how to implement SCIP for very special models such as discrete Markov chains and Gaussian distributions. The second limitation is that SCIP is not able to generate all valid knockoff distributions. Recall that we want knockoffs to have low correlations with the original variables so that a feature importance statistic will correctly detect true effects. To achieve this goal, we might need a wider range of sampling mechanisms.

Procedure 1: Sequential Conditional Independent Pairs (SCIP)

for $j = 1$ **to** p **do**
  Sample $\tilde{X}_j$ from $\mathcal{L}(X_j \mid X_{\text{-}j}, \tilde{X}_{1:(j-1)})$, conditionally independently from X_j
end

2.3 Sequential formulation of knockoff distributions

We begin by introducing a sequential characterization of *all* valid knockoff distributions, which will later lead to a new class of knockoff samplers.

Theorem 1 (Sequential characterization of knockoff distributions)**.** *Let $(X, \tilde{X}) \in \mathbb{R}^{2p}$ be a random vector. Then pairwise exchangeability* (1) *holds if and only if both of the following conditions hold:*

Conditional exchangeability *For each $j \in \{1, \dots, p\}$,*

$$(X_j, \tilde{X}_j) \mid X_{\text{-}j}, \tilde{X}_{1:(j-1)} \stackrel{d}{=} (\tilde{X}_j, X_j) \mid X_{\text{-}j}, \tilde{X}_{1:(j-1)}. \tag{4}$$

Knockoff symmetry *For each $j \in \{1, \dots, p\}$,*

$$\mathbb{P}((X_j, \tilde{X}_j) \in A \mid X_{\text{-}j}, \tilde{X}_{1:(j-1)}) \tag{5}$$

is $\sigma(X_{(j+1):p}, \{X_1, \tilde{X}_1\}, \dots, \{X_{j-1}, \tilde{X}_{j-1}\})$-measurable for any Borel set A, where $\{\cdot, \cdot\}$ denotes the unordered pair. That is, the conditional distribution does not change if we swap previously sampled knockoffs with the original features.

Theorem 1 implies that a sequential knockoff sampling algorithm faithful to these two conditions is as general as it gets. The challenge now becomes creating exchangeable random variables at each step (with a little caution on the dependence on the previous pairs of variables). In turn, this task happens to be equivalent to designing a time-reversible Markov chain, as formalized below.

Proposition 1. *A pair of random variables $(Z, \tilde{Z})$ is exchangeable, i.e., $(Z, \tilde{Z}) \stackrel{d}{=} (\tilde{Z}, Z)$, with marginal distribution π for Z—and, therefore, for $\tilde{Z}$ as well—if and only if there exists a time-reversible Markov chain $\{Z_n\}_{n=1}^{\infty}$ such that $Z_1 \sim \pi$ is a stationary distribution of the chain, and $(Z_1, Z_2) \stackrel{d}{=} (Z, \tilde{Z})$.*

Combining these two results gives SCEP (Procedure 2 below), which is a completely general strategy for generating knockoffs: at each step j, we design a time-reversible Markov chain with stationary distribution $\mathcal{L}(X_j \mid X_{\text{-}j}, \tilde{X}_{1:(j-1)})$, and draw a sample by taking one step of this chain starting from X_j. Proposition 1 implies that the conditional exchangeability (4) holds. Furthermore, the symmetry requirement on the transition kernel implies that SCEP does not break the

exchangeability from previous steps; that is, the knockoff symmetry (5) also holds. Theorem 1 then implies that such a procedure produces valid knockoffs.

Procedure 2: Sequential Conditional Exchangeable Pairs (SCEP)

for $j = 1$ **to** p **do**

Sample $\tilde{X}_j$ by taking one step of a time-reversible Markov chain starting from X_j. The transition kernel must be such that it depends only on $X_{(j+1):p}$ and the unordered pairs $\{X_1, \tilde{X}_1\}, \ldots, \{X_{j-1}, \tilde{X}_{j-1}\}$, and admits $\mathcal{L}(X_j \mid X_{\text{-}j}, \tilde{X}_{1:(j-1)})$ as a stationary distribution.

end

To rehearse the universality of SCEP, consider an arbitrary knockoff sampler producing $\tilde{X}_1, \ldots, \tilde{X}_p$. Then from Theorem 1 we know that X_1 and $\tilde{X}_1$ must be exchangeable conditional on $X_{\text{-}1}$. Therefore, $\tilde{X}_1$ may be sampled by taking one step of a reversible Markov chain starting at X_1. Moving on to X_2, Theorem 2 informs us that X_2 and $\tilde{X}_2$ are exchangeable conditional on $\{X_1, \tilde{X}_1\}, X_3, \ldots, X_p$, so $\tilde{X}_2$ can again be viewed as taking one step of a reversible Markov chain starting at X_2. Continuing in this fashion for $j = 3, \ldots, p$ establishes our claim.

SCEP as stated remains too abstract to be considered an implementable algorithm, so we will next develop a concrete version of this procedure. Although this may not yet be clear, we would like to stress that formulating a knockoff sampler in terms of reversible Markov chains is an important step forward because it will ultimately enable the use of flexible MCMC tools.

3 The Metropolized knockoff sampler

We now demonstrate how one can generate knockoffs in a sequential manner by making proposals which are either accepted or rejected in a Metropolis–Hastings-like fashion as to ensure pairwise exchangeability.

3.1 Algorithm description

The celebrated Metropolis–Hastings (MH) algorithm (Metropolis et al. 1953; Hastings 1970) provides a general time-reversible Markov transition kernel whose stationary distribution is an arbitrary density function π. To construct a transition from x to y, MH operates as follows: generate a proposal x^* from a distribution $q(\cdot \mid x)$ (any distribution depending on x) and set[c]

$$y = \begin{cases} x^* & \text{with prob. } \alpha, \\ x & \text{with prob. } 1 - \alpha, \end{cases} \qquad \alpha = \min\left(1, \frac{\pi(x^*)q(x \mid x^*)}{\pi(x)q(x^* \mid x)}\right).$$

This can be implemented even when the density π is unnormalized, as the normalizing constants cancel. In our setting, we shall make sure that the choice of the proposal distribution depends on the previously sampled pairs in a symmetric fashion, thereby remaining faithful to the knockoff symmetry condition (5) in Theorem 1. As such, we call such proposals *faithful*.

Consider now running SCEP (Procedure 2) with the MH kernel, where at the jth step, the target distribution π is taken to be $\mathcal{L}(X_j \mid X_{\text{-}j}, \tilde{X}_{1:j-1})$. The issue with such a naïve implementation is that the target π cannot be readily evaluated. To understand why this is the case, set $j = 2$ and consider

[c]More generally, we take as acceptance probability $\gamma\,\alpha$ with $\gamma \in (0, 1]$. In this work, γ is set to 1 as default, except in Section 3.3 and Appendix F.2, which are cases where tuning γ is recommended.

$\mathcal{L}(X_2 \mid X_{\text{-}2}, \tilde{X}_1)$. This distribution has density proportional to $\mathbb{P}(X = x)\mathbb{P}(\tilde{X}_1 = \tilde{x}_1 \mid X = x)$, which is equal to

$$\mathbb{P}(X = x)\Bigg[q(\tilde{x}_1 \mid x_1)\min\left(1, \frac{q(x_1 \mid \tilde{x}_1)\mathbb{P}(X_1 = \tilde{x}_1, X_{\text{-}1} = x_{\text{-}1})}{q(\tilde{x}_1 \mid x_1)\mathbb{P}(X_1 = x_1, X_{\text{-}1} = x_{\text{-}1})}\right) + \delta(\tilde{x}_1 - x_1)\int q(x^* \mid x_1)\left(1 - \min\left(1, \frac{q(x_1 \mid x^*)\mathbb{P}(X_1 = x^*, X_{\text{-}1} = x_{\text{-}1})}{q(x^* \mid x_1)\mathbb{P}(X_1 = x_1, X_{\text{-}1} = x_{\text{-}1})}\right)\right)dx^*\Bigg]. \quad (6)$$

The first term in the summation within the brackets corresponds to the acceptance case while the second corresponds to the rejection case. This latter term cannot be evaluated because of the integral over x^*. Hence, the target density cannot be evaluated either.

We propose an effective solution to this problem: *condition on the proposals* and at step j, let the target distribution be $\mathcal{L}(X_j \mid X_{\text{-}j}, \tilde{X}_{1:j-1}, X^*_{1:j-1})$ rather than $\mathcal{L}(X_j \mid X_{\text{-}j}, \tilde{X}_{1:j-1})$. This has the effect of removing the integral and makes computing the rejection probability tractable. This is best seen by returning to our example where $j = 2$. Here, $\mathcal{L}(X_j \mid X_{\text{-}j}, \tilde{X}_{1:j-1}, X^*_{1:j-1})$ has density now proportional to

$$\mathbb{P}(X = x)q(x^*_1 \mid x_1)\Bigg[\delta(\tilde{x}_1 - x^*_1)\min\left(1, \frac{q(x_1 \mid \tilde{x}_1)\mathbb{P}(X_1 = \tilde{x}_1, X_{\text{-}1} = x_{\text{-}1})}{q(\tilde{x}_1 \mid x_1)\mathbb{P}(X_1 = x_1, X_{\text{-}1} = x_{\text{-}1})}\right) + \delta(\tilde{x}_1 - x_1)\left(1 - \min\left(1, \frac{q(x_1 \mid x^*_1)\mathbb{P}(X_1 = x^*_1, X_{\text{-}1} = x_{\text{-}1})}{q(x^*_1 \mid x_1)\mathbb{P}(X_1 = x_1, X_{\text{-}1} = x_{\text{-}1})}\right)\right)\Bigg]. \quad (7)$$

We will show in Section 4 how such terms can be efficiently computed. Leaving aside implementation details for the moment, this strategy leads to Algorithm 1. Here and elsewhere, $\mathbb{P}$ denotes the density of the variables under study, or formally, the Radon–Nikodym derivative with respect to a common dominating measure.

Algorithm 1: Metropolized knockoff sampling (Metro).

for $j = 1$ **to** p **do**

Sample $X^*_j = x^*_j$ from a faithful proposal distribution q_j.

Accept the proposal with probability

$$\min\left(1, \frac{q_j(x_j \mid x^*_j)\mathbb{P}\left(X_{\text{-}j} = x_{\text{-}j}, X_j = x^*_j, \tilde{X}_{1:(j-1)} = \tilde{x}_{1:(j-1)}, X^*_{1:(j-1)} = x^*_{1:(j-1)}\right)}{q_j(x^*_j \mid x_j)\mathbb{P}\left(X_{\text{-}j} = x_{\text{-}j}, X_j = x_j, \tilde{X}_{1:(j-1)} = \tilde{x}_{1:(j-1)}, X^*_{1:(j-1)} = x^*_{1:(j-1)}\right)}\right).$$

Upon acceptance, set $\tilde{x}_j = x^*_j$; otherwise, set $\tilde{x}_j = x_j$.

end

Return $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_p)$

At this point, it should be clear that Metropolized knockoff sampling generates exact knockoffs, a fact we formally record below.

Corollary 1. *Metropolized knockoff sampling (Metro) produces valid knockoffs.*

Proof. For the sake of the proof, let U_j be the indicator of acceptance at step j, and $Z_j = (1-U_j)X^*_j$. We will prove pairwise exchangeability jointly with the U_j's and Z_j's; marginalizing out these variables will establish the claim. For $1 \le j \le p$, let $f_j(x_j, x_{\text{-}j}, \tilde{x}_{1:(j-1)}, u_{1:(j-1)}, z_{1:(j-1)})$ be the joint density function of $(X_j, X_{\text{-}j}, \tilde{X}_{1:(j-1)}, U_{1:(j-1)}, Z_{1:(j-1)})$, in this order. We will use induction to show that the density of $(X, \tilde{X}_{1:j}, U_{1:j}, Z_{1:j})$ is symmetric in X_k and $\tilde{X}_k$ for $1 \le k \le j$. For $1 \le j \le p$,

the inductive hypothesis is that f_j is symmetric in x_k and $\tilde{x}_k$ for $1 \le k \le j-1$ (since f_j is just the density of $(X, \tilde{X}_{1:(j-1)}, U_{1:(j-1)}, Z_{1:(j-1)})$ after reordering the variables). For $1 \le j \le p$,

$$
\begin{aligned}
&\text{the density of } (X, \tilde{X}_{1:j}, U_{1:j}, Z_{1:j}) \text{ at } (x, \tilde{x}_{1:j}, u_{1:j}, z_{1:j}) \\
&= f_j(x_j, x_{\text{-}j}, \tilde{x}_{1:(j-1)}, u_{1:(j-1)}, z_{1:(j-1)}) \times \\
&\quad \Bigg[\mathbf{1}_{u_j=1} \delta(z_j - 0) q_j(\tilde{x}_j \mid x_j) \min\left(1, \frac{f_j(\tilde{x}_j, x_{\text{-}j}, \tilde{x}_{1:(j-1)}, u_{1:(j-1)}, z_{1:(j-1)}) q_j(x_j \mid \tilde{x}_j)}{f_j(x_j, x_{\text{-}j}, \tilde{x}_{1:(j-1)}, u_{1:(j-1)}, z_{1:(j-1)}) q_j(\tilde{x}_j \mid x_j)}\right) \\
&\qquad + \mathbf{1}_{u_j=0} \delta(\tilde{x}_j - x_j) q_j(z_j \mid x_j) \left(1 - \min\left(1, \frac{f_j(z_j, x_{\text{-}j}, \tilde{x}_{1:(j-1)}, u_{1:(j-1)}, z_{1:(j-1)}) q_j(x_j \mid z_j)}{f_j(x_j, x_{\text{-}j}, \tilde{x}_{1:(j-1)}, u_{1:(j-1)}, z_{1:(j-1)}) q_j(z_j \mid x_j)}\right)\right) \Bigg],
\end{aligned}
$$

which is symmetric in the first $j-1$ pairs by the inductive hypothesis. For the symmetry in the jth pair, when $u_j = 1$, the density simplifies to

$$
\begin{aligned}
\delta(z_j - 0) \times \min\big(& f_j(x_j, x_{\text{-}j}, \tilde{x}_{1:(j-1)}, u_{1:(j-1)}, z_{1:(j-1)}) q_j(\tilde{x}_j \mid x_j), \\
& f_j(\tilde{x}_j, x_{\text{-}j}, \tilde{x}_{1:(j-1)}, u_{1:(j-1)}, z_{1:(j-1)}) q_j(x_j \mid \tilde{x}_j) \big),
\end{aligned}
$$

which is invariant to swapping x_j and $\tilde{x}_j$; when $u_j = 0$, the delta function $\delta(\tilde{x}_j - x_j)$ ensures $x_j = \tilde{x}_j$, and thus swapping them has no effect. Hence, when the algorithm terminates, all pairs are exchangeable and therefore remain exchangeable after marginalizing out the U_j's and Z_j's. □

Anticipating possible future applications, we wish to remark that Metro can be easily adapted to sampling *group* knockoffs (Dai and Barber 2016); see Appendix E.

3.2 Covariance-guided proposals

Now that we have available a broad class of knockoff samplers, we turn to the question of finding faithful proposal distributions that will generate statistically powerful knockoffs. The overall challenge is to propose samples that are far away from X to make good knockoffs, but not as far that they are systematically rejected. A rejection at the jth step gives $\tilde{X}_j = X_j$, leading to a knockoff with poor contrast. Below, we shall borrow ideas from existing knockoff samplers for Gaussian models to make sensible proposals.

Suppose that X has mean μ and covariance $\mathbf{\Sigma}$, and consider $s \in \mathbb{R}^p$ with non-negative entries such that $\mathbf{\Gamma}(s)$ from (2) is positive semidefinite. Such a vector s can be found with techniques from Barber and Candès (2015) and from Candès et al. (2018). We have seen that if X were Gaussian, this covariance matrix would induce a multivariate Gaussian joint distribution over X and $\tilde{X}$ with the correct symmetry. In non-Gaussian settings, our observation is that we can still make proposals as if the variables were Gaussian, but use the MH correction to guarantee exact conditional exchangeability. This can be viewed as a Metropolis-adjustment to the second-order knockoff construction of Candès et al. (2018). Concretely, the distribution q_j for a covariance-guided proposal—used to generate a proposal X_j^*—is normal with mean

$$
\mu_j + \left(\Gamma_{12}^{(j)}\right)^\top \left(\mathbf{\Gamma}_{11}^{(j)}\right)^\dagger \left(X - \mu, X_{1:(j-1)}^* - \mu_{1:(j-1)}\right)^\top
$$

and variance

$$
\Gamma_{22}^{(j)} - \left(\Gamma_{12}^{(j)}\right)^\top \left(\mathbf{\Gamma}_{11}^{(j)}\right)^\dagger \Gamma_{12}^{(j)};
$$

here, $X_{1:(j-1)}^*$ is the sequence of already generated proposals, $\mathbf{\Gamma}_{11}^{(j)} = \mathbf{\Gamma}_{1:(p+j-1),1:(p+j-1)}$, $\Gamma_{22}^{(j)} = \Gamma_{p+j,p+j}$, $\Gamma_{12}^{(j)} = \Gamma_{1:(p+j-1),p+j}$, μ is the mean of X, and $\dagger$ stands for the pseudoinverse. The

parameters of q_j can be efficiently computed using the special structure of $\mathbf{\Gamma}$; see Appendix D. The faithfulness of the proposal is shown in Appendix B.

The covariance-guided proposals are valid even when $\mathbf{\Sigma}$ is replaced by any other positive semidefinite matrix—*any faithful proposal distribution will give valid knockoffs*. This allows us to use an empirical estimate of $\mathrm{Cov}(X)$ based on simulated samples from $\mathcal{L}(X)$, or even to apply the covariance-guided proposals to discrete distributions by rounding each proposal X_j^* to the nearest point in the support of X_j. These proposals will be most successful when X is well-approximated by a Gaussian density, indeed when X is exactly Gaussian and the true covariance is used, the covariance-guided proposals will never be rejected. Numerical simulations in a variety of settings can be found in Section 5.

3.3 Multiple-try Metropolis

A possibility for sampling $\tilde{X}_j$ "far away" from X_j is to run multiple MH steps instead of a single one. The issue with this is that this would make the conditional distributions from Metro prohibitively complex at later steps. Longer chains also require conditioning later proposals on a longer sequence of proposals and acceptances or rejections, which will constrain those proposals to be closer to their corresponding true variables and thus reduce power. Instead, we use the multiple-try Metropolis (MTM) technique introduced in Liu et al. (2000).

The key idea of MTM is to propose a set of several candidate moves in order to increase the probability of acceptance. As in Qin and Liu (2001), we take the candidate set to be $C_x^{m,t} = \{x \pm kt : 1 \le k \le m\}$, where m is a positive integer and t is a positive number; see Figure 1 for an illustration. MTM proceeds by choosing one element x^* from the set $C_x^{m,t}$, with probability proportional to the target density, i.e.,

$$\mathbb{P}(\text{select } x^* \text{ from } C_x^{m,t}) = \frac{\pi(x^*)}{\sum_{u \in C_x^{m,t}} \pi(u)}. \tag{8}$$

This proposal is then accepted with probability

$$\gamma \min\left(1, \frac{\sum_{u \in C_x^{m,t}} \pi(u)}{\sum_{v \in C_{x^*}^{m,t}} \pi(v)}\right), \quad \gamma \in (0,1), \tag{9}$$

where γ is an additional tuning parameter explained in Appendix F.2. This parameter should be taken to be near 1 in most settings. If no element of $C_x^{m,t}$ has positive probability, then one automatically rejects. MTM is a special case of MH with the proposal $q(x^* \mid x)$ distribution defined implicitly by the above rules, and furthermore, the proposals are faithful. Thus, MTM can be used in Metro.

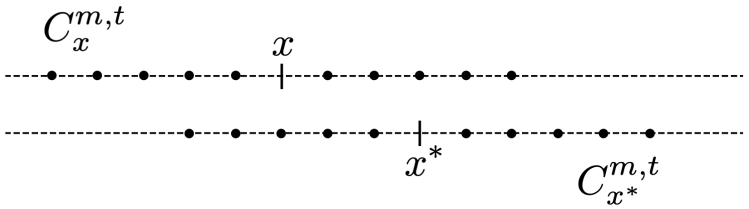

Figure 1: Multiple-try Metropolis (adapted from Figure 2 in Qin and Liu (2001)).

While there is no universally optimal combination of m and t, we provide guidance about default values based on our experimental results from Section 5. To understand the choice of parameters, first observe that with a fixed t, the knockoff distribution produced by the algorithm

should eventually stabilize as m grows to infinity, since (non-pathological) π will vanish at positive and negative infinities and equations (8) and (9) will converge. Large values of m require more density evaluations, so we would like to choose the smallest value of m such that the distribution defined by equations (8) and (9) is nearly converged to its limit as $m \to \infty$. Turning our attention to t, smaller values cause higher acceptance rates, and at the same time, encourage $\tilde{X}_j$ to be close to X_j. Clearly, there is a trade-off. Based on our experiments, a sensible default setting is $m = 4$ and $t_j = 1.5\sqrt{1/(\mathbf{\Sigma}^{-1})_{jj}}$ where $\mathbf{\Sigma} = \mathrm{Cov}(X)$. In the Gaussian case, $\mathrm{Var}(X_j|X_{\text{-}j}) = 1/(\mathbf{\Sigma}^{-1})_{jj}$ for any observed value of $X_{\text{-}j}$ (Anderson 2009), hence this choice of scaling is intuitive. In the non-Gaussian case $1/(\mathbf{\Sigma}^{-1})_{jj}$ should be viewed as an approximation to the conditional variance. We have found that this parameter setting achieves nearly the best performance in most of our experiments.

4 Graphical models and conditional independence

One outstanding issue is whether the Metropolized knockoff sampler can be run in reasonable time for cases of interest. We begin by showing why sequential knockoff sampling is prohibitively expensive without additional structure, and then turn our attention to a common type of structure that enables efficient sampling: graphical models. The central contribution of this section will be a complexity bound on Metro showing how the graphical structure affects the difficulty of sampling. To complete this line of investigation, we give a complexity lower bound for all knockoff samplers which shows that Metro is optimal in some cases.

4.1 Why do we need structure?

Consider running Metro for some input distribution $\mathbb{P}$ and sample $X = x$. In view of (7), at step j we need to evaluate $\mathbb{P}(X_{\text{-}j}, X_j = z_j, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)})$ for $z_j \in \{x_j, x^*_j\}$ up to a constant.[d] Metro defines a joint distribution on $(X, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)})$ implicitly, so the only way to evaluate this density is to compute it step by step, from 1 to $j-1$, i.e., through the sequential decomposition

$$\mathbb{P}(X_{\text{-}j}, X_j = z_j, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)}) = \mathbb{P}(X_{\text{-}j}, X_j = z_j) \times \prod_{k=1}^{j-1} \left[\mathbb{P}(\tilde{X}_k \mid X_{\text{-}j}, X_j = z_j, \tilde{X}_{1:(k-1)}, X^*_{1:k}) \mathbb{P}(X^*_k \mid X_{\text{-}j}, X_j = z_j, \tilde{X}_{1:(k-1)}, X^*_{1:(k-1)}) \right]. \quad (10)$$

Consider the term $\mathbb{P}(\tilde{X}_k \mid X_{\text{-}j}, X_j = z_j, \tilde{X}_{1:(k-1)}, X^*_{1:k})$. By the definition of Metro, computing this term will require evaluating an acceptance probability of the form

$$\min\left(1, \frac{q_k(x_k \mid x^*_k)\mathbb{P}(X_{\text{-}(j,k)}, X_k = x^*_k, X_j = z_j, \tilde{X}_{1:(k-1)}, X^*_{1:(k-1)})}{q_k(x^*_k \mid x_k)\mathbb{P}(X_{\text{-}(j,k)}, X_k = x_k, X_j = z_j, \tilde{X}_{1:(k-1)}, X^*_{1:(k-1)})}\right). \quad (11)$$

Now, to compute the terms in the acceptance probability, we must use the same sequential decomposition (10) for the terms $\mathbb{P}(X_{\text{-}(j,k)}, X_k = z_k, X_j = z_j, \tilde{X}_{1:k-1}, X^*_{1:k-1})$ for $z_k \in \{x_k, x^*_k\}$. Considering $k = j-1$, we see that step j is making two calls to the probability at step $j-1$, each of which is in turn making two calls to the probability function at step $j-2$ and so on. Thus, each evaluation of (10) will require $\Omega(2^j)$ function calls. This behavior is not due to a shortcoming of Metro; any genuine knockoff sampler with access only to an unnormalized density will require time exponential in p. We will present the formal statement of this lower bound later in Theorem 3.

[d] In this section, when not explicitly specified, a variable is set to its observed value, e.g., $\mathbb{P}(X_1 \mid X_2 = z_2, X_3, \tilde{X}_1, X^*_1)$ is shorthand for $\mathbb{P}(X_1 = x_1 \mid X_2 = z_2, X_3 = x_3, \tilde{X}_1 = \tilde{x}_1, X^*_1 = x^*_1)$.

Although knockoff sampling with no restriction on the distribution is prohibitively slow, we will show how to avoid the exponential complexity when there is additional known structure. Consider a Markov chain, i.e., a density that factors as $\mathbb{P}(x) = \prod_{j=1}^{p-1} \phi_j(x_j, x_{j+1})$. In this case, the joint density (10) can be evaluated efficiently provided we proceed along the chain in the natural order. Assume for simplicity that the proposal distribution is fixed in advance so that the second term within the square brackets in (10) does not depend on any variables and can be ignored. Due to the Markovian structure, only the $k = j - 1$ term in the product depends on z_j, so it suffices to compute the acceptance probability (11) for $k = j - 1$. Again using the Markovian structure, this simplifies to

$$\min\left(1, a_{j-1}\frac{q_{j-1}(x_{j-1} \mid x^*_{j-1})\mathbb{P}(X_{\text{-}(j,j-1)}, X_{j-1} = x^*_{j-1}, X_j = z_j)}{q_{j-1}(x^*_{j-1} \mid x_{j-1})\mathbb{P}(X_{\text{-}(j,j-1)}, X_{j-1} = x_{j-1}, X_j = z_j)}\right)$$
$$= \min\left(1, a_{j-1}\frac{q_{j-1}(x_{j-1} \mid x^*_{j-1})\phi_{j-2}(x_{j-2}, x^*_{j-1})\phi_{j-1}(x^*_{j-1}, z_j)}{q_{j-1}(x^*_{j-1} \mid x_{j-1})\phi_{j-2}(x_{j-2}, x_{j-1})\phi_{j-1}(x_{j-1}, z_j)}\right)$$

where a_{j-1} is the ratio

$$a_{j-1} = \frac{\mathbb{P}(X^*_{1:(j-2)}, \tilde{X}_{1:(j-2)} \mid X_{\text{-}(j,j-1)}, X_{j-1} = x^*_{j-1}, X_j = z_j)}{\mathbb{P}(X^*_{1:(j-2)}, \tilde{X}_{1:(j-2)} \mid X_{\text{-}(j,j-1)}, X_{j-1} = x_{j-1}, X_j = z_j)}$$

which does not depend on z_j by the Markov structure. The key here is that a_{j-1} was previously computed with $z_j = x_j$ when sampling $\tilde{X}_{j-1}$. Thus, the acceptance probability can be computed in constant time. Putting this all together, for a Markov chain, each of the necessary joint probabilities (10) can be computed in constant time, and the time to sample the entire vector $\tilde{X}$ is linear in the dimension p. Markov chains are not the only case where computing the acceptance probability can be done quickly; for other distributions with conditional independence structure, we next develop a systematic way of computing (10), using the graphical structure to control the depth of the recursion and hence control the running time.

4.2 Time complexity of Metro for graphical models

We have seen that we must restrict our attention to a subset of distributions in order to efficiently sample knockoffs, so in this section we show how to implement Metropolized knockoff sampling for a very broad class of distributions: graphical models. Let $X \in \mathbb{R}^p$ be a random vector whose density factors over a graph G:

$$\mathbb{P}(x) \propto \Phi(x) = \prod_{c \in C} \phi_c(x_c); \tag{12}$$

here, C is the set of maximal cliques of the graph G and Φ is an unnormalized version of $\mathbb{P}$. The variables in X can be either discrete or continuous. All graphical models with positive density or mass take this form (Hammersley and Clifford 1971), and such distributions are known to have particular conditional independence properties. We refer the reader to Koller and Friedman (2009) for a general treatment.

In order to take advantage of the conditional independence structure of X, we use a graph-theoretic object known as a *junction tree* (Bertele and Brioschi 1972) which encodes properties of the graph G.

Definition 1 (Junction tree)**.** *Let T be a tree with vertices that are subsets of the nodes $\{1, \ldots, p\}$ of a graph G. T is a* junction tree *for G if the following hold:*

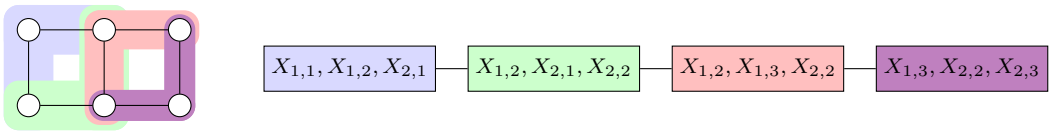

Figure 2: A junction tree of treewidth 2 for the 2×3 grid, which happens to be a chain.

1. *Each $j \in \{1, \ldots, p\}$ appears in some vertex V of T.*
2. *For every edge (j, k) in G, $j \in V$ and $k \in V$ for some vertex V.*
3. (Running intersection property) *If the vertices V and V' both contain a node of G, then every vertex in the unique path from V to V' also contains this node.*

Figure 2 gives an example of a junction tree over a 2×3 grid. The size of the largest vertex of T minus one is known as the *width* of the junction tree T, and the smallest width of a junction tree over G is called the *treewidth* of G, a measure of graph complexity. Finding the junction tree of lowest width for a graph G is known to be NP-hard (Arnborg et al. 1987), but there exist efficient heuristic algorithms for finding a junction tree with small width (Kjærulff 1990; Koller and Friedman 2009).

Given a junction tree T for the graph G, we will soon prove that Metro can be run with $O(p2^w)$ queries of the unnormalized density Φ, where w is the width of T. In view of (7), at step j of Metro we need to evaluate $\mathbb{P}(X_{\text{-}j}, X_j = z_j, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)})$ for $z_j \in \{x_j, x_j^*\}$ up to a constant as well as sample from and evaluate the proposal distribution $q_j(\cdot \mid x_j)$. We can use the graphical model structure to make these operations tractable by both (1) sampling the variables in a specific order, and (2) choosing proposal distributions that are not unnecessarily complex. We formalize these two requirements below.

We first consider the order in which we sample the variables. Recalling (10), the complexity of the computations of $\mathbb{P}(X_{\text{-}j}, X_j = z_j, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)})$ depends on the number of function calls implied by the recursion (10). For simplicity, assume that the proposal terms in the product, $\mathbb{P}(X^*_k \mid X_{\text{-}j}, X_j = z_j, \tilde{X}_{1:(k-1)}, X^*_{1:(k-1)})$, never depends on z_j; this will be relaxed soon. In that case, we need only consider the terms in (10) of the form $\mathbb{P}(\tilde{X}_k \mid X_{\text{-}j}, X_j = z_j, \tilde{X}_{1:(k-1)}, X^*_{1:k})$ for $k < j$. When there is graphical structure, not all such terms will depend on z_j, and the number of terms that do depend on z_j determines the recursion depth. In particular, if at step j only r_j terms depend on z_j, then there will be $O(2^{r_j})$ function calls in the recursion. A desirable ordering of the variables is then one that minimizes the largest r_j, and such an ordering can be extracted from a junction tree T using Algorithm 2.

Algorithm 2: Junction tree variable ordering for Metro

Initialize tree $T_{\text{active}} = T$ and list $J = \{\}$.
while $T_{\text{active}} \neq \emptyset$ **do**
  Select a leaf node V of T_{active}. V is connected to at most one other node V' of T_{active} because it is a tree.
  In any order, append each $j \in V \setminus V'$ to the end of the list J. If no V' exists, append all $j \in V$ to J in any order.
  Remove V from the active tree T_{active}.
end
Return J

Algorithm 2 is valid in that when a node is removed, no $j \in J$ remains in any node in T_{active}.[e]

[e]This simple fact follows from the running intersection property; we refer the reader to Lemma 1 in Appendix A.

From now on we assume that the variables are numbered according to this ordering. Our second consideration is to create proposals that do not add unnecessary complexity. No matter which proposal distribution we choose, $\mathbb{P}(X_j = z_j \mid X_{\text{-}j}, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)})$ will still depend on some X_ℓ for $\ell > j$ due to dependencies among coordinates of X; we however restrict ourselves to proposal distributions that do not add any additional dependencies.

Definition 2 (Compatible proposal distributions)**.** *Let V_j be the node of the junction tree when j is appended to J from Algorithm 2. Set $\bar{V}_j = \{1, \ldots, j-1\} \cup V_j$. We say that proposal distributions q_j are* compatible *with a junction tree T if they depend only on $X_{\bar{V}_j}, \tilde{X}_{1:(j-1)}$, and $X^*_{1:(j-1)}$.*

This definition is motivated by the property

$$X_{1:j} \perp\!\!\!\perp X_{\bar{V}_j^c} \mid X_{\bar{V}_j \setminus \{1,\ldots,j\}},$$

since $\bar{V}_j \setminus \{1, \ldots, j\}$ separates $\{1, \ldots, j\}$ from $\bar{V}_j^c$ in the graph G. Thus, a proposal distribution at step j that violates the compatibility property and relies on X_ℓ for some $\ell \notin \bar{V}_j$ will result in additional non-one terms in the product in (10) at step ℓ, so $\bar{V}_j$ is the largest set that the proposal can be allowed to depend on without increasing the number of function calls/runtime. Although not all proposals are compatible, it is a rich enough class to handle a broad range of knockoff distributions, including the distribution induced by SCIP.

With these two conditions in place, we now state our main result about the efficiency of knockoff sampling, giving an upper bound on the number of evaluations of the unnormalized density function Φ that is required by Metro when the graphical structure is known. Assuming the variable ordering from Algorithm 2 and faithful proposal distributions compatible for T such that sampling from and evaluating the proposal distributions does not require evaluating Φ, we reach the following result:

Theorem 2 (Computational efficiency of Metro)**.** *Let X be a random vector with a density which factors over a graph G as in* (12)*. Let T be a junction tree of width w for the graph G. Under the conditions above, Metro uses $O(p2^w)$ queries of Φ.*

This result means that we can efficiently implement Metropolized knockoff sampling for many interesting distributions, and it shows precisely how the complexity of the conditional independence structure of X affects the complexity of the sampling algorithm. Furthermore, in the next section we will prove that this is the optimal complexity in some cases.

4.3 Time complexity of general knockoff sampling

In the previous section we analyzed the runtime of Metro and showed that it will be tractable for graphs of sufficiently low treewidth. Now, we investigate the computational complexity of knockoff sampling in general. To formalize our investigation, we discuss a model of computation in which we have no information about the distribution of X beyond its graphical structure and the ability to query its (possibly unnormalized) density at any given point.

Oracle model. In this model, we are given as inputs (a) a p-dimensional vector X drawn from a density $\lambda\Phi$, where λ is a (possibly unknown) positive scalar so that we can think of Φ as an unnormalized density, (b) the support of Φ, and (c) a black box capable of evaluating Φ at arbitrary query points, and (d) a graph G for which the density is known to have the form (12). No other information about Φ is available.

We show that in the oracle model with the complete graph, i.e., when there is no graphical structure, knockoff sampling requires exponential time in the number of covariates, p. Please note

that any complexity bound must take into account the quality of the generated knockoffs since $\tilde{X} = X$ is a trivial knockoff that can be sampled in no time.

Theorem 3 (Complexity lower bound for knockoff sampling)**.** *Consider a procedure operating in the oracle model which makes a finite number of calls to the black box Φ and returns $\tilde{X}$, thereby inducing a joint distribution $(X, \tilde{X})$ obeying the pairwise exchangeability* (1) *for all Φ. If G is the complete graph so that the procedure generates valid knockoffs for any input density, then the total number N of queries of Φ must obey $N \geq 2^{\#\{j:X_j \neq \tilde{X}_j\}} - 1$ a.s..*

This result means that for any knockoff sampler, we cannot have both full generality and time efficiency. Put differently, in order to efficiently generate nontrivial knockoffs, we will need to restrict our attention to a subset of distributions for which we have structure. This fact justifies our decision to focus on distributions with graphical structure. We also derive a lower bound for the complexity of knockoff sampling for graphical models, stated next.

Corollary 2 (Complexity lower bound for graphical models)**.** *Consider the setting of Theorem 3. Fix a graph G with maximal cliques C. Suppose that for all Φ of the form $\Phi(x) = \prod_{c \in C} \phi_c(x_c)$, the procedure induces a joint distribution $(X, \tilde{X})$ obeying pairwise exchangeability* (1)*. Then $N \geq \max_{c \in C} 2^{\#\{j \in c: X_j \neq \tilde{X}_j\}} - 1$ a.s..*

This proposition shows that even after making some useful structural assumptions, there is still a trade-off between knockoff quality and computation. We next derive a byproduct, which proves that Metro is achieving a good runtime.

Proposition 2 (Optimality of Metro for chordal Gaussian graphical models)**.** *Consider continuous distributions of the form $\Phi(x) = \prod_{c \in C} \phi_c(x_c)$ over a chordal graph G.*[f] *On the one hand, for any input, Metro can be run with $O(p^2 + p2^w)$ queries of Φ. Furthermore, in the case where the distribution is Gaussian with zero mean and positive definite covariance (i.e., $\Phi(x) \propto \exp\left(-x\boldsymbol{\Sigma}^{-1}x^\top/2\right)$), Metro can produce knockoffs with $X_j \neq \tilde{X}_j$ for all j with probability 1. On the other hand, any general procedure that samples knockoffs such that $X_j \neq \tilde{X}_j$ for all j with probability $\epsilon > 0$ will require at least $2^w - 1$ queries of Φ with probability at least ϵ.*

Proposition 2 means that for chordal graphs, any general knockoff sampling algorithm such that $\mathbb{P}(X_j \neq \tilde{X}_j \text{ for all } j)$ is bounded away from zero needs, in expectation, the same exponential order of queries as Metro (with the proviso that p is negligible compared to 2^w).

4.4 Divide-and-conquer to reduce treewidth

Theorem 2 shows that Metro enables efficient computations for random vectors whose densities factor over a graph G of low treewidth. Not all graphs corresponding to random vectors of interest have low treewidth, however. A $d_1 \times d_2$ grid, for example, has treewidth $\min(d_1, d_2)$ (Diestel 2018). This section develops a mechanism for simplifying the graphical structure of a random vector X, allowing for faster computation of exact knockoffs at the cost of reduced knockoff quality.

To simplify graphical structure, we fix a set of variables C that separates the graph G into two subgraphs A and B. After fixing the variables in C, knockoffs can be constructed for the variables in A and B independently.

Proposition 3 (Validity of divide-and-conquer knockoffs)**.** *Suppose the sets A, B, C form a partition of $\{1, \ldots, p\}$ such that C separates A and B in the graph G, i.e., there is no path from some $j \in A$*

[f] A chordal graph is a graph such that any cycle of length 4 or larger has a chord.

Figure 3: Two examples of conditioning to reduce the treewidth of a 6×6 grid from 6 to 3.

to some $k \in B$ in G that does not contain some $\ell \in C$. Suppose $\tilde{X}$ is a random vector such that $X_C = \tilde{X}_C$ a.s. and for all $j_A \in A$ and $j_B \in B$,

$$(X_D, \tilde{X}_D) \stackrel{d}{=} (X_D, \tilde{X}_D)_{\text{swap}(j_D)} \mid X_C \qquad \text{for} \quad D = A, B.$$

Furthermore, assume we construct the knockoffs for A and B separately, i.e. $(X_A, \tilde{X}_A) \perp\!\!\!\perp (X_B, \tilde{X}_B) \mid X_C$. Then $\tilde{X}$ is a valid knockoff.

The divide-and-conquer technique can be applied recursively to split the graph into components of low treewidth until the junction-tree algorithm for constructing knockoffs can be used on each component. For example, for an arbitrary planar graph with p nodes, the planar separator theorem gives the existence of a subset of nodes C of size $O(\sqrt{p})$ that separates the graph into components A and B with $\max(|A|, |B|) \leq 2p/3$ (Lipton and Tarjan 1979), suggesting that this technique will apply to many cases of interest. Figure 3 illustrates this technique for a $d_1 \times d_1$ grid. We split the grid into rectangular ribbons of size $d_1 \times d_2$ for small d_2; each resulting ribbon has treewidth d_2.

The drawback of this approach is that for $j \in C$, we shall have $X_j = \tilde{X}_j$. When we think of deploying the knockoff framework in statistical applications, one should remember that we will work with multiple copies of X corresponding to distinct observations. We can then choose different separator sets for each observation so that in the end, $X_j \neq \tilde{X}_j$ for most of the observations. For example, in the setting of Figure 3, one would randomly choose between the two choices of C for each observation. This technique is explored numerically in Section 5.2.3.

4.5 Discrete distributions

For discrete distributions with a small number of states for each coordinate X_j, the junction tree techniques from Section 4.2 can be directly applied without using Metropolized knockoff sampling. When each variable X_j can take on at most K values, the probability mass function $\mathbb{P}(X_j \mid X_{\text{-}j}, \tilde{X}_{1:(j-1)})$ can be represented as a vector in $\mathbb{R}^K$, so at step j of the algorithm we simply need to evaluate $\mathbb{P}(X_j = z_j, X_{\text{-}j}, \tilde{X}_{1:(j-1)})$ for z_j in the support of X_j. This is the same quantity we computed in Section 4.2; see, e.g., (10). Once these probabilities have been computed, sampling from the resulting multinomial probability gives the SCIP procedure. In principle, this can be viewed as a special case of Metro, but for a practical implementation it is simpler to work directly with the probability vectors. A similar analysis to the proof of Theorem 2 then shows that the procedure requires $O(pK^w)$ queries of the density Φ; see Appendix C.5 for details. For discrete distributions with infinite or large K, this is not tractable. However, Metro still applies and is much faster.

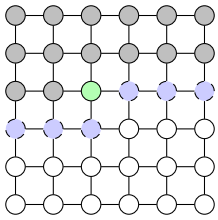

Figure 4: An illustration of sampling knockoffs for an Ising model on a grid from Section 4.6. The blue dashed nodes represent the active variables of the junction tree when variable $X_{3,3}$ (shown in green) is being sampled. Gray nodes indicate variables that have already been sampled, and white nodes indicate variables that have not been sampled yet and are not in the active node of the junction tree.

4.6 Knockoffs for the Ising model

The tools from this section have the power to generate knockoffs for the Ising model on a grid (3). To construct an efficient knockoff sampler for this distribution, we need to find a junction tree of minimal width for the $d_1 \times d_2$ grid so that we can apply the technique from Section 4.5. A junction tree for the 2×3 grid of width 2 is shown in Figure 2, and the construction immediately generalizes to a junction tree of width $\min(d_1, d_2)$ for the $d_1 \times d_2$ grid, which is the optimal width. When $d_1 \geq d_2$, this leads to a knockoff sampler that proceeds from left to right, top to bottom; when variable $X_{i,j}$ is sampled, the other variables in the active node of the junction tree are $X_{i,j+1:p}$ and $X_{i+1,1:j}$; see Figure 4. Per our upper bound, this knockoff sampler will have runtime $O(d_1 d_2 2^{\min(d_1,d_2)})$. If $\min(d_1, d_2)$ is large, this runtime may still be prohibitively long, but the divide-and-conquer technique from Section 4.4 greatly increases speed at the cost of slightly worse knockoffs than the impractical full procedure. We conduct a simulation experiment of both the small-grid and large-grid setting in Section 5.2.3.

5 Numerical experiments

We now empirically examine the Metropolized knockoff sampler, beginning with the few models where previously known samplers are available as a baseline, and then continuing on to cases with no previously known samplers. Condensed plots are presented in the main text, while more comprehensive versions can be found in Appendix F. We provide approximate runtimes with a single-core[g] implementation in either R or Python. All source code is available from `https://github.com/wenshuow/metro` with interactive computing notebooks at `http://web.stanford.edu/group/candes/metro` demonstrating the usage of the code and presenting further experimental results.

Measuring knockoff quality

The *mean absolute correlation* (MAC) is a useful measure of knockoff quality for a joint distribution of $(X, \tilde{X})$:

$$\mathrm{MAC}(\mathcal{L}(X, \tilde{X})) := \frac{1}{p} \sum_{j=1}^{p} |\mathrm{cor}(X_j, \tilde{X}_j)|. \tag{13}$$

We will use this as our measure of knockoff quality in our simulation experiments. Lower values of MAC are preferred. Let $\mathbf{\Gamma}$ be the correlation matrix of $(X, \tilde{X})$; pairwise exchangeability implies $\mathbf{\Gamma}$

[g]The hardware varies across simulations, but each CPU is between 2.5Ghz and 3.3Ghz.

is of the form (2). The MAC is then $\frac{1}{p}\sum_{j=1}^{p}|1-s_j|$. Since $\mathbf{\Gamma}=\mathbf{\Gamma}(s)$ has to be positive semidefinite, a lower bound on the MAC achievable by any knockoff-generation algorithm for a given distribution is the optimal value of the program

$$\min_{s}\frac{1}{p}\sum_{j=1}^{p}|1-s_j|,\ \text{subject to}\ \mathbf{\Gamma}(s)\succeq 0. \tag{14}$$

This minimization problem can be solved efficiently with semidefinite programming (Barber and Candès 2015); we call the solution the *SDP lower bound* for the MAC. This lower bound can be achieved for Gaussian distributions (Candès et al. 2018). Valid knockoffs, however, must match *all* moments, not just the second moments, so this lower bound is not expected to be achievable in general; still it provides a useful goalpost in our simulations.

5.1 Models with previously known knockoff samplers

5.1.1 Gaussian Markov chains

We first apply our algorithm to Gaussian Markov chains and compare with the SDP Gaussian knockoffs, whose MAC achieves the SDP lower bound exactly, and SCIP knockoffs, both from Candès et al. (2018). We take $p=500$ features such that $X_1\sim\mathcal{N}(0,1)$ and $X_{j+1}\mid X_{1:j}\sim\mathcal{N}(\rho_j X_j, 1-\rho_j^2)$. First, since the model is multivariate Gaussian, the covariance-guided proposal with s computed by the SDP method (14) will be identical to the SDP Gaussian knockoffs, so already a clever implementation of Metro is as good as a method specifically designed for Gaussian distributions, and since both achieve the SDP lower bound, one cannot do better in terms of MAC. Thus, we only investigate the MTM-proposals for implementing Metro. Note that the Gaussian knockoffs from Candès et al. (2018) do not use the Markovian structure of this problem, but instead rely on operations on $2p\times 2p$ matrices, whereas the MTM knockoffs from this work utilize the Markovian structure to achieve time complexity linear in p.

The results are presented in Figure 5. Following Section 3.3, we vary the number of proposals and the step size. We find that choosing the step size for X_j to be proportional to $\sqrt{1/(\mathbf{\Sigma}^{-1})_{jj}}$ gives consistent results across different sets of ρ_j's. The MTM consistently outperforms the SCIP procedure, and is reasonably close to the SDP procedure. It is observed that the defaults from Section 3.3 of eight proposals ($m=4$) and $t_j=1.5\sqrt{1/(\mathbf{\Sigma}^{-1})_{jj}}$ performs nearly the best in all settings. Confirming our reasoning in Section 3.3, we find that the performance stabilizes as m grows and the step size should not be too large or too small, although for sufficiently large m the MAC is fairly stable to the choice of t. In this setting, it takes around 1 second for MTM to sample one knockoff vector with $m=4$ and $t_j=1.5\sqrt{1/(\mathbf{\Sigma}^{-1})_{jj}}$.

5.1.2 Discrete Markov chains

For discrete Markov Chains there is one previously-known knockoff sampler, which is an implementation of the SCIP procedure (Sesia et al. 2018). We consider here Metro with MTM proposals. (The covariance-guided proposals would require ad-hoc rounding so we do not consider this here.) We take a simple Markov Chain with $K\in\{5,10\}$ states with uniform initial distribution and transition probabilities $Q(j,j')$ defined as

$$Q(j,j')=\frac{(1-\alpha)^{|j-j'|}}{\sum_{\ell=1}^{K}(1-\alpha)^{|j-\ell|}}. \tag{15}$$

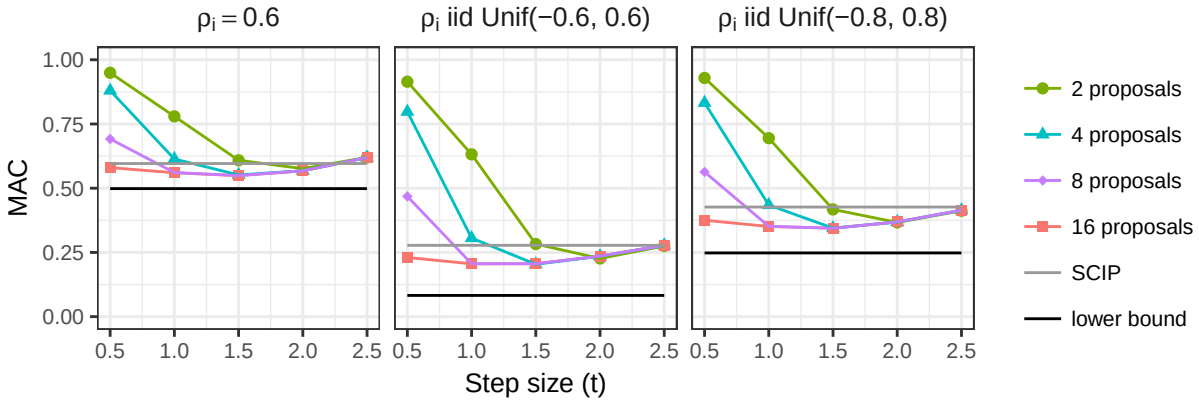

Figure 5: Simulation results for Gaussian Markov chains. The unit of step sizes is $\sqrt{1/(\boldsymbol{\Sigma}^{-1})_{jj}}$. All standard errors are below 0.001. In this case, the lower bound is achieved by the SDP Gaussian knockoffs, or equivalently, the covariance-guided proposal with an s given by the SDP (14).

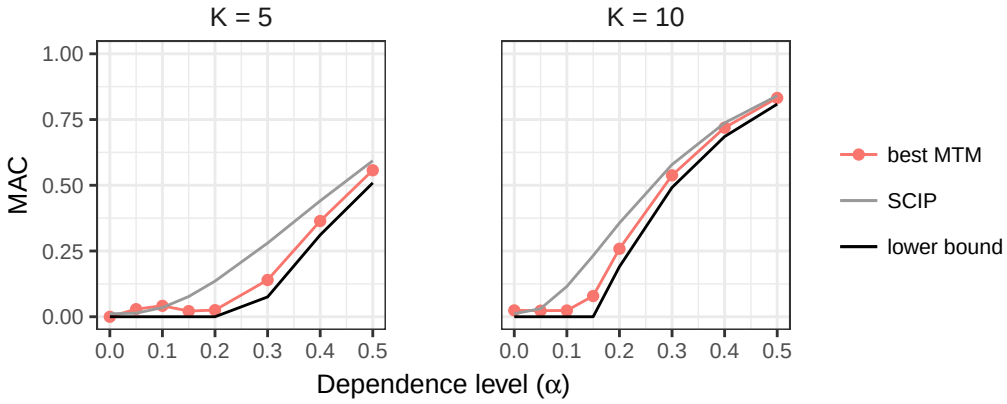

Figure 6: A comparison of the MTM procedure for discrete Markov chains with SCIP and the SDP lower bound. All standard errors are below 0.002.

We examine α from 0 (independent coordinates) to 0.5 (strong dependence between adjacent coordinates), with $p = 500$ features.

We examine the MTM methods across a range of values of the tuning parameters, and the results are presented in Figure 6. Full simulation results are given in Appendix F. Note that the cases with $K = 5$ and $\alpha \leq 0.15$ are tuned with the additional parameter γ from (9), as detailed in Appendix F.2. We find that the best-tuned MTM method outperforms the SCIP method and achieves MAC near the lower bound for all dependence levels α. It takes around 0.5 seconds and 0.7 seconds respectively, to run MTM ($m = 4$ and $t = 1$) for $K = 5$ and $K = 10$.

5.2 Models with no previously-known knockoff sampler

5.2.1 Heavy-tailed Markov chains

As an example of a heavy-tailed distribution, we consider a Markov chain with t-distributed tails.

$$X_1 = \sqrt{\frac{\nu-2}{\nu}} Z_1, \quad X_{j+1} = \rho_j X_j + \sqrt{1-\rho_j^2}\sqrt{\frac{\nu-2}{\nu}} Z_{j+1}, \quad Z_j \overset{\text{i.i.d.}}{\sim} t_\nu, \tag{16}$$

for $j = 1, \dots, p = 500$ where t_ν represents the Student's t-distribution with $\nu > 2$ degrees of freedom (note this is not a multivariate t-distribution). We try both the covariance-guided proposal with s provided by the SDP method (14) and the MTM proposals. We set $\nu = 5$ and use the same ρ_j's as in the Gaussian setting. As in Section 5.1.1, a step size of $1.5\sqrt{1/(\mathbf{\Sigma}^{-1})_{jj}}$ again performs well. The covariance-guided proposals also perform well, although unlike the Gaussian case, there is now a gap between the lower bound and the performance of the covariance-guided proposals. In this setting, it takes around 1.6 seconds for MTM to sample one knockoff vector with $m = 4$ (eight proposals) and $t_j = 1.5\sqrt{1/(\mathbf{\Sigma}^{-1})_{jj}}$. For the covariance-guided proposals, it takes around 12.5 seconds for the one-time computation of the parameters (excluding time used for computing s, which varies depending on the method) and then 0.3 seconds to sample each knockoff vector.

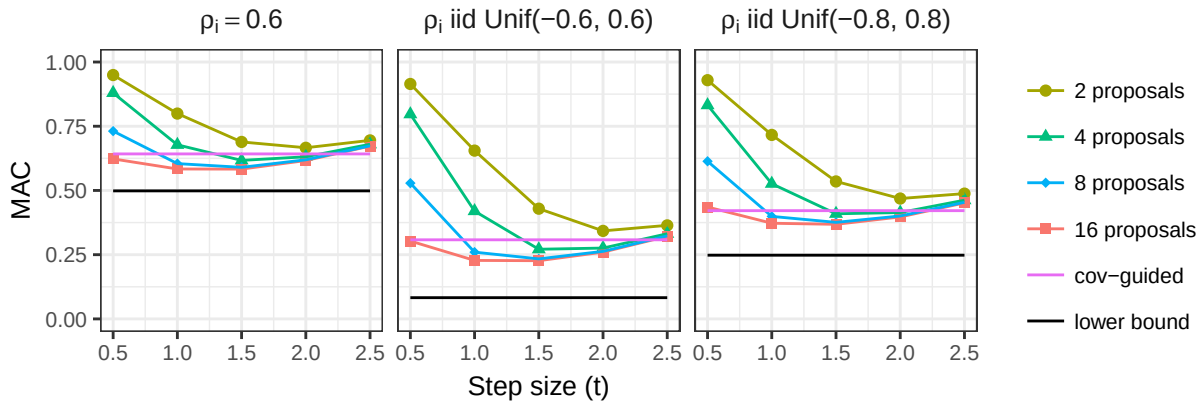

Figure 7: Simulation results for the t-distributed Markov chains. The unit of step sizes is $\sqrt{1/(\mathbf{\Sigma}^{-1})_{jj}}$. All standard errors are below 0.001.

5.2.2 Asymmetric Markov chains

As an example of asymmetric, continuous distributions, we take a standardized equal mixture of Gaussian and exponential random variables and then form a Markov chain. Explicitly,

$$Z_j \overset{\text{i.i.d.}}{\sim} \frac{I \cdot |\, Y_{\text{G}} \,| - (1-I) \cdot Y_{\text{E}} - \mu}{\sigma} \quad \text{for } j = 1, \dots, p = 500,$$

where $Y_{\text{G}} \sim \mathcal{N}(0,1)$, $Y_{\text{E}} \sim \text{Expo}(1)$ and $I \sim \text{Bern}(1/2)$ are independent. The parameters μ and σ are chosen so that Z_j has mean 0 and variance 1. We then take

$$X_1 = Z_1, \quad X_{j+1} = \rho_j X_j + \sqrt{1-\rho_j^2} Z_{j+1} \quad \text{for } j = 2, \dots, p.$$

We examine both the covariance-guided proposal with s provided by the SDP (14) and the multiple-try proposals. We use the same ρ_j's as in the Gaussian setting. As in the previous case, $m = 4$

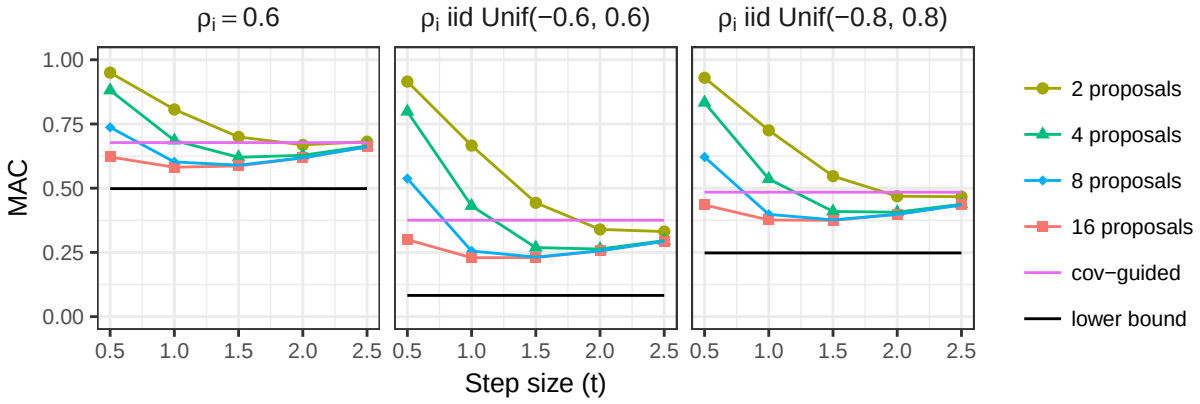

Figure 8: Simulation results for the asymmetric Markov chains. The unit of step sizes is $\sqrt{1/(\mathbf{\Sigma}^{-1})_{jj}}$. All standard errors are below 0.001.

(eight proposals) and $t_j = 1.5\sqrt{1/(\mathbf{\Sigma}^{-1})_{jj}}$ performs essentially as well as any other MTM parameter choices, and significantly outperforms the covariance-guided proposals. The timing results are the same as in the heavy-tailed Markov chains.

5.2.3 Ising model

In this section, we consider an Ising model over a square grid (3). We generate knockoffs with the method for discrete random variables from Section 4.5 combined with the divide-and-conquer technique, the combination of which was described for Ising models in Section 4.6; no other exact knockoff samplers are known for the Ising model. Although our sampling procedures for the Ising model do not explicitly use the Metropolis–Hastings step, as explained in Section 4.5, we will refer to the sampler as "Metro" in this section for simplicity.

First, we take a 10×10 grid and set all $\beta_{i,j,i',j'} = \beta_0$ and all $\alpha_{i,j} = 0$. The results are presented in Figure 9. The left panel shows how the MAC increases—or, the quality decreases— as the dependence between adjacent variables—β_0—increases. We see that the procedure is close to the lower bound for large β_0. In the middle panel, we plot $\mathrm{cor}(X_{j,k}, \tilde{X}_{j,k})$ across different coordinates (j, k). We see that on the edges of the grid, especially on the corners, knockoffs have lower correlation with their original counterparts. These variables are less determined by the values of the rest of the grid, so this is expected. In this setting, it takes about 12 seconds to sample a knockoff.

Next, we demonstrate the divide-and-conquer technique from Section 4.4. Here we consider the Ising model from above on a 100×100 grid, for a total dimension of $10,000$. The 100×100 grid has treewidth 100, so Metro would not be tractable without the the divide-and-conquer technique. We divide the graph into subgraphs of width w, by fixing entire columns as in Figure 3. To measure the effect of the slicing, we compute the MAC on the interior points and compare this to the MAC of the interior points of a smaller grid for a procedure without slicing, see Appendix F.3 for details. We find that the quality of the knockoffs increases as we take larger slices, as expected. Furthermore, even modest values of w such as $w = 5$ result in a procedure that achieves a MAC close to that of the baseline. Recall that the complexity of Metro scales as 2^w, so fixing $w = 5$ dramatically reduces the computation time compared to $w = 100$. With $w = 5$, it takes about 2.5 minutes to generate one knockoff for the 100×100 grid.

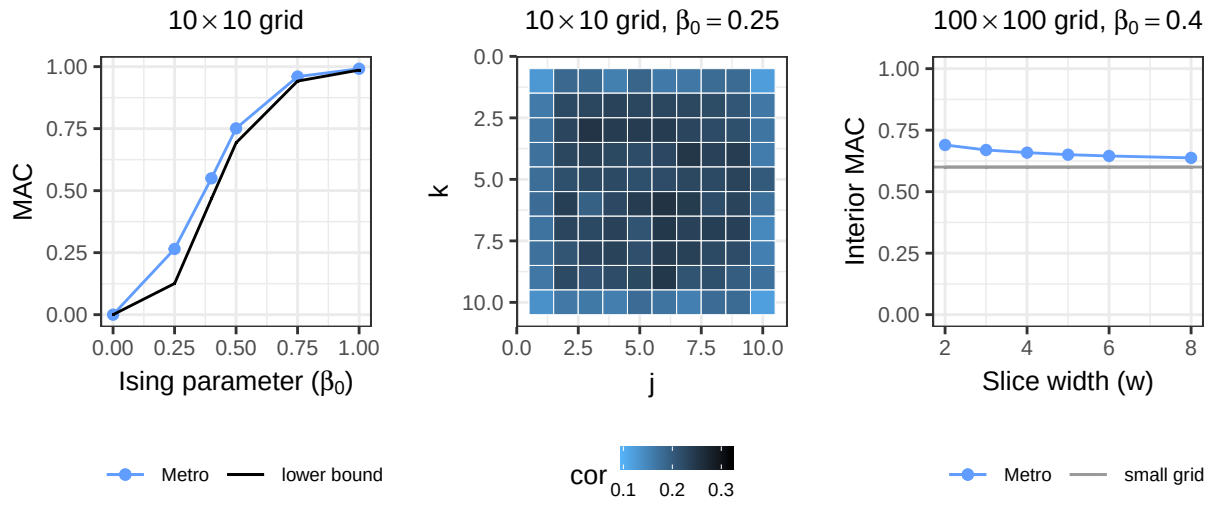

Figure 9: Results of the Ising model experiments. All standard errors in the line plots are less than 0.005.

5.2.4 Gibbs measure on a grid

Lastly, we demonstrate the MTM proposals simultaneously with the junction tree techniques for complex dependence structure. Consider a Gibbs measure on $\{1, \dots, K\}^{d \times d}$, with a probability mass function

$$\mathbb{P}(X) = \frac{1}{Z(\beta_0)} \exp\left(-\beta_0 \sum_{\substack{s,t \in \mathcal{I} \\ \|s-t\|_1 = 1}} (x_s - x_t)^2 \right), \quad \mathcal{I} = \{(i_1, i_2) : 1 \le i_1, i_2 \le d\},$$

and note that like the Ising model, this density factors over the grid. For our experiment, we take a 10×10 grid and examine different dependence levels β_0 with $K = 20$ possible states for each variable. We apply Metro with the MTM proposals and the divide-and-conquer technique on the grid, tuning the procedure across a range of parameters as detailed in Appendix F. The condensed results are given in Figure 10. We do not know of another knockoff sampler in this setting. Having said this, we observe that our procedure has MAC close to the lower bound. We also observe that in the case where $w = 3$, with as few as two proposals, our procedure performs well and takes about half a second to generate a knockoff copy; when we increase the number of proposals to ten, the compute time is around 2 minutes. When w is set to 5, the slowest setting is $m = t = 1$, which takes less than 4 minutes.

6 Discussion

This paper introduced a sequential characterization of all valid knockoff-generating procedures and used it along with ideas from MCMC and graphical models to create Metropolized knockoff sampling, an algorithm which generates valid knockoffs in complete generality with access only to X's unnormalized density. Although we proved in Theorem 3 that no algorithm (including Metro) can sample exact knockoffs *efficiently* for arbitrary X distributions, we characterized one way out of this impossibility result: conditional independence structure in X. An interesting future direction

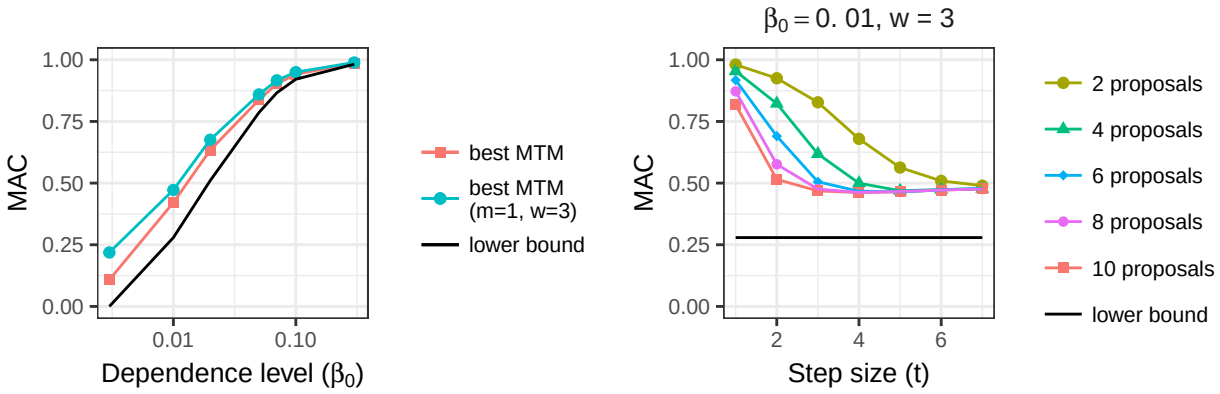

Figure 10: Results of the Gibbs measure experiments. All standard errors are below 0.002. In the left panel, β_0 is shown in logarithmic scale.

would be to establish other sufficient conditions on a model family that would allow one to sample knockoffs efficiently. Another way out of the lower bound in Theorem 3 is to forgo exact knockoffs and settle for approximations. Although this arguably is a tall order, it would be interesting to establish theoretical guarantees on the approximation quality of these or other approximate knockoff constructions, and better understand the tradeoff between knockoff approximation quality and time complexity.

Acknowledgements

E. C. was partially supported by the Office of Naval Research under grant N00014-16- 1-2712, by the National Science Foundation via DMS 1712800, and by a generous gift from TwoSigma. S. B. was supported by a Ric Weiland Graduate Fellowship. S. B. and E. C. would like to thank Yaniv Romano and Matteo Sesia for useful comments on an early version of this work. L. J. and W. W. would like to thank Jun Liu for fruitful discussions on MCMC.

References

Anderson, T. (2009). *An Introduction to Multivariate Statistical Analysis, 3rd edition*. Wiley India Pvt. Limited.

Arnborg, S., Corneil, D. G., and Proskurowski, A. (1987). Complexity of finding embeddings in a k-tree. *SIAM Journal on Algebraic Discrete Methods*, 8(2):277–284.

Barber, R. F. and Candès, E. J. (2015). Controlling the false discovery rate via knockoffs. *The Annals of Statistics*, 43(5):2055–2085.

Bertele, U. and Brioschi, F. (1972). *Nonserial Dynamic Programming*. Academic Press, Inc., Orlando, FL, USA.

Candès, E., Fan, Y., Janson, L., and Lv, J. (2018). Panning for gold: Model-X knockoffs for high-dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B*, 80(3):551–577.

Dai, R. and Barber, R. (2016). The knockoff filter for FDR control in group-sparse and multitask regression. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1851–1859. PMLR.

Diestel, R. (2018). *Graph Theory*. Springer Publishing Company, Incorporated, 5th edition.

Fan, Y., Lv, J., Sharifvaghefi, M., and Uematsu, Y. (2018). IPAD: stable interpretable forecasting with knockoffs inference. *Available at SSRN 3245137*.

Gao, C., Sun, H., Wang, T., Tang, M., Bohnen, N. I., Müller, M. L. T. M., Herman, T., Giladi, N., Kalinin, A., Spino, C., Dauer, W., Hausdorff, J. M., and Dinov, I. D. (2018). Model-based and model-free machine learning techniques for diagnostic prediction and classification of clinical outcomes in Parkinson's disease. *Scientific Reports*, 8(1):7129.

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

Hammersley, J. M. and Clifford, P. E. (1971). Markov random fields on finite graphs and lattices. Unpublished manuscript.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Ising, E. (1925). Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik*, 31(1):253–258.

Jordon, J., Yoon, J., and van der Schaar, M. (2019). KnockoffGAN: Generating knockoffs for feature selection using generative adversarial networks. In *International Conference on Learning Representations*.

Kjærulff, U. (1990). Triangulation of graphs – algorithms giving small total state space. Technical report.

Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. Adaptive computation and machine learning. MIT Press.

Lipton, R. J. and Tarjan, R. E. (1979). A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189.

Liu, J. S., Liang, F., and Wong, W. H. (2000). The multiple-try method and local optimization in Metropolis sampling. *Journal of the American Statistical Association*, 95(449):121–134.

Liu, Y. and Zheng, C. (2018). Auto-encoding knockoff generator for FDR controlled variable selection. *arXiv preprint arXiv:1809.10765*.

Lu, Y., Fan, Y., Lv, J., and Noble, W. S. (2018). DeepPINK: reproducible feature selection in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 8689–8699.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

Qin, Z. S. and Liu, J. S. (2001). Multipoint Metropolis method with application to hybrid Monte Carlo. *Journal of Computational Physics*, 172(2):827–840.

Romano, Y., Sesia, M., and Candès, E. (2018). Deep knockoffs. *arXiv preprint arXiv:1811.06687*.

Sesia, M., Sabatti, C., and Candès, E. J. (2018). Gene hunting with hidden Markov model knockoffs. *Biometrika*.

Xiao, Y., Angulo, M., Friedman, J., Waldor, M., Weiss, S., and Liu, Y.-Y. (2017). Mapping the ecological networks of microbial communities from steady-state data. *bioRxiv*, 8:150649.

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

A Junction tree lemmas

This section includes several important lemmas which will be used in some proofs in the appendix.

Lemma 1. *If the variables are ordered by Algorithm 2, then for each $1 \leq j \leq p$, any node in the junction tree that contains j is an element of the set $\{V_1, V_2, \ldots, V_j\}$. In addition, if $j \in V_k$ for some $k > j$, then $V_k = V_j$.*

Proof of Lemma 1. According to Algorithm 2, when a node V is selected, all variables in $V \setminus V'$—here, V' is the unique neighbor of V in the remaining junction tree—are sampled before the next node is selected. Recall that V_j is the selected node when j is sampled; let $k \geq j$ be the last sampled variable when V_j is selected (in this case, we have $V_j = V_{j+1} = \cdots = V_k$ by definition). After k is sampled, V_j is removed, and by Algorithm 2, this means $j, j+1, \ldots, k$ do not appear in V_j's only remaining neighbor. Now we claim no remaining node contains j. Otherwise, if there is a node V_j^* which contains j and still remains, by the running intersection property, all nodes on the unique path between V_j and V_j^* contain j. This would imply that V_j's remaining neighbor in the junction tree also contains j since the unique path must pass through V_j's only remaining neighbor. This is a contradiction. Now we know that any node that contains j is some V_ℓ with $1 \leq \ell \leq k$. Since $V_j = V_{j+1} = \cdots = V_k$, the lemma follows. □

Lemma 2. *If the variables are ordered by Algorithm 2, then for any $j > \ell$ such that $j \in \bar{V}_\ell$, we have $\bar{V}_\ell \subseteq \bar{V}_j$.*

Proof of Lemma 2. Consider any $k \in \bar{V}_\ell$. Now we show such a k must be in $\bar{V}_j$. The case $k \leq j$ is trivial, since $k \in \bar{V}_j$ by definition. If $k > j$, then $j, k \in V_\ell$. Assume $V_\ell \neq V_j$; otherwise there is nothing to prove. Before j—and, therefore, k—are sampled, each time a node V containing j and k (e.g., V_ℓ) is selected, j and k appear in V's neighbor. By a recursive argument, before V_j is selected, each time the node containing j and k is selected, j does not get sampled and neither does k (k is sampled after j). Hence, before V_j is selected, there is always at least one remaining node that contains both j and k. By Algorithm 2 and Lemma 1, V_j is the last selected node j appears in, so it has to contain both j and k (otherwise no node contains both j and k at this point). This means that $k \in V_j \subseteq \bar{V}_j$. □

Lemma 3. *If the variables are ordered by Algorithm 2, then for any $j \neq k$, if j and k are connected in G, we have $j \in \bar{V}_k$ and $k \in \bar{V}_j$.*

Proof of Lemma 3. Without loss of generality, we assume $k > j$, so $j \in \bar{V}_k$ by definition. Now we show $k \in \bar{V}_j$ also holds. By the second property of the junction tree, k co-appears with j at least once in some node V_ℓ. But j does not appear in any node after V_j by Lemma 1, so there is some $\ell \leq j$ such that $\{j, k\} \subseteq V_\ell$. If $j = \ell$, then we already have $k \in V_\ell = V_j \subseteq \bar{V}_j$; otherwise, $j > \ell$, so by Lemma 2, $k \in V_\ell \subseteq \bar{V}_\ell \subseteq \bar{V}_j$. □

B Covariance-guided proposals

This section includes details on the covariance-guided proposal introduced in Section 3.2, and proofs of its faithfulness and compatibility. See Appendix D for details on how to do the necessary linear algebra computations for the covariance-guided proposals efficiently.

We first recall the definition of the covariance-guided proposals. Let Z be a $2p$-dimensional vector drawn from $\mathcal{N}((\mu,\mu),\mathbf{\Gamma}(s))$, where $\mathbf{\Gamma}(s)$ is as in (2). Let q_j be the probability density function of Z_{p+j} conditional on $Z_{1:(p+j-1)}$. The proposal distribution at the jth step $q_j(x_j^* \mid x_{1:p}, x^*_{1:(j-1)})$, is defined to be the conditional density of Z_{p+j} given $Z_{1:(p+j-1)} = (x_{1:p}, x^*_{1:(j-1)})$.[h]

Proposition 4. *The covariance-guided proposals are faithful (see Section 3.1) in that the proposal distribution at the jth step depends on $(X_k, \tilde{X}_k)$ in a symmetric way for $1 \le k < j$.*

Proof of Proposition 4. Consider the proposal at step j. To see how the proposal distribution depends on $(X_k, \tilde{X}_k)$ for $k < j$, note that we are using the distribution of Z_{j+p} conditional on

$$Z_{1:(p+j-1)} = (x_1, x_2, \ldots, x_p, \mathbf{1}_{x_1=\tilde{x}_1} x_1^* + \mathbf{1}_{x_1\neq\tilde{x}_1}\tilde{x}_1, \ldots, \mathbf{1}_{x_{j-1}=\tilde{x}_{j-1}} x^*_{j-1} + \mathbf{1}_{x_{j-1}\neq\tilde{x}_{j-1}}\tilde{x}_{j-1}). \quad (17)$$

We only need to check if the proposal density changes when swapping x_k and $\tilde{x}_k$ for $k \le j-1$. Note that if we rejected at the kth step, $x_k = \tilde{x}_k$, so there is no effect of swapping the two; if we accepted at the kth step, the dependence is symmetric in $(x_k, \tilde{x}_k)$ because of the structure of the covariance matrix. □

Proposition 5. *If $(\mathbf{\Sigma}^{-1})_{ij} = 0$ whenever $i \neq j$ and (i,j) is not an edge in the graph G, then the covariance-guided proposals are compatible with G (see Definition 2).*

Proof of Proposition 5. We wish to show, $\mathcal{L}(X_j^* \mid X, X^*_{1:j-1})$ only depends on X_k if $k \in \bar{V}_j$. To do this, we use induction over j. For $j = 1$, since $X_1^* \mid X$ is a Gaussian distribution whose conditional variance does not depend on X, it suffices to show that $\mathbb{E}[X_1^* \mid X]$ depends on X_k only if $k \in \bar{V}_1$. Note that

$$(X_1, X_1^*) \mid X_{\text{-}1} \sim \mathcal{N}\left((\mu^{\text{cond}}, \mu^{\text{cond}}), \mathbf{\Sigma}^{\text{cond}}\right), \quad \mu^{\text{cond}} = \mathbb{E}[X_1 \mid X_{\text{-}1}], \quad \mathbf{\Sigma}^{\text{cond}} = \operatorname{Cov}\left((X_1, X_1^*) \mid X_{\text{-}1}\right),$$

Since $\mathbf{\Sigma}^{\text{cond}}$ does not depend on X, $\mathbb{E}[X_1^* \mid X_1, X_{\text{-}1}]$ is a linear function of X_1 and μ^{cond}. It is easy to see that $\mu^{\text{cond}} = \mathbb{E}[X_1 \mid X_{\text{-}1}]$ depends on X_k only if k and 1 co-appear in some node of the junction tree. By Lemma 1, this node can only be V_1 and, therefore, $k \in V_1$. Thus the base case $j = 1$ holds.

Suppose the claim is true up to $j-1$. By the same argument on the conditional distribution $(X_j, X_j^*) \mid X_{\text{-}j}, X^*_{1:(j-1)}$, it suffices to show that $\mathbb{E}[X_j \mid X_{\text{-}j}, X^*_{1:(j-1)}]$ only depends on X_k if $k \in \bar{V}_j$. We have

$$\begin{aligned}
\mathbb{P}(x_j \mid x_{\text{-}j}, x^*_{1:(j-1)}) &= \frac{\mathbb{P}(x_j, x_{\text{-}j}, x^*_{1:(j-1)})}{\int_{\mathbb{R}} \mathbb{P}(x_j', x_{\text{-}j}, x^*_{1:(j-1)})\, dx_j'} \\
&= \frac{\mathbb{P}(x_j, x_{\text{-}j}) \prod_{\ell=1}^{j-1} \mathbb{P}(x_\ell^* \mid x_j, x_{\text{-}j}, x^*_{1:(\ell-1)})}{\int_{\mathbb{R}} \mathbb{P}(x_j', x_{\text{-}j}) \prod_{\ell=1}^{j-1} \mathbb{P}(x_\ell^* \mid x_j', x_{\text{-}j}, x^*_{1:(\ell-1)})\, dx_j'} \\
&= \frac{\mathbb{P}(x_{\text{-}j})\mathbb{P}(x_j \mid x_{\text{-}j}) \prod_{\ell=1}^{j-1} \mathbb{P}(x_\ell^* \mid x_j, x_{\text{-}j}, x^*_{1:(\ell-1)})}{\int_{\mathbb{R}} \mathbb{P}(x_{\text{-}j})\mathbb{P}(x_j' \mid x_{\text{-}j}) \prod_{\ell=1}^{j-1} \mathbb{P}(x_\ell^* \mid x_j', x_{\text{-}j}, x^*_{1:(\ell-1)})\, dx_j'} \\
&= \frac{\mathbb{P}(x_j \mid x_{\text{-}j}) \prod_{\ell=1}^{j-1} \mathbb{P}(x_\ell^* \mid x_j, x_{\text{-}j}, x^*_{1:(\ell-1)})}{\int_{\mathbb{R}} \mathbb{P}(x_j' \mid x_{\text{-}j}) \prod_{\ell=1}^{j-1} \mathbb{P}(x_\ell^* \mid x_j', x_{\text{-}j}, x^*_{1:(\ell-1)})\, dx_j'}.
\end{aligned}$$

[h] It seems equally plausible to use $x_{1:p}$ and $\tilde{x}_{1:(j-1)}$ (i.e., $(x_1, x_2, \ldots, x_p, \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{j-1})$ instead of equation (17)). However, we find that its empirical performance is not as good as the version presented in the main text.

By the induction hypothesis, for $\ell < j$, $\mathbb{P}(x^*_\ell \mid x_j, x_{\text{-}j}, x^*_{1:(\ell-1)})$ does not depend on x_j unless $j \in \bar{V}_\ell$ (which implies $j \in V_\ell$ since $j > \ell$), so the ℓth term in the product can be removed from the numerator and the denominator if $j \notin V_\ell$. Thus, we now have

$$\mathbb{P}(x_j \mid x_{\text{-}j}, x^*_{1:(j-1)}) = \frac{\mathbb{P}(x_j \mid x_{\text{-}j}) \prod_{\ell:\ell<j,\ j\in V_\ell} \mathbb{P}(x^*_\ell \mid x_j, x_{\text{-}j}, x^*_{1:(\ell-1)})}{\int_{\mathbb{R}} \mathbb{P}(x'_j \mid x_{\text{-}j}) \prod_{\ell:\ell<j,\ j\in V_\ell} \mathbb{P}(x^*_\ell \mid x'_j, x_{\text{-}j}, x^*_{1:(\ell-1)})\, dx'_j}.$$

Now we will prove that $\mathbb{P}(x_j \mid x_{\text{-}j}) \prod_{\ell:\ell<j,\ j\in V_\ell} \mathbb{P}(x^*_\ell \mid x_j, x_{\text{-}j}, x^*_{1:(\ell-1)})$ depends on x_k only if $k \in \bar{V}_j$, which will conclude case j. Consider first the terms $\mathbb{P}(x^*_\ell \mid x_j, x_{\text{-}j}, x^*_{1:(\ell-1)})$ in the product: such a term depends on x_k only if $k \in \bar{V}_\ell$ by the induction hypothesis, and by Lemma 2, $\bar{V}_\ell \subseteq \bar{V}_j$. Next, the term $\mathbb{P}(x_j \mid x_{\text{-}j})$ only depends on x_k if $k = j$ or k is connected to j in G. If $k = j$, $k \in \bar{V}_j$ by definition; otherwise, $k \in \bar{V}_j$ follows from Lemma 3. □

We have established that as long as $\mathbf{\Sigma}$ reflects the structure of G, i.e., for $i \neq j$, $\left(\mathbf{\Sigma}^{-1}\right)_{ij} \neq 0$ only if i and j are connected in G, the covariance-guided proposals are compatible. For these proposals, sampling and evaluating the proposal density can be done without without querying the density Φ, so Theorem 2 implies that Metro with covariance-guided proposals requires $O(p2^w)$ queries of Φ.

It is easy to see that if X is Gaussian and $\gamma = 1$, we always accept because the acceptance ratio is always 1.

C Proofs

C.1 Necessity of the knockoff symmetry condition

At first glance, it might not be directly clear why we need the symmetry condition (5) in Theorem 1. To illustrate why this is necessary, consider the following example: let p be the density function

$$p(x, \tilde{x}) = 1 + \sin\left(2\pi\Big(x_p + \tilde{x}_p + \sum_{j=1}^{p-1}(x_j - \tilde{x}_j)\Big)\right), \qquad (x, \tilde{x}) \in [0,1]^{2p}.$$

Each pair $(X_j, \tilde{X}_j)$ is unexchangeable unless $j = p$. However, marginalizing out any coordinate would yield the uniform distribution. In other words, in any sequential construction, we would have that X_j and $\tilde{X}_j$ are conditionally independent and, therefore, exchangeable up until the last step. Since X_p and $\tilde{X}_p$ are exchangeable conditionally on everything else, conditional exchangeability would hold. This example shows that if we require conditional exchangeability only, we would not necessarily end up with valid knockoffs. To press this point further, imagine that in the SCIP algorithm, we change the last step: instead of conditional independence we simply require conditional exchangeability. Then we are not guaranteed to get valid knockoffs. Violation of the symmetry condition in just one step is, in general, not allowed.

C.2 Section 2 proofs

Proof of Theorem 1. When condition 1 is met, we have

$$(X_1, \ldots, X_p, \tilde{X}_1, \ldots, \tilde{X}_j) \overset{d}{=} (X_1, \ldots, X_p, \tilde{X}_1, \ldots, \tilde{X}_j)_{\text{swap}(k)}, \quad 1 \leq k \leq j, \tag{18}$$

for each $j = 1, \ldots, p$ by marginalizing out $\tilde{X}_{(j+1):p}$ in (1). This implies both (4) and (5).

Assume now that (4) and (5) hold. We prove by induction that (18) holds for $j = 1, 2, \ldots, p$; when $j = p$, we achieve pairwise exchangeability (1). When $j = 0$, there is nothing to prove. Assume (18) holds up until $j - 1$. The distribution of $(X_1, \ldots, X_p, \tilde{X}_1, \ldots, \tilde{X}_j)$ can be decomposed into the marginal distribution of $(X_{-j}, \tilde{X}_{1:(j-1)})$ and the conditional distribution $(X_j, \tilde{X}_j) \mid X_{-j}, \tilde{X}_{1:(j-1)}$. The former is symmetric in X_k and $\tilde{X}_k$, $1 \le k \le j-1$ as seen by taking the induction hypothesis and marginalizing out X_j. The latter is symmetric in X_j and $\tilde{X}_j$ because of (4), and symmetric in X_k and $\tilde{X}_k$ for $1 \le k \le j-1$ because of (5). □

Proof of Proposition 1. Let $Z_1 \sim \pi$, and the Markov kernel be given by the law $\mathcal{L}(\tilde{Z} \mid Z)$. Then the chain has π as a stationary distribution. Also, $(Z_1, Z_2) \overset{d}{=} (Z, \tilde{Z})$. Time reversibility also holds since

$$(Z_t, Z_{t+1}) \overset{d}{=} (Z, \tilde{Z}) \overset{d}{=} (\tilde{Z}, Z) \overset{d}{=} (Z_{t+1}, Z_t).$$

The converse is a direct consequence of time reversibility. □

C.3 Proof of Theorem 3

Proof. Suppose we are given a procedure $\mathcal{K}$ that always generates valid knockoffs for X given an unnormalized density function Φ for X and (implicitly) the support of Φ (or the dominating measure). Below, the symbols $\mathbb{P}_\Phi$, $\mathcal{L}_\Phi$, etc., indicate that we are working in the probability space defined by Φ (together with its support) and, implicitly, the procedure $\mathcal{K}$; $(X, \tilde{X}, N)$ are jointly defined on this space. Let π be the normalized density, which is defined by Φ through

$$\pi(x) = \lambda_\Phi \Phi(x), \quad x \in \mathbb{R}^p.$$

We abuse notation slightly and for a Borel set M write $\Phi(X \in M)$ for $\mathbb{P}_\Phi(X \in M)/\lambda_\Phi$.

We will consider the conditional probability $\mathbb{P}_\Phi(N \ge 2^{\#\{j:X_j \ne \tilde{X}_j\}} - 1 \mid X, \tilde{X})$ and prove it is almost surely one. Conditioning on $(X, \tilde{X})$ makes the probability easier to analyze because it fixes the exponent $\#\{j : X_j \ne \tilde{X}_j\}$. We will basically identify which $2^{\#\{j:X_j \ne \tilde{X}_j\}} - 1$ points have to be queried: any point obtained by changing x_j to $\tilde{x}_j$ for j in any non-empty subset of $\{j : x_j \ne \tilde{x}_j\}$. We will prove the theorem for both discrete and continuous distributions. Analysis of the conditional probability can be done directly in the discrete case, while in the continuous case we cover the possible values of $(X, \tilde{X})$ by a countable union of sets, and then prove the conditional probability of interest is one on each of the sets. Although more technical, the proof for the continuous case shares the same structure as that for the discrete case.

Discrete case. Let $(x, \tilde{x})$ be some pair of input and output, respectively, of $\mathcal{K}$. For any $S \subseteq \{1, 2, \ldots, p\}$, define $x_{\mathrm{ch}(S)}$ as x except with x_j changed to $\tilde{x}_j$ for all $j \in S$, and vice versa for $\tilde{x}_{\mathrm{ch}(S)}$, so that

$$(x, \tilde{x})_{\mathrm{swap}(S)} = (x_{\mathrm{ch}(S)}, \tilde{x}_{\mathrm{ch}(S)}).$$

We will prove that as long as $x_{\mathrm{ch}(S)} \ne x$, then $\mathcal{K}$ must have queried the oracle at $x_{\mathrm{ch}(S)}$; now assume $x_{\mathrm{ch}(S)} \ne x$. We also assume $\mathbb{P}_\Phi(X = x, \tilde{X} = \tilde{x}) > 0$ (otherwise $(x, \tilde{x})$ is not a possible pair of input and output), which implies $\pi(x_{\mathrm{ch}(S)}) > 0$ for any $S \subseteq \{1, 2, \ldots, p\}$ by pairwise exchangeability. Let $q_\Phi(\tilde{x} \mid x) = \mathbb{P}_\Phi(\tilde{X} = \tilde{x} \mid X = x)$. Also by pairwise exchangeability,

$$\pi(x) q_\Phi(\tilde{x} \mid x) = \pi(x_{\mathrm{ch}(S)}) q_\Phi(\tilde{x}_{\mathrm{ch}(S)} \mid x_{\mathrm{ch}(S)}) \le \pi(x_{\mathrm{ch}(S)}). \tag{19}$$

Let A_x be the event that the input vector is x, so

$$\mathbb{P}_\Phi(A_x) = \pi(x) = \lambda_\Phi \Phi(x).$$

Let $B_{\tilde{x}}$ be the event that the output of $\mathcal{K}$ is $\tilde{x}$, so

$$\mathbb{P}_\Phi(B_{\tilde{x}} \mid A_x) = q_\Phi(\tilde{x} \mid x).$$

Let C_S be the event that $\mathcal{K}$ does not query the oracle at $x_{\mathrm{ch}(S)}$. Dividing (19) by λ_Φ gives

$$\Phi(x_{\mathrm{ch}(S)}) \geq \Phi(x) q_\Phi(\tilde{x} \mid x) = \Phi(x)\mathbb{P}_\Phi(B_{\tilde{x}} \mid A_x) \geq \Phi(x)\mathbb{P}_\Phi(B_{\tilde{x}} \cap C_S \mid A_x). \tag{20}$$

Equation (20) holds for any Φ such that $\Phi(x), \Phi(\tilde{x}) > 0$. Consider a new unnormalized density

$$\Phi_\eta^S(y) = \begin{cases} \eta\Phi(y), & y = x_{\mathrm{ch}(S)}, \\ \Phi(y), & \text{otherwise}, \end{cases}$$

where $\eta \in (0, 1]$. This new density has the same support as Φ, so $\Phi_\eta^S(x_{\mathrm{ch}(S')}) > 0$ for any $S' \subseteq \{1, 2, \ldots, p\}$, and thus (20) also holds for Φ_η^S. Now consider $\mathbb{P}_{\Phi_\eta^S}(B_{\tilde{x}} \cap C_S \mid A_x) = \mathbb{P}_{\Phi_\eta^S}(B_{\tilde{x}} \mid C_S, A_x)\mathbb{P}_{\Phi_\eta^S}(C_S \mid A_x)$. The first probability does not depend on η because the conditioning on C_S means changing the oracle only at $x_{\mathrm{ch}(S)}$ does not affect the procedure $\mathcal{K}$ in any way. The second probability does not depend on η either, for the points that $\mathcal{K}$ queries can only depend on η after $\mathcal{K}$ queries the oracle at $x_{\mathrm{ch}(S)}$. Therefore, $\mathbb{P}_{\Phi_\eta^S}(B_{\tilde{x}} \cap C_S \mid A_x) = \mathbb{P}_\Phi(B_{\tilde{x}} \cap C_S \mid A_x)$ for any $\eta \in (0, 1]$. Thus, we get from equation (20) that (recall we are assuming $x \neq x_{\mathrm{ch}(S)}$)

$$\eta\Phi(x_{\mathrm{ch}(S)}) = \Phi_\eta^S(x_{\mathrm{ch}(S)}) \geq \Phi_\eta^S(x)\mathbb{P}_{\Phi_\eta^S}(B_{\tilde{x}} \cap C_S \mid A_x) = \Phi(x)\mathbb{P}_\Phi(B_{\tilde{x}} \cap C_S \mid A_x).$$

Since $\Phi(x) > 0$ (i.e., x is a possible input), we conclude by letting $\eta \to 0$ that $\mathbb{P}_\Phi(B_{\tilde{x}} \cap C_S \mid A_x) = 0$. Combining this with $\mathbb{P}_\Phi(B_{\tilde{x}} \mid A_x) > 0$ (i.e., given x as an input, $\tilde{x}$ is a possible output), we conclude $\mathbb{P}_\Phi(C_S \mid A_x, B_{\tilde{x}}) = 0$. That is, if $\mathcal{K}$ generates $\tilde{x}$ from input x, it must have queried $x_{\mathrm{ch}(S)}$. Thus, combining the results for all the S's that make $x \neq x_{\mathrm{ch}(S)}$, we can claim that given x as input, $\mathcal{K}$ outputs $\tilde{x}$ only if it has queried the oracle at least at the set of points

$$H_{(x,\tilde{x})} = \{x_{\mathrm{ch}(S)} : S \subseteq \{j : x_j \neq \tilde{x}_j\}, S \neq \emptyset\}.$$

Mathematically, when $\mathbb{P}_\Phi(A_x, B_{\tilde{x}}) > 0$,

$$\mathbb{P}_\Phi(\mathcal{K} \text{ queried } \Phi \text{ at } z \text{ for all } z \in H_{(x,\tilde{x})} \mid A_x, B_{\tilde{x}}) = 1. \tag{21}$$

Since there are $2^{\#\{j:x_j \neq \tilde{x}_j\}} - 1$ points in $H_{(x,\tilde{x})}$, we have

$$\mathbb{P}_\Phi(N \geq 2^{\#\{j:x_j \neq \tilde{x}_j\}} - 1 \mid A_x, B_{\tilde{x}}) = 1.$$

After marginalizing out x and $\tilde{x}$, this leads to the claimed a.s. inequality:

$$N \geq 2^{\#\{j:X_j \neq \tilde{X}_j\}} - 1.$$

Continuous case. The proof works similarly for the continuous case. Loosely speaking, we will construct non-overlapping hypercubes around the points in $H_{(x,\tilde{x})}$ defined previously, and show they all contain points of query with probability one. Concretely, consider a hypercube around z, defined as

$$F_{(z,\tilde{z})} = \{x : |x_k - z_k| \leq g(z_k, \tilde{z}_k), 1 \leq k \leq p\},$$

where

$$g(z_k, \tilde{z}_k) = \begin{cases} \dfrac{|z_k - \tilde{z}_k|}{3}, & z_k \neq \tilde{z}_k, \\ 1, & z_k = \tilde{z}_k. \end{cases}$$

The denominator 3 in the definition of g is not essential, as long as it is large enough so $F_{(z,\tilde{z})}$ does not overlap with $F_{(z,\tilde{z})_{\text{swap}(j)}}$ if $z_j \neq \tilde{z}_j$. Let $E_{(z,\tilde{z})}$ be a joint hypercube around $(z, \tilde{z})$, which is defined through $F_{(z,\tilde{z})}$ and $F_{(\tilde{z},z)}$ as

$$E_{(z,\tilde{z})} = \{(x, \tilde{x}) : x \in F_{(z,\tilde{z})}, \tilde{x} \in F_{(\tilde{z},z)}\} = \{(x, \tilde{x}) : |x_k - z_k|, |\tilde{x}_k - \tilde{z}_k| < g(z_k, \tilde{z}_k), 1 \leq k \leq p\}.$$

We use the fact that the rational points are dense in $\mathbb{R}^{2p}$ to cover the entire space using a countable collection of sets. Hence, if we can prove the conditional probability $\mathbb{P}_\Phi(N \geq 2^{\#\{j:X_j \neq \tilde{X}_j\}} - 1 \mid X, \tilde{X}) = 1$ on every set in this collection, we can claim the corresponding equality holds unconditionally. Formally, we first consider $E_{(r,\tilde{r})}$, where $(r, \tilde{r}) \in \mathbb{Q}^{2p}$ and $r_k \neq \tilde{r}_k, 1 \leq k \leq q$ for some positive integer $q \leq p$. We will show $\mathbb{P}_\Phi(N \geq 2^q - 1 \mid (X, \tilde{X}) \in E_{(r,\tilde{r})}) = 1$ as long as $\mathbb{P}_\Phi((X, \tilde{X}) \in E_{(r,\tilde{r})}) > 0$. Now suppose $\mathbb{P}_\Phi((X, \tilde{X}) \in E_{(r,\tilde{r})}) > 0$, which implies $\mathbb{P}_\Phi(X \in F_{(r,\tilde{r})_{\text{swap}(S)}}) > 0$ for any $S \subseteq \{1, 2, \ldots, p\}$. Define

$$A_{(r,\tilde{r})} = \{X \in F_{(r,\tilde{r})}\}, \quad B_{(\tilde{r},r)} = \{\tilde{X} \in F_{(\tilde{r},r)}\}.$$

Let S be any non-empty subset of $\{1, 2, \ldots, q\}$, so $(r, \tilde{r})_{\text{swap}(S)} \neq (r, \tilde{r})$. Now we have

$$\mathbb{P}_\Phi((X, \tilde{X}) \in E_{(r,\tilde{r})}) = \mathbb{P}_\Phi(A_{(r,\tilde{r})} \cap B_{(\tilde{r},r)}) = \mathbb{P}_\Phi(A_{(r,\tilde{r})}) \mathbb{P}_\Phi(B_{(\tilde{r},r)} \mid A_{(r,\tilde{r})}),$$

and

$$\begin{aligned} \mathbb{P}_\Phi((X, \tilde{X}) \in E_{(r,\tilde{r})}) &= \mathbb{P}_\Phi((X, \tilde{X})_{\text{swap}(S)} \in E_{(r,\tilde{r})}) \\ &= \mathbb{P}_\Phi((X, \tilde{X}) \in E_{(r,\tilde{r})_{\text{swap}(S)}}) \\ &= \mathbb{P}_\Phi(X \in F_{(r,\tilde{r})_{\text{swap}(S)}}, \tilde{X} \in F_{(\tilde{r},r)_{\text{swap}(S)}}) \\ &= \mathbb{P}_\Phi(X \in F_{(r,\tilde{r})_{\text{swap}(S)}}) \mathbb{P}_\Phi(\tilde{X} \in F_{(\tilde{r},r)_{\text{swap}(S)}} \mid X \in F_{(r,\tilde{r})_{\text{swap}(S)}}) \\ &\leq \mathbb{P}_\Phi(X \in F_{(r,\tilde{r})_{\text{swap}(S)}}). \end{aligned}$$

Hence, by dividing by the common normalizing constant λ_Φ in the above two equations and combining them, we get

$$\Phi(X \in F_{(r,\tilde{r})_{\text{swap}(S)}}) \geq \Phi(X \in F_{(r,\tilde{r})}) \mathbb{P}_\Phi(B_{(\tilde{r},r)} \mid A_{(r,\tilde{r})}). \tag{22}$$

Now, similar to the discrete case, we consider a new unnormalized density

$$\Phi_\eta^S(x) = \begin{cases} \eta\Phi(x) & x \in F_{(r,\tilde{r})_{\text{swap}(S)}}, \\ \Phi(x) & \text{otherwise}, \end{cases}$$

for $\eta \in (0, 1]$, which has the same support/dominating measure as Φ. It is easy to check that equation (22) also holds for Φ_η^S. Let C_S be the event that $\mathcal{K}$ does not query Φ at any point in $F_{(r,\tilde{r})_{\text{swap}(S)}}$. To use the same trick as in the discrete case, we next prove $\mathbb{P}_{\Phi_\eta^S}(C_S \cap B_{(\tilde{r},r)} \mid A_{(r,\tilde{r})})$ does not depend on η. We have $\Phi_\eta^S(X \in F_{(r,\tilde{r})}) = \Phi(X \in F_{(r,\tilde{r})})$ because $F_{(r,\tilde{r})} \cap F_{(r,\tilde{r})_{\text{swap}(S)}}$ is empty (so $\Phi_\eta^S = \Phi$ on $F_{(r,\tilde{r})}$). Note that by definition,

$$\begin{aligned} \mathbb{P}_\Phi(C_S \cap B_{(\tilde{r},r)} \mid A_{(r,\tilde{r})}) &= \mathbb{E}_\Phi[\mathbb{P}_\Phi(C_S \cap B_{(\tilde{r},r)} \mid X) \mid A_{(r,\tilde{r})}], \\ \mathbb{P}_{\Phi_\eta^S}(C_S \cap B_{(\tilde{r},r)} \mid A_{(r,\tilde{r})}) &= \mathbb{E}_{\Phi_\eta^S}[\mathbb{P}_{\Phi_\eta^S}(C_S \cap B_{(\tilde{r},r)} \mid X) \mid A_{(r,\tilde{r})}], \end{aligned} \tag{23}$$

and $\mathbb{P}_{\Phi_\eta^S}(C_S \cap B_{(\tilde{r},r)} \mid X)$, as a function of X, does not depend on η; this holds since all Φ_η^S's for $\eta \in (0,1]$ have the same support, and we can follow the same argument as in the discrete case. Specifically,

$$\mathbb{P}_{\Phi_\eta^S}(C_S \cap B_{(\tilde{r},r)} \mid X) = \mathbb{P}_{\Phi}(C_S \cap B_{(\tilde{r},r)} \mid X).$$

In addition, since the unnormalized density is the same on the set $F_{(r,\tilde{r})}$, we have

$$\mathcal{L}_{\Phi}(X \mid A_{(r,\tilde{r})}) = \mathcal{L}_{\Phi_\eta^S}(X \mid A_{(r,\tilde{r})}).$$

The last two equations together with equations (23) imply $\mathbb{P}_{\Phi}(C_S \cap B_{(\tilde{r},r)} \mid A_{(r,\tilde{r})}) = \mathbb{P}_{\Phi_\eta^S}(C_S \cap B_{(\tilde{r},r)} \mid A_{(r,\tilde{r})})$. By (22),

$$\begin{aligned}
\eta\Phi(X \in F_{(r,\tilde{r})_{\mathrm{swap}(S)}}) &= \Phi_\eta^S(X \in F_{(r,\tilde{r})_{\mathrm{swap}(S)}}) \\
&\geq \Phi_\eta^S(A_{(r,\tilde{r})})\mathbb{P}_{\Phi_\eta^S}(B_{(\tilde{r},r)} \mid A_{(r,\tilde{r})}) \\
&\geq \Phi_\eta^S(A_{(r,\tilde{r})})\mathbb{P}_{\Phi_\eta^S}(C_S \cap B_{(\tilde{r},r)} \mid A_{(r,\tilde{r})}) \\
&= \Phi(A_{(r,\tilde{r})})\mathbb{P}_{\Phi}(C_S \cap B_{(\tilde{r},r)} \mid A_{(r,\tilde{r})}).
\end{aligned} \tag{24}$$

The left hand side of (24) goes to 0 as $\eta \to 0$. Thus, recall that we are assuming $\mathbb{P}_\Phi((X,\tilde{X}) \in E_{(r,\tilde{r})}) > 0$ and $\Phi(X \in F_{(r,\tilde{r})}) > 0$, and so we get $\mathbb{P}_\Phi(C_S \cap B_{(\tilde{r},r)} \mid A_{(r,\tilde{r})}) = 0$. By Bayes' rule, since $\mathbb{P}_\Phi(A_{(r,\tilde{r})} \cap B_{(\tilde{r},r)}) > 0$, we have

$$\begin{aligned}
\mathbb{P}_\Phi(C_S \mid A_{(r,\tilde{r})}, B_{(\tilde{r},r)}) &= \frac{\mathbb{P}_\Phi(C_S \cap A_{(r,\tilde{r})} \cap B_{(\tilde{r},r)})}{\mathbb{P}_\Phi(A_{(r,\tilde{r})} \cap B_{(\tilde{r},r)})} \\
&= \frac{\mathbb{P}_\Phi(A_{(r,\tilde{r})})\mathbb{P}_\Phi(C_S \cap B_{(\tilde{r},r)} \mid A_{(r,\tilde{r})})}{\mathbb{P}_\Phi(A_{(r,\tilde{r})} \cap B_{(\tilde{r},r)})} = 0.
\end{aligned}$$

That is, unless $(X,\tilde{X}) \in E_{(r,\tilde{r})}$ happens with zero probability, with probability one at least one point in $F_{(r,\tilde{r})_{\mathrm{swap}(S)}}$ is queried conditional on $(X,\tilde{X}) \in E_{(r,\tilde{r})}$. Combining the results for all the S's that make $(r,\tilde{r})_{\mathrm{swap}(S)} \neq (r,\tilde{r})$, we get $2^q - 1$ disjoint sets, each of which must contain at least one point of query. Hence, now we can claim that for any $E_{(r,\tilde{r})}$, where $(r,\tilde{r}) \in \mathbb{Q}^{2p}$ and $r_k \neq \tilde{r}_k, 1 \leq k \leq q$, either

$$\mathbb{P}_\Phi((X,\tilde{X}) \in E_{(r,\tilde{r})}) = 0$$

or

$$\mathbb{P}_\Phi(N \geq 2^q - 1 \mid (X,\tilde{X}) \in E_{(r,\tilde{r})}) = 1.$$

Note that this is equivalent to $\mathbb{P}_\Phi(N \geq 2^q - 1 \mid X,\tilde{X}) = 1$ almost surely on $E_{(r,\tilde{r})}$.[i] These two equations imply that, as a function of $(X,\tilde{X})$, the conditional probability satisfies

$$\mathbb{P}_\Phi(N \geq 2^q - 1 \mid X,\tilde{X}) = 1, \quad a.s. \text{ on } \bigcup_{\substack{(r,\tilde{r}) \in \mathbb{Q}^{2p} \\ r_k \neq \tilde{r}_k, 1 \leq k \leq q}} E_{(r,\tilde{r})},$$

because the union is over a countable index set. There is nothing special about choosing the $r_k \neq \tilde{r}_k$ on first q coordinates, so we have

$$\mathbb{P}_\Phi(N \geq 2^{|D|} - 1 \mid X,\tilde{X}) = 1, \quad a.s. \text{ on } \bigcup_{\substack{(r,\tilde{r}) \in \mathbb{Q}^{2p} \\ r_k \neq \tilde{r}_k, k \in D}} E_{(r,\tilde{r})}, \quad D \subseteq \{1,2,\ldots,p\}. \tag{25}$$

[i] The conditional probability $\mathbb{P}_\Phi(N \geq 2^q - 1 \mid X,\tilde{X}) = 1$ is almost surely one on a set U means $\mathbb{P}_\Phi(N \geq 2^q - 1 \mid (X,\tilde{X}) = (x,\tilde{x})) = 1$ for $(x,\tilde{x}) \in U \setminus V$, where V is some set satisfying $\mathbb{P}_\Phi((X,\tilde{X}) \in V) = 0$.

Note the case $D = \emptyset$ does not follow the exact same proof, but no proof is needed in this case, since $N \geq 2^{|\emptyset|} - 1 = 0$ holds trivially. We shall thus keep in mind that (25) holds for any Φ.

Now we go back to the conditional probability we are interested in, mathematically defined as

$$f_\Phi(X, \tilde{X}) = \mathbb{P}_\Phi(N \geq 2^{\#\{j:X_j \neq \tilde{X}_j\}} - 1 \mid X, \tilde{X}) = \mathbb{E}_\Phi[\mathbf{1}_{N \geq 2^{\#\{j:X_j \neq \tilde{X}_j\}} - 1} \mid X, \tilde{X}].$$

We want to show $f_\Phi(X, \tilde{X}) = 1$, a.s. Let

$$T_{n,D} = \{(x, \tilde{x}) : |x_k - \tilde{x}_k| > 1/n, k \in D \text{ and } x_k = \tilde{x}_k, k \notin D\};$$

therefore, D is the set of coordinates where x and $\tilde{x}$ could differ, and $1/n$ measures the minimum difference between these original and knockoff coordinates. Since any point $(x, \tilde{x}) \in \mathbb{R}^{2p}$ is contained in

$$T_{\left\lfloor 1/\min_{j, x_j \neq \tilde{x}_j} |x_j - \tilde{x}_j| \right\rfloor + 1, \{j:x_j \neq \tilde{x}_j\}}$$

if $x \neq \tilde{x}$, and in $T_{1,\emptyset}$ if $x = \tilde{x}$, we have

$$\mathbb{R}^{2p} = \bigcup_{n=1}^{\infty} \bigcup_{D \subseteq \{1,2,\ldots,p\}} T_{n,D}.$$

This is also a countable union, so in order to show $f_\Phi(X, \tilde{X}) = 1$ a.s., we only have to show that $f_\Phi(X, \tilde{X}) = 1$ a.s. for any $T_{n,D}$ that has positive probability of containing $(X, \tilde{X})$. Note that since there are exactly $|D|$ coordinates that differ for x and $\tilde{x}$ in the set $T_{n,D}$,

$$f_\Phi(X, \tilde{X}) = \mathbb{P}_\Phi(N \geq 2^{\#\{j:X_j \neq \tilde{X}_j\}} - 1 \mid X, \tilde{X}) = \mathbb{P}_\Phi(N \geq 2^{|D|} - 1 \mid X, \tilde{X}) \text{ on } T_{n,D}.$$

So now we only need to show

$$\mathbb{P}_\Phi(N \geq 2^{|D|} - 1 \mid X, \tilde{X}) = 1, \quad a.s. \text{ on } T_{n,D},$$

which would be implied by (25) if we can show that

$$\bigcup_{\substack{(r,\tilde{r}) \in \mathbb{Q}^{2p} \\ r_k \neq \tilde{r}_k, k \in D}} E_{(r,\tilde{r})} \supseteq T_{n,D}.$$

To see this, take any point $(x, \tilde{x})$ from $T_{n,D}$. Find rational numbers $r_k \in (x_k - 1/5n, x_k + 1/5n)$ and $\tilde{r}_k \in (\tilde{x}_k - 1/5n, \tilde{x}_k + 1/5n)$, $1 \leq k \leq p$. We have $|r_k - \tilde{r}_k| > 3/5n$ (hence also $r_k \neq \tilde{r}_k$) for $k \in D$, since $|x_k - \tilde{x}_k| > 1/n$ for $k \in D$. We can now check that $(x, \tilde{x}) \in E_{(r,\tilde{r})}$. For $k \in D$ (if any), $|x_k - r_k|, |\tilde{x}_k - \tilde{r}_k| < 1/5n < |r_k - \tilde{r}_k|/3$, and for $k \notin D$ (if any), $|r_k - x_k|, |\tilde{r}_k - \tilde{x}_k| < 1/5n < 1$. □

C.4 Divide-and-conquer knockoffs

Proof of Proposition 3. Consider the distribution of X conditional on $X_C = x_C$. Since C separates A and B in the graph G, we have

$$X_A \perp\!\!\!\perp X_B \mid X_C.$$

The assumptions of the proposition then imply that

$$(X_A, X_B, \tilde{X}_A, \tilde{X}_B) \stackrel{d}{=} (X_A, X_B, \tilde{X}_A, \tilde{X}_B)_{\text{swap}(j)} \mid X_C, \quad j \in A \cup B$$

and since $\tilde{X}_C = X_C$ a.s.,

$$(X, \tilde{X}) \stackrel{d}{=} (X, \tilde{X})_{\mathrm{swap}(j)} \mid X_C, \quad j \in A \cup B \cup C.$$

Lastly, we note that conditional exchangeability implies marginal exchangeability, so

$$(X, \tilde{X}) \stackrel{d}{=} (X, \tilde{X})_{\mathrm{swap}(j)}, \quad j \in A \cup B \cup C$$

as claimed. □

C.5 Complexity proofs for Metropolized knockoff sampling

Lemma 4. *When the proposal distributions are compatible for the junction tree T, for $1 \leq j \leq p$, $\mathbb{P}(\tilde{X}_j, X_j^* \mid X, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)})$, depends on X_k only if $k \in \bar{V}_j$.*

Proof of Lemma 4. We use induction over j. For the base case $j = 1$, $\mathbb{P}(X_1^* \mid X)$ depends on X_k only if $k \in \bar{V}_1$ by our assumption of compatible proposals. And $\mathbb{P}(\tilde{X}_1 \mid X, X_1^*)$ is a function of the acceptance probability

$$\frac{\mathbb{P}(X_1^* = x_1^* \mid X_1 = x_1, X_{\text{-}1})}{\mathbb{P}(X_1^* = x_1 \mid X_1 = x_1^*, X_{\text{-}1})} \frac{\mathbb{P}(X_1 = x_1^*, X_{\text{-}1})}{\mathbb{P}(X_1 = x_1, X_{\text{-}1})}.$$

The first term depends only on $X_k \in \bar{V}_1$ by assumption of compatible proposals. The second term depends on X_k if k is connected to 1 in G (or $k = 1$). Since the variables are ordered by Algorithm 2, 1 only appears in V_1 by Lemma 1, so k has to appear in V_1 if 1 and k are connected. The base case is thus proved.

Suppose the claim is true for $1, \ldots, j-1$. First we have $\mathbb{P}(X_j^* \mid X, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)})$ depends on X_k only if $k \in \bar{V}_j$ by our assumption of compatible proposals. Now $\mathbb{P}(\tilde{X}_j = \tilde{x}_j \mid X, \tilde{X}_{1:(j-1)}, X^*_{1:j})$ is a function of the acceptance probability, which is computed from the ratio of the proposal densities (which depends only on X_k for $k \in \bar{V}_j$ by assumption of compatible proposals) and the following ratio

$$\frac{\mathbb{P}(X_j = x_j^*, X_{\text{-}j}, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)})}{\mathbb{P}(X_j = x_j, X_{\text{-}j}, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)})} = \frac{\mathbb{P}(X_j = x_j^* \mid X_{\text{-}j})}{\mathbb{P}(X_j = x_j \mid X_{\text{-}j})} \frac{\mathbb{P}(\tilde{X}_{1:(j-1)}, X^*_{1:(j-1)} \mid X_j = x_j^*, X_{\text{-}j})}{\mathbb{P}(\tilde{X}_{1:(j-1)}, X^*_{1:(j-1)} \mid X_j = x_j, X_{\text{-}j})}. \quad (26)$$

We first consider the second term in the right hand side of the above. Consider the decomposition

$$\mathbb{P}(\tilde{X}_{1:(j-1)}, X^*_{1:(j-1)} \mid X_j = z_j, X_{\text{-}j}) = \prod_{\ell=1}^{j-1} \mathbb{P}(\tilde{X}_\ell, X_\ell^* \mid X_j = z_j, X_{\text{-}j} \tilde{X}_{1:(\ell-1)}, X^*_{1:(\ell-1)}).$$

The ℓth term in this product depends on X_j only if $j \in \bar{V}_\ell$, by the induction hypothesis. Lemma 2 then implies that for such ℓ, $\bar{V}_\ell \subseteq \bar{V}_j$. Any term in the product that does not depend on X_j will be identical in the numerator and denominator of (26) and so will cancel. Together, this shows that the second term on the right hand side of (26) depends only on k for $k \in \bar{V}_j$.

Next, the numerator and denominator of the first term of the right hand side of (26) only depends on X_k if $k = j$ or k is connected to j in G. If $k = j$, then $k \in \bar{V}_j$ by definition; otherwise $k \in \bar{V}_j$ by Lemma 3. Now we have showed (26) depends only on X_k for $k \in \bar{V}_j$ and the proof of the lemma is complete. □

Proof of Theorem 2. Take $\gamma = 1$ for simplicitly, and take a proposal distribution q_j that can be sampled from and evaluated without an evaluation of Φ (e.g., an independent Gaussian proposal in the continuous setting). We will show how Algorithm 2 uses the conditional dependence structure encoded in the graph G to make computations of (10) simpler.

Define

$$F_j(X_{V_j} = z_{V_j}) := \mathbb{P}(\tilde{X}_j, X_j^* \mid X_{V_j} = z_{V_j}, X_{V_j^c}, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)}).$$

By the definition of Metro, we can write F_j as the product of the proposal density and the acceptance/rejection probability:

$$\begin{aligned} F_j(X_{V_j} = z_{V_j}) &= \mathbb{P}(\tilde{X}_j, X_j^* \mid X_{V_j} = z_{V_j}, X_{V_j^c}, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)}) \\ &= \mathbb{P}(X_j^* \mid X_{V_j} = z_{V_j}, X_{V_j^c}, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)}) \mathbb{P}(\tilde{X}_j \mid X_{V_j} = z_{V_j}, X_{V_j^c}, \tilde{X}_{1:(j-1)}, X^*_{1:j}) \\ &= q^{(2)} \alpha^{\mathbf{1}_{\text{accept}}} (1-\alpha)^{\mathbf{1}_{\text{reject}}}; \end{aligned}$$

where

$$\begin{aligned} q^{(1)} &= \mathbb{P}(X_j^* = z_j \mid X_j = x_j^*, X_{V_j \setminus j} = z_{V_j \setminus j}, X_{V_j^c}, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)}) \\ q^{(2)} &= \mathbb{P}(X_j^* = x_j^* \mid X_j = z_j, X_{V_j \setminus j} = z_{V_j \setminus j}, X_{V_j^c}, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)}). \end{aligned}$$

are the proposal terms and

$$\alpha = \min\left(1, \frac{q^{(1)} \mathbb{P}(X_j = x_j^*, X_{V_j \setminus j} = z_{V_j \setminus j}, X_{V_j^c}, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)})}{q^{(2)} \mathbb{P}(X_j = z_j, X_{V_j \setminus j} = z_{V_j \setminus j}, X_{V_j^c}, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)})}\right)$$

is the acceptance probability. Sequentially decomposing the ratio of probabilities in the term α, we get

$$\begin{aligned} &\frac{\mathbb{P}(X_j = x_j^*, X_{V_j \setminus j} = z_{V_j \setminus j}, X_{V_j^c}, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)})}{\mathbb{P}(X_j = z_j, X_{V_j \setminus j} = z_{V_j \setminus j}, X_{V_j^c}, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)})} \\ &= \frac{\mathbb{P}(X_j = x_j^*, X_{V_j \setminus j} = z_{V_j \setminus j}, X_{V_j^c})}{\mathbb{P}(X_j = z_j, X_{V_j \setminus j} = z_{V_j \setminus j}, X_{V_j^c})} \times \prod_{k=1}^{j-1} \frac{\mathbb{P}(\tilde{X}_k, X_k^* \mid X_j = x_j^*, X_{V_j \setminus j} = z_{V_j \setminus j}, X_{V_j^c}, \tilde{X}_{1:(k-1)}, X^*_{1:(k-1)})}{\mathbb{P}(\tilde{X}_k, X_k^* \mid X_j = z_j, X_{V_j \setminus j} = z_{V_j \setminus j}, X_{V_j^c}, \tilde{X}_{1:(k-1)}, X^*_{1:(k-1)})} \\ &= \frac{\Phi(X_j = x_j^*, X_{V_j \setminus j} = z_{V_j \setminus j}, X_{V_j^c})}{\Phi(X_j = z_j, X_{V_j \setminus j} = z_{V_j \setminus j}, X_{V_j^c})} \times \prod_{k=1}^{j-1} \frac{\mathbb{P}(\tilde{X}_k, X_k^* \mid X_j = x_j^*, X_{V_j \setminus j} = z_{V_j \setminus j}, X_{V_j^c}, \tilde{X}_{1:(k-1)}, X^*_{1:(k-1)})}{\mathbb{P}(\tilde{X}_k, X_k^* \mid X_j = z_j, X_{V_j \setminus j} = z_{V_j \setminus j}, X_{V_j^c}, \tilde{X}_{1:(k-1)}, X^*_{1:(k-1)})}, \end{aligned} \tag{27}$$

where all we did in the second equality was cancel the normalizing constants in the first ratio. In light of Lemma 4, the ratio

$$\frac{\mathbb{P}(\tilde{X}_k, X_k^* \mid X_j = x_j^*, X_{V_j \setminus j} = z_{V_j \setminus j}, X_{V_j^c}, \tilde{X}_{1:(k-1)}, X^*_{1:(k-1)})}{\mathbb{P}(\tilde{X}_k, X_k^* \mid X_j = z_j, X_{V_j \setminus j} = z_{V_j \setminus j}, X_{V_j^c}, \tilde{X}_{1:(k-1)}, X^*_{1:(k-1)})}$$

is one unless $j \in \bar{V}_k$, since the value of X_j is the only one that differs between the numerator and denominator. Recall that $\bar{V}_k = V_k \cup \{1, 2, \ldots, k\}$ and note $j > k$, so $j \in \bar{V}_k$ is equivalent to $j \in V_k$.

Thus, (27) gives

$$\frac{\mathbb{P}(X_j = x_j^*, X_{V_j\setminus j} = z_{V_j\setminus j}, X_{V_j^c}, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)})}{\mathbb{P}(X_j = z_j, X_{V_j\setminus j} = z_{V_j\setminus j}, X_{V_j^c}, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)})}$$
$$= \frac{\Phi(X_j = x_j^*, X_{V_j\setminus j} = z_{V_j\setminus j}, X_{V_j^c})}{\Phi(X_j = z_j, X_{V_j\setminus j} = z_{V_j\setminus j}, X_{V_j^c})} \times$$
$$\prod_{k:k<j,\ j\in V_k} \frac{\mathbb{P}(\tilde{X}_k, X_k^* \mid X_j = x_j^*, X_{V_j\setminus j} = z_{V_j\setminus j}, X_{V_j^c}, \tilde{X}_{1:(k-1)}, X^*_{1:(k-1)})}{\mathbb{P}(\tilde{X}_k, X_k^* \mid X_j = z_j, X_{V_j\setminus j} = z_{V_j\setminus j}, X_{V_j^c}, \tilde{X}_{1:(k-1)}, X^*_{1:(k-1)})}.$$

Now we will show that the numerators in the product satisfy

$$\mathbb{P}(\tilde{X}_k, X_k^* \mid X_j = x_j^*, X_{V_j\setminus j} = z_{V_j\setminus j}, X_{V_j^c}, \tilde{X}_{1:(k-1)}, X^*_{1:(k-1)})$$
$$= F_k(X_j = x_j^*, X_{V_k\cap V_j\setminus j} = z_{V_k\cap V_j\setminus j}, X_{V_k\setminus V_j} = x_{V_k\setminus V_j}).$$

By the definition of F_k, we need to show that

$$\mathbb{P}(\tilde{X}_k, X_k^* \mid X_j = x_j^*, X_{V_j\setminus j} = z_{V_j\setminus j}, X_{V_j^c}, \tilde{X}_{1:(k-1)}, X^*_{1:(k-1)})$$
$$= \mathbb{P}(\tilde{X}_k, X_k^* \mid X_j = x_j^*, X_{V_k\cap V_j\setminus j} = z_{V_k\cap V_j\setminus j}, X_{V_k\setminus V_j} = x_{V_k\setminus V_j}, X_{V_k^c}, \tilde{X}_{1:(k-1)}, X^*_{1:(k-1)}).$$

Inspecting the equation, we note that the only difference between the two quantities is the value of the variables that are being conditioned on, and only the values of $X_{V_j\setminus V_k}$ are are different. Thus, we only need to show this set of values do not affect the conditional density. By Lemma 4, we just have to show that $V_j \setminus V_k$ does not overlap with $\bar{V}_k$. To see this, take any $\ell \in \bar{V}_k = V_k \cup \{1, 2, \ldots, k\}$. If $\ell \in V_k$, then certainly $\ell \notin V_j \setminus V_k$. Now we consider the case where $\ell \in \{1, 2, \ldots, k\}$ and thus less than j. If $\ell \in V_j \setminus V_k$, then it must be in V_j. By Lemma 1, we have $V_j = V_\ell$, which means variables $\ell, \ldots, j$ are all sampled when V_ℓ is selected; specifically, k is sampled when V_ℓ is selected, so $V_k = V_\ell = V_j$. But in this case certainly $V_j \setminus V_k = \emptyset$, which is a contradiction.

Similarly, we also have that the corresponding denominators in the product satisfy

$$\mathbb{P}(\tilde{X}_k, X_k^* \mid X_j = z_j, X_{V_j\setminus j} = z_{V_j\setminus j}, X_{V_j^c}, \tilde{X}_{1:(k-1)}, X^*_{1:(k-1)})$$
$$= F_k(X_j = z_j, X_{V_k\cap V_j\setminus j} = z_{V_k\cap V_j\setminus j}, X_{V_k\setminus V_j} = x_{V_k\setminus V_j}).$$

Combining all these together, the acceptance probability becomes

$$\alpha = \min\left(1, \frac{q^{(1)}\mathbb{P}(X_j = x_j^*, X_{V_j\setminus j} = z_{V_j\setminus j}, X_{V_j^c}, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)})}{q^{(2)}\mathbb{P}(X_j = z_j, X_{V_j\setminus j} = z_{V_j\setminus j}, X_{V_j^c}, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)})}\right)$$
$$= \min\left(1, \frac{q^{(1)}c^{(1)}\Phi(X_j = x_j^*, X_{V_j\setminus j} = z_{V_j\setminus j}, X_{V_j^c})}{q^{(2)}c^{(2)}\Phi(X_j = z_j, X_{V_j\setminus j} = z_{V_j\setminus j}, X_{V_j^c})}\right),$$

where

$$c^{(1)} = \prod_{k:k<j,\ j\in V_k} F_k(X_j = x_j^*, X_{V_k\cap V_j\setminus j} = z_{V_k\cap V_j\setminus j}, X_{V_k\setminus V_j} = x_{V_k\setminus V_j}),$$
$$c^{(2)} = \prod_{k:k<j,\ j\in V_k} F_k(X_j = z_j, X_{V_k\cap V_j\setminus j} = z_{V_k\cap V_j\setminus j}, X_{V_k\setminus V_j} = x_{V_k\setminus V_j}).$$

Note that the only difference between $c^{(1)}$ and $c^{(2)}$ is changing x_j^* to z_j. From this expression, we see that $\left\{F_j(X_{V_j} = z_{V_j}) : z_\ell \in \{x_\ell, x_\ell^*\} \text{ for all } \ell \in V_j\right\}$ can be computed in terms of

1. $F_k(X_{V_k} = z_{V_k})$ for all $k < j$ with z_{V_k} such that $z_\ell \in \{x_\ell, x_\ell^*\}$ for all $\ell \in V_k$,

2. $\Phi(X_{V_j} = z_{V_j}, X_{V_j^c} = x_{V_j^c})$ for all z_{V_j} with $z_\ell \in \{x_\ell, x_\ell^*\}$ for all $\ell \in V_j$.

Thus, it requires $2^{|V_j|}$ evaluations of Φ to compute $\left\{F_j(X_{V_j} = z_{V_j}) : z_\ell \in \{x_\ell, x_\ell^*\} \text{ for all } \ell \in V_j\right\}$ from the previously computed values of $F_k(X_{V_k} = z_{V_k})$ for $k < j$. Since $|V_j| \leq w + 1$, $1 \leq j \leq p$, we have that the total number of queries of Φ is $O(p2^w)$. Having access to the $F_k(X_{V_k} = z_{V_k})$'s is sufficient to run the algorithm, because at each step of the algorithm, it is clear that the acceptance ratio α can be computed from these $F_k(X_{V_k} = z_{V_k})$'s, so the proof is complete. □

More generally, if an evaluation of Φ costs a units of computation and a floating point operation requires 1 unit, then the same proof shows that the algorithm takes $O(p(p+a)2^w)$ since computing the $F_j(X_{V_j} = z_{V_j})$ requires a total of $O(p2^w)$ evaluations of Φ and each $F_j(X_{V_j} = z_{V_j})$ requires $O(p)$ floating point operations to compute $c^{(1)}$ and $c^{(2)}$.

MTM

For the MTM method of Section 3.3, the proposal distribution requires evaluations of Φ, so the requirements of Theorem 2 do not hold. The proof of Theorem 2 can easily be adapted to apply to MTM, however. Instead, for the MTM method we see that at step j we need access to

$$\mathbb{P}(X_j = z_j, X_{\text{-}j} = x_{\text{-}j}, \tilde{X}_{1:(j-1)} = \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)} = x^*_{1:(j-1)})$$

up to a common constant for $z_j \in C^{m,t}_{x_j} \cup C^{m,t}_{x_j^*}$. By an analysis similar to the proof of Theorem 2, it suffices to have access to

1. $F_k(X_{V_k} = z_{V_k})$ for all $k < j$ for z_{V_k} such that $z_\ell \in C^{m,t}_{x_\ell} \cup C^{m,t}_{x_\ell^*}$ for all $k \in V_k$,

2. $\Phi(X_j = z_j, X_{\text{-}j} = x_{\text{-}j})$ for all $z_j \in C^{m,t}_{x_j} \cup C^{m,t}_{x_j^*}$.

In order to compute $F_j(X_{V_j} = z_{V_j})$ for all z_{V_j} such that $z_\ell \in C^{m,t}_{x_\ell} \cup C^{m,t}_{x_\ell^*}$ for all $\ell \in V_j$ for use in later steps, we additionally need to compute

$$\Phi(X_{V_j} = z_{V_j}, X_{V_j^c} = x_{V_j^c}) \text{ for all } z_{V_j} \text{ such that } z_\ell \in C^{m,t}_{x_\ell} \cup C^{m,t}_{x_\ell^*} \text{ for all } \ell \in V_j.$$

Thus, MTM requires $O(p(3m+1)^w)$ queries of Φ, where $3m+1$ is an upper bound on $|C^{m,t}_{x_\ell} \cup C^{m,t}_{x_\ell^*}|$ for all ℓ.

Discrete distributions with small support

Consider the direct methods for discrete distributions with small support in Section 4.5, which can be viewed as a Metro algorithm that never rejects. The proof of Theorem 2 can easily be adapted to apply to the this case. Note that since the procedure never rejects, we have $X^* = \tilde{X}$ and we can omit writing terms of X^* in all of the following discussion.

Let C_j be the support of X_j and suppose that $|C_j| \leq K$ for all j. Then at step j, the method requires access to

$$\mathbb{P}(X_j = z_j, X_{\text{-}j} = x_{\text{-}j}, \tilde{X}_{1:(j-1)} = \tilde{x}_{1:(j-1)})$$

for all $z_j \in C_j$. By an analysis similar to the proof of Theorem 2, it suffices to have access to

1. $F_k(X_{V_k} = z_{V_k})$ for all $k < j$ for z_{V_k} such that $z_\ell \in C_k$ for all $\ell \in V_j$.

2. $\Phi(X_j = z_j, X_{\text{-}j} = x_{\text{-}j})$ for all $z_j \in C_j$.

In order to compute $F_j(X_{V_j} = z_{V_j})$ for all z_{V_j} such that $z_\ell \in C_\ell$ for all $\ell \in V_j$ for use in later steps, we additionally need to compute

$$\Phi(X_{V_j} = z_{V_j}, X_{V_j^c} = x_{V_j^c}) \text{ for all } z_{V_j} \text{ such that } z_\ell \in C_\ell \text{ for all } \ell \in V_j.$$

Thus, the rejection-free procedure requires $O(pK^w)$ queries of Φ.

C.6 Lower bounds for graphical models

Proof of Proposition 2. We only have to prove that we can design a Metro algorithm that meets the requirement in the proposition. The rest of the proposition is implied by Corollary 2, proved next. We focus on the continuous case (and when the support is all of $\mathbb{R}^p$), because we are interested in the Gaussian case. Since Metro can only learn about the distribution by making queries to the oracle, we describe an algorithm which first makes about $p^2/2$ queries to attempt to recover the covariance matrix. This can be done in such a way that if the model is Gaussian as described in Proposition 2, we recover the correct covariance matrix. Thus, covariance-guided proposals will accept at every step and get us a knockoff vector that differs with the input original vector at every coordinate (with probability one).

Let e_j be the jth vector of the canonical basis. Assume the model is $\mathcal{N}(0, \mathbf{\Sigma})$ let Φ be the unnormalized density. Then

$$\begin{aligned}
\Phi(0) &= W \cdot \exp\left(-\frac{1}{2} 0 \mathbf{\Sigma}^{-1} 0^\top\right) = W, \\
\Phi(e_j) &= W \cdot \exp\left(-\frac{1}{2} e_j \mathbf{\Sigma}^{-1} e_j^\top\right) = W \cdot \exp\left(-\frac{1}{2}\left(\mathbf{\Sigma}^{-1}\right)_{jj}\right), \ 1 \le j \le p, \\
\Phi(e_j + e_k) &= W \cdot \exp\left(-\frac{1}{2}\left(\left(\mathbf{\Sigma}^{-1}\right)_{jj} + \left(\mathbf{\Sigma}^{-1}\right)_{kk} + 2\left(\mathbf{\Sigma}^{-1}\right)_{jk}\right)\right), \ j \neq k,
\end{aligned}$$

where W is an unknown positive constant. Hence, if we query the oracle at the above $1 + p(p+1)/2$ points, we can always solve for a potential precision matrix. The algorithm's next step depends on the solution to these equations.

- If the matrix formed by the solution to this system of equations is positive definite, and reflects the structure of the graph G, i.e., the (i, j)th entry is non-zero only if $i = j$ or i and j are connected by an edge in G, then the algorithm inverts this solution matrix to get $\mathbf{\Sigma}$, and then proceeds with covariance-guided proposals with any positive s which makes $\mathbf{\Gamma}(s)$ in (2) positive definite.
- Otherwise, the model must not be a multivariate Gaussian distribution with zero mean, positive definite covariance matrix and have the required conditional independence structure. In that case, the algorithm will just choose any proposal distribution (e.g., independent Gaussian proposals).

Since running Metro with the covariance-guided proposal requires $O(p2^w)$ queries of Φ, this algorithm in total requires $O(p^2 + p2^w)$ queries of Φ. If indeed $\Phi(x) \propto \exp\left(-x\mathbf{\Sigma}^{-1}x^\top/2\right)$, the algorithm will recover the right covariance matrix $\mathbf{\Sigma}$, and therefore will never reject, and produce a knockoff $\tilde{X}$ such that $X_j \neq \tilde{X}_j$ for all j (with probability one). □

Proof of Corollary 2. Call the procedure $\mathcal{K}$. We argue by contradiction and show that if the Corollary did not hold, we would be able to exploit $\mathcal{K}$ and construct an algorithm that contradicts Theorem 3. Loosely speaking, if there is one clique c_0 for which, with positive probability, the inequality fails to hold, then we can design an algorithm that generates knockoffs for any $|c_0|$-dimensional random vector X_{c_0} by inferring the "missing" variables $X_{\{1:p\}\setminus c_0}$, applying $\mathcal{K}$ to get $(\tilde{X}_{c_0}, \tilde{X}_{\{1:p\}\setminus c_0})$, and keeping only $\tilde{X}_{c_0}$, which is a valid knockoff of X_{c_0}.

Formally, if there exists a $\Phi_0 = \prod_{c\in C} \phi_c(x_c)$ such that with positive probability

$$N < \max_{c\in C} 2^{\#\{j\in c: X_j \neq \tilde{X}_j\}} - 1,$$

then there must exist some clique $c_0 \in C$ such that simultaneously $\#\{j \in c_0 : X_j \neq \tilde{X}_j\} = \max_{c\in C} \#\{j \in c : X_j \neq \tilde{X}_j\}$ and $N < 2^{\#\{j\in c_0: X_j \neq \tilde{X}_j\}} - 1$ with positive probability. Note that such a c_0 must not be empty ($|c_0| \geq 1$), since we cannot have $N < 0$ with positive probability. Fix this distribution Φ_0 and this clique c_0. For each x_{c_0} in the domain, use $\Phi_0(\cdot \mid x_{c_0})$ to denote the normalized density of $X_{\{1:p\}\setminus c_0} \mid X_{c_0} = x_{c_0}$ when $X \sim \Phi_0$, and consider any sampler $\mathcal{S}_{x_{c_0}}$ for this conditional distribution. This sampler takes a $|c_0|$-dimensional vector x_{c_0} and produces a sample from $\Phi_0(\cdot \mid x_{c_0})$. Now consider the following generic procedure for knockoff sampling:

1. The user inputs unnormalized density Ψ_{c_0} and vector X_{c_0}, where X_{c_0} follows the distribution induced by the unnormalized density Ψ_{c_0}.
2. Sample $X_{\{1:p\}\setminus c_0}$ from the conditional distribution $\Phi_{c_0}(\cdot \mid X_{c_0})$ using the sampler $\mathcal{S}_{X_{c_0}}$.
3. Provide $\Phi'(z) := \Psi_{c_0}(z_{c_0})\Phi_{c_0}(z_{\{1:p\}\setminus c_0} \mid z_{c_0})$ as a function of z and the realization $(X_{c_0}, X_{\{1:p\}\setminus c_0})$ as an input to procedure $\mathcal{K}$, which then returns $(\tilde{X}_{c_0}, \tilde{X}_{\{1:p\}\setminus c_0})$. Let N bet the number of queries of Φ' required by $\mathcal{K}$.
4. Return $\tilde{X}_{c_0}$.

This procedure queries Ψ_{c_0} exactly N times, since step 2 uses $\mathcal{S}_{x_{c_0}}$, which does not rely on Ψ_{c_0} and does not query it. Furthermore, step 3 queries Φ' and hence Ψ_{c_0} exactly N times. We now show that the procedure is also guaranteed to produce valid knockoffs for any Ψ_{c_0}. To do this, we only need to show that $\Phi'(z)$ factors over G; note that

$$\Phi'(z) = \Psi_{c_0}(z_{c_0}) \frac{\Phi_0(z)}{\int \Phi_0(z_{c_0}, w_{\{1:p\}\setminus c_0})\, dw_{\{1:p\}\setminus c_0}} = \underbrace{\frac{\Psi_{c_0}(z_{c_0})\phi_{c_0}(z_{c_0})}{\int \Phi_0(z_{c_0}, w_{\{1:p\}\setminus c_0})\, dw_{\{1:p\}\setminus c_0}}}_{\text{only depends on } z_{c_0}} \prod_{\substack{c\in C \\ c\neq c_0}} \phi_c(z_c).$$

Since Φ' has the assumed structure implied by G, by the assumption on the validity of $\mathcal{K}$, $(\tilde{X}_{c_0}, \tilde{X}_{\{1:p\}\setminus c_0})$ is a valid knockoff for the augmented random vector $(X_{c_0}, X_{\{1:p\}\setminus c_0})$. Marginally, $X_{c_0} \sim \Psi_{c_0}$, so we simply marginalize out $X_{\{1:p\}\setminus c_0}$ and $\tilde{X}_{\{1:p\}\setminus c_0}$ which preserves pairwise exchangeability.

Finally, because c_0 is the complete graph on $|c_0|$ coordinates, this is a generic knockoff sampler for random vectors of dimension $|c_0|$. Specifically, by our initial choice of Φ_0, letting Ψ_{c_0} correspond to Φ_{c_0} (the marginal density of X_{c_0} when $X \sim \Phi_0$) we have $N < 2^{\#\{j\in c_0: X_j \neq \tilde{X}_j\}} - 1$ with positive probability. This contradicts Theorem 3, which says the inequality must hold with zero probability for any input density, including Φ_{c_0}. □

D Efficient matrix inversion for covariance-guided proposals

Let $\mathbf{\Sigma}_j$ be the matrix composed of the first $(p+j)$ rows and columns of

$$\mathbf{\Gamma} = \begin{bmatrix} \mathbf{\Sigma} & \mathbf{\Sigma} - \operatorname{diag}(s) \\ \mathbf{\Sigma} - \operatorname{diag}(s) & \mathbf{\Sigma} \end{bmatrix}.$$

We want to find the inverses of $\mathbf{\Sigma}_0, \mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_{p-1}$ (assuming $\mathbf{\Sigma}_{p-1}$ is invertible). Note that

$$\mathbf{\Sigma}_{j+1} = \begin{bmatrix} \mathbf{\Sigma}_j & \gamma_{j+1} \\ \gamma_{j+1}^\top & \sigma_{j+1}^2 \end{bmatrix},$$

where σ_{j+1}^2 is the $(j+1)$th diagonal element of $\mathbf{\Sigma}$, and γ_{j+1} is the truncated $(p+j)$th column of $\mathbf{\Gamma}$. We have

$$\mathbf{\Sigma}_{j+1}^{-1} = \begin{bmatrix} \left(\mathbf{\Sigma}_j - \frac{1}{\sigma_{j+1}^2}\gamma_{j+1}\gamma_{j+1}^\top\right)^{-1} & -\frac{1}{\sigma_{j+1}^2 - \gamma_{j+1}^\top \mathbf{\Sigma}_j^{-1}\gamma_{j+1}}\mathbf{\Sigma}_j^{-1}\gamma_{j+1} \\ -\frac{1}{\sigma_{j+1}^2 - \gamma_{j+1}^\top \mathbf{\Sigma}_j^{-1}\gamma_{j+1}}\gamma_{j+1}^\top\mathbf{\Sigma}_j^{-1} & \frac{1}{\sigma_{j+1}^2 - \gamma_{j+1}^\top \mathbf{\Sigma}_j^{-1}\gamma_{j+1}} \end{bmatrix}.$$

And by the Sherman–Morrison formula,

$$\left(\mathbf{\Sigma}_j - \frac{1}{\sigma_{j+1}^2}\gamma_{j+1}\gamma_{j+1}^\top\right)^{-1} = \mathbf{\Sigma}_j^{-1} - \frac{\mathbf{\Sigma}_j^{-1}\gamma_{j+1}\left(\mathbf{\Sigma}_j^{-1}\gamma_{j+1}\right)^\top}{-\sigma_{j+1}^2 + \gamma_{j+1}^\top\mathbf{\Sigma}_j^{-1}\gamma_{j+1}}.$$

With all the elements of recursion in place, we can invert $\mathbf{\Sigma}_0$ and recursively calculate the inverse matrices of $\mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_{p-1}$. We have made code available that implements this recursion efficiently.

E Group knockoffs

We can easily generalize our work to the group knockoff filter first presented in Dai and Barber (2016) to control the group false discovery rate. As in that work, let $\{I_1, I_2, \ldots, I_k\}$ be a partition of $\{1, 2, \ldots, p\}$, and suppose we want to construct $\tilde{X}$ such that for each $j = 1, \ldots, k$,

$$(X_{I_1}, X_{I_2}, \ldots, X_{I_k}, \tilde{X}_{I_1}, \tilde{X}_{I_2}, \ldots, \tilde{X}_{I_k}) \overset{d}{=} (X_{I_1}, X_{I_2}, \ldots, X_{I_k}, \tilde{X}_{I_1}, \tilde{X}_{I_2}, \ldots, \tilde{X}_{I_k})_{\operatorname{swap}(I_j)}.$$

At each step, we can draw a proposal $X_{I_j}^* = x_{I_j}^*$ from a faithful multivariate distribution, and accept it with probability

$$\min\left(1, \frac{q_j(x_{I_j} \mid x_{I_j}^*)\mathbb{P}(X_{\text{-}I_j} = x_{\text{-}I_j}, X_{I_j} = x_{I_j}^*, \tilde{X}_{I_{1:(j-1)}} = \tilde{x}_{I_{1:(j-1)}}, X_{I_{1:(j-1)}}^* = x_{I_{1:(j-1)}}^*)}{q_j(x_{I_j}^* \mid x_{I_j})\mathbb{P}(X_{\text{-}I_j} = x_{\text{-}I_j}, X_{I_j} = x_{I_j}, \tilde{X}_{I_{1:(j-1)}} = \tilde{x}_{I_{1:(j-1)}}, X_{I_{1:(j-1)}}^* = x_{I_{1:(j-1)}}^*)}\right).$$

F Extended simulation results

F.1 Discrete Markov chains simulation details

In Figure 6, the best MTM specification is taken from $\{(\gamma, m, t) : \gamma = 0.999, 1 \le m \le 10, 1 \le t \le 5\}$ for $\alpha = 0.2, 0.3, 0.4, 0.5$, and from $\{(\gamma, m, t) : \gamma = 0.999, 1 \le m \le 10, 1 \le t \le 5\} \cup \{(\gamma, m, t) : m = 4, t = 1, \gamma = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.999\}$ for $\alpha = 0, 0.05, 0.1, 0.15$. Plots showing the the individual performance of each of these methods are included below.

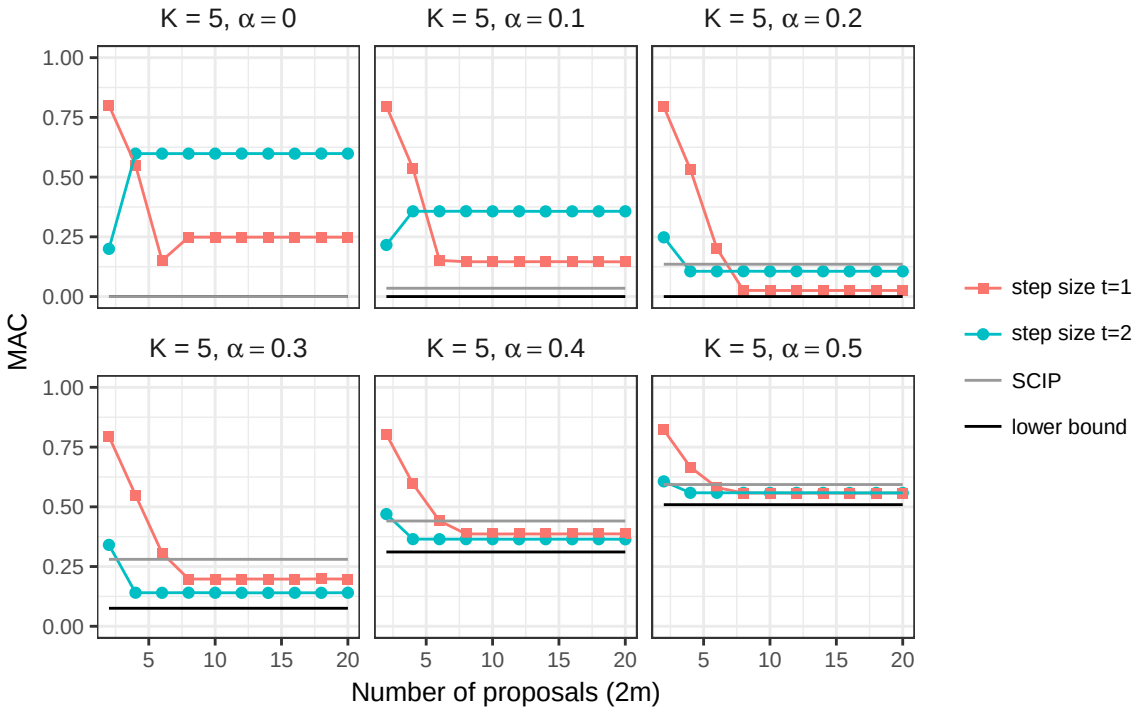

Figure 11: Simulation results for the discrete Markov chains with MTM, $K = 5$, $\gamma = 0.999$. All standard errors are below 0.001.

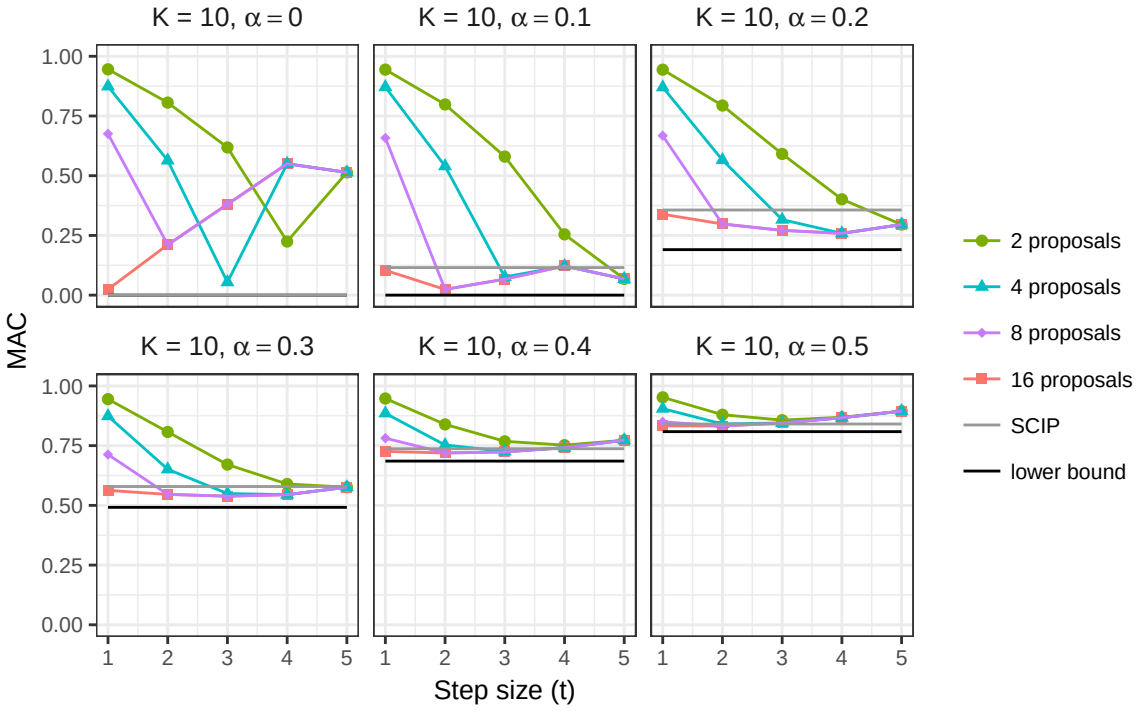

Figure 12: Simulation results for the discrete Markov chains with MTM, $K = 10$, $\gamma = 0.999$. All standard errors are below 0.001.

F.2 Effect of γ

The tuning parameter γ for the MTM procedure introduced in Section 3.3 may appear mysterious to the reader and warrants an explanation. In most of our MTM simulations, γ is set to be 0.999, but there are cases where it is necessary to tune γ for improved performance. For example, in the discrete Markov chain experiment in Figure 11, we sometimes observe that MAC increases with the number of proposals. This is surprising, and upon closer inspection, we find that the reason is that many pairs $(X_j, \tilde{X}_j)$ have negative correlations, which leads to increased MAC. In this case, what is happening is that we are proposing and accepting points that are so far away from X_j that they become negatively correlated with X_j, which is undesirable. To shift the negative correlations toward zero, γ can be decreased so that $X_j = \tilde{X}_j$ more frequently. We illustrate this in Figure 13. For example, in the setting where have independent coordinates taking on $K = 5$ possible states, the best performance is obtained with $\gamma = 4/5$, since with this value of γ there will be a probability $1 - \gamma = 1/5$ of rejection, which makes X_j and $\tilde{X}_j$ independent. Tuning γ may also enable fewer rejections at later stages of the algorithms, since the knockoffs at later coordinates will be less constrained.[j] Based on our simulation results, we only recommend tuning γ when the variables are discrete with small support and the dependence between variables is weak.

F.3 Ising model simulation details

Sampling Ising variables X is done with a Metroplis–Hastings sampler implemented in the `bayess` R package.

F.3.1 Divide-and-conquer simulation details

We set $\tilde{X}_{i_1,i_2} := X_{i_1,i_2}$ for all (i_1, i_2) such that i_1 belongs to the set C of columns defined as $C = \{1 \leq i \leq 100 : i = a_0 + b(w-1), b \in \mathbb{N}\}$; the spacing w is a fixed constant (see Figure 3 for an illustration) and the offset a_0 is chosen uniformly from $\{2, \ldots, w+1\}$. This implies that for all sites i_1, i_2), $\mathbb{P}(X_{i_1,i_2} = \tilde{X}_{i_1,i_2}) < 1$.

The SDP lower bound is not available in this case, because it would require computing a 10000×10000 covariance matrix and then solving the SDP, which is intractable. Instead, to evaluate the quality of the knockoffs, we compare to Ising model knockoffs on a smaller grid that does not require the divide-and-conquer technique. Our baseline is thus the MAC evaluated at interior nodes $1 < i_1, i_2 < 10$ achieved by the SCIP procedure. We consider interior nodes because we recall that correlations on the edges of the grid are smaller. We compare this figure of merit to the MAC of the interior variables of the 100×100 grid. Without the divide-and-conquer strategy, the MAC of the two procedures would be very similar—hence, this is a sensible baseline.

F.4 Gibbs model simulation details

In Figure 10, the best MTM specification is taken from $\{(\gamma, m, t, w) : \gamma = 0.999, 1 \leq m \leq 5, 1 \leq t \leq 7, w = 3\} \cup \{(\gamma, m, t, w) : \gamma = 0.999, m = 1, 1 \leq t \leq 10, w = 5\}$ for $\beta = 0.07, 0.1, 0.3$, and from $\{(\gamma, m, t, w) : \gamma = 0.999, 1 \leq m \leq 5, 1 \leq t \leq 7, w = 3\} \cup \{(\gamma, m, t, w) : \gamma = 0.999, m = 1, 1 \leq t \leq$

[j] Choosing γ less than 1 means that no matter what the proposal X_j^* is, it will be rejected with positive probability. While we typically want to avoid rejections, rejecting at early stages in the algorithm my lead to better performance by enabling higher quality knockoffs at later steps in the algorithm. In particular, with $\gamma = 1$, at some step $k > j$, it might be the case that none of the points in the proposal set have positive probability, because any point in the proposal set is inconsistent with a rejection that occurred previously in step j. When $\gamma < 1$, however, any proposed value at step k is consistent with a rejection at step j, because there is always at least a $1 - \gamma$ chance of rejecting at step j, so we avoid the undesirable situation described above.

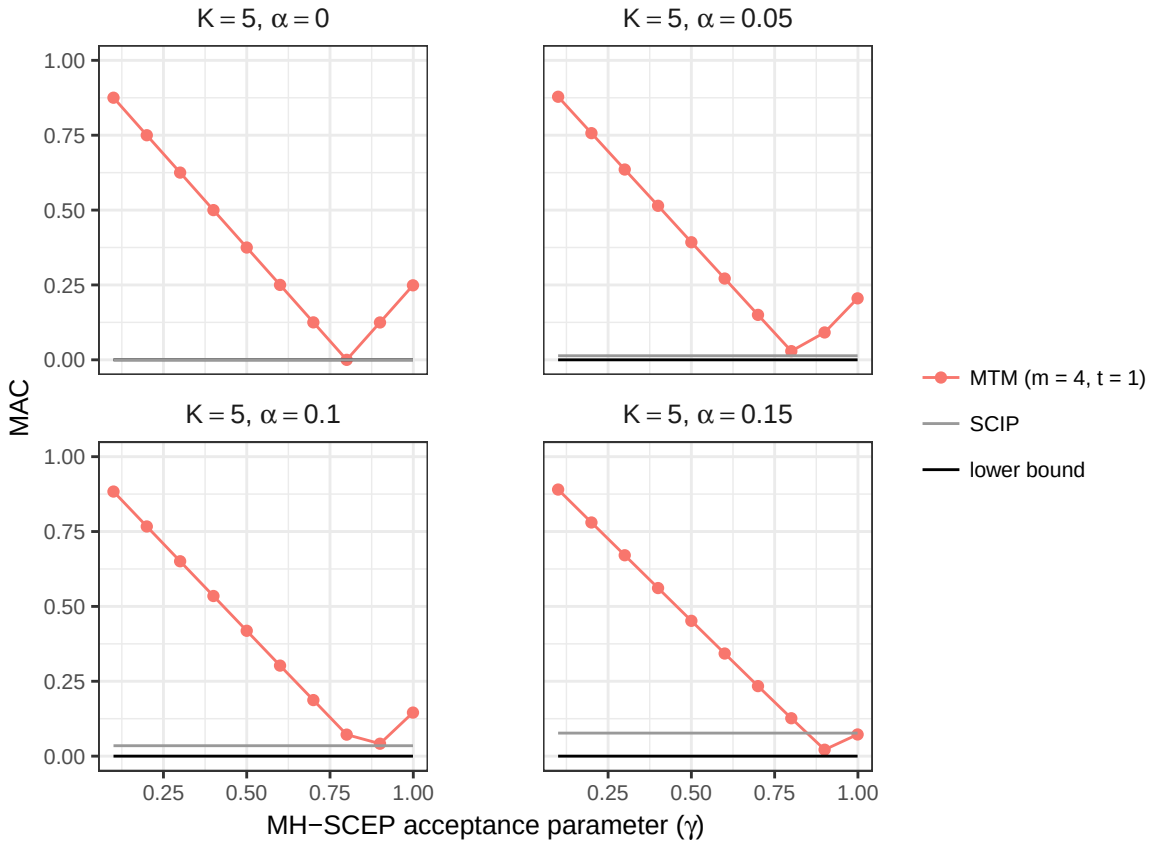

Figure 13: Simulation results showing the effect of the parameter γ for the discrete Markov chains with the MTM method. Here, $K = 5$, $\alpha = 0$, 0.05, 0.1, 0.15, $m = 4$ and $t = 1$. All standard errors are below 0.001.

$10, w = 3, 5\}$ for $\beta = 0.003, 0.01, 0.02, 0.05$. The best $m = 1$, $w = 3$ MTM for each β is taken from the same set intersecting $\{(\gamma, m, t, w) : m = 1, w = 3\}$.