The Kernel Interaction Trick: Fast Bayesian Discovery of Pairwise Interactions in High Dimensions

Raj Agrawal [1]   Jonathan H. Huggins [2]   Brian L. Trippe [1]   Tamara Broderick [1]

Abstract

Discovering interaction effects on a response of interest is a fundamental problem faced in biology, medicine, economics, and many other scientific disciplines. In theory, Bayesian methods for discovering pairwise interactions enjoy many benefits such as coherent uncertainty quantification, the ability to incorporate background knowledge, and desirable shrinkage properties. In practice, however, Bayesian methods are often computationally intractable for even moderate-dimensional problems. Our key insight is that many hierarchical models of practical interest admit a particular Gaussian process (GP) representation; the GP allows us to capture the posterior with a vector of $O(p)$ kernel hyper-parameters rather than $O(p^2)$ interactions and main effects. With the implicit representation, we can run Markov chain Monte Carlo (MCMC) over model hyper-parameters in time and memory *linear* in p per iteration. We focus on sparsity-inducing models and show on datasets with a variety of covariate behaviors that our method: (1) reduces runtime by orders of magnitude over naive applications of MCMC, (2) provides lower Type I and Type II error relative to state-of-the-art LASSO-based approaches, and (3) offers improved computational scaling in high dimensions relative to existing Bayesian and LASSO-based approaches.

1. Introduction

Many decision-making and scientific tasks require understanding how a set of covariates relate to a target response. For example, in clinical trials and precision medicine, re-searchers seek to characterize how individual-level traits impact treatment effects, and in modern genomic studies, researchers seek to identify genetic variants that are risk factors for particular diseases. While linear regression is a default method for these tasks and many others due to its ease of interpretability, its simplicity often comes at the cost of failing to learn more nuanced information from the data. A common way to increase flexibility, while still retaining the interpretability of linear regression, is to augment the covariate space. For instance, two genes together might be highly associated with a disease even though individually they exhibit only moderate association; thus, an analyst might want to consider the multiplicative effect of pairs of covariates co-occurring.

Unfortunately, augmenting the covariate space by including all possible *pairwise interactions* means the number of parameters to analyze grows quadratically with the number of covariates p. This growth leads to many statistical and computational difficulties that are only made worse in the high-dimensional setting, where p is much larger than the number of observations N. And $p \gg N$ is often exactly the case of interest in genomic and medical applications. To address the statistical challenges, practitioners often enforce a sparsity constraint on the model, reflecting an assumption that only a small subset of all covariates affect the response. The problem of identifying this subset is a central problem in high-dimensional statistics and many different LASSO-based approaches have been proposed to return sparse point estimates. However, these methods do not address how to construct valid confidence intervals or adjust for multiple comparisons[1] (Bien et al., 2013; Lim & Hastie, 2015; Nakagawa et al., 2016; Shah, 2016; Wu et al., 2009).

Fortunately, hierarchical Bayesian methods have a shrinkage effect, naturally handle multiplicity, can provide better statistical power than multiple comparison corrections (Gelman et al., 2012), and can leverage background knowledge. However, naive approaches to Bayesian inference are computationally intractable for even moderate-dimensional prob-

---

[1]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA [2]Department of Biostatistics, Harvard, Cambridge, MA. Correspondence to: Raj Agrawal <r.agrawal@csail.mit.edu>.

---

[1]While the knockoff filter introduced in Barber & Candès (2015) is a promising way to control the false discovery rate, such a method has not been evaluated theoretically or empirically for interaction models.

lems. This intractability has two sources. The first source can be seen even in the simple case of conjugate linear regression with a multivariate Gaussian prior. Let $\tilde{X}$ denote the augmented data matrix including all pairwise interactions, Σ the multivariate Gaussian prior covariance on parameters, and σ^2 the noise variance. Given N observations, computing the posterior requires inverting $\Sigma^{-1} + \frac{1}{\sigma^2}\tilde{X}^T\tilde{X}$, which takes $O(p^2 N^2 + N^3)$ time. The second source is that reporting on $O(p^2)$ parameters simply has $O(p^2)$ cost.

We propose to speed up inference in Bayesian linear regression with pairwise interactions by addressing both problems. In the first case, we show how to represent the original model using a Gaussian process (GP). We use the GP kernel in our *kernel interaction sampler* to take advantage of the special structure of interactions and avoid explicitly computing or inverting $\Sigma^{-1} + \tilde{X}^T\tilde{X}$. In the second case, we develop a *kernel interaction trick* to compute posterior summaries exactly for main effects and interactions between selected main effects to avoid the full $O(p^2)$ reporting cost. In sum, we show that we can recover posterior means and variances of regression coefficients in $O(pN^2 + N^3)$ time, a p-fold speed-up. We demonstrate the utility and efficacy of our general-purpose computational tools for the *sparse kernel interaction model* (SKIM), which we propose in Section 6 for identifying sparse interactions. In Section 7 we empirically show (1) improved Type I and Type II error relative to state-of-the-art LASSO-based approaches and (2) improved computational scaling in high dimensions relative to existing Bayesian and LASSO-based approaches. Our methods extend naturally beyond pairwise interactions to higher-order *multi-way* interactions, as detailed in Appendix A.

2. Preliminaries and Related Work

Suppose we observe data $D = \{(x^{(n)}, y^{(n)})\}_{n=1}^N$ with covariates $x^{(n)} \in \mathbb{R}^p$ and responses $y^{(n)} \in \mathbb{R}$. Let $X \in \mathbb{R}^{N \times p}$ denote the design matrix and $Y \in \mathbb{R}^N$ denote the vector of responses. Linear models assume that each $y^{(n)}$ is a (noisy) linear function of the covariates $x^{(n)}$. A common strategy to increase the expressivity of linear models is to augment the original covariates $x^{(n)}$ with their pairwise interactions

$$\Phi_2^T(x) := [1, x_1, \cdots, x_p, x_1 x_2, \cdots, x_{p-1} x_p, x_1^2, \cdots, x_p^2].$$

That is, for a parameter $\theta \in \mathbb{R}^{p(p+1)/2}$ and zero-mean i.i.d. errors $\epsilon^{(n)}$, we assume the data are generated according to

$$y^{(n)} = \theta^T \Phi_2(x^{(n)}) + \epsilon^{(n)}. \tag{1}$$

Our goal is to identify which interaction terms have a significant effect on the response. Detecting such interactions is important for many applications. For example, in genomics, two-way interaction terms are needed to detect possible epistasis between genes (Aschard, 2016; Slim et al., 2018) and

to appropriately account for the site- and sample-specific effects of GC content on genomic and other types of sequencing data (Benjamini & Speed, 2012; Risso et al., 2011). In economics and clinical trials, pairwise interactions between covariates and treatment are used to estimate the heterogeneous effect a treatment has across different subgroups (Lipkovich et al., 2017, Section 6). Unfortunately, having $O(p^2)$ parameters creates statistical and computational challenges when p is large.

To address the statistical issues, practitioners often assume that θ is *sparse* (i.e., contains only a few non-zero values), and that θ satisfies *strong hierarchy*. That is, an interaction effect $\theta_{x_i x_j}$ is present only if both of the main effects θ_{x_i} and θ_{x_j} are present, where $\theta_{x_i x_j}$ and θ_{x_i} are the regression coefficients of the variables $x_i x_j$ and x_i respectively (Bien et al., 2013; Lim & Hastie, 2015; Nakagawa et al., 2016; Wu et al., 2009). By assuming such low-dimensional structure, inference tasks such as parameter estimation and variable selection become more tractable statistically. However, sparsity constraints create computational difficulties. For example, finding the maximum-likelihood estimator (MLE) subject to $\|\theta\|_0 \leq s$ requires searching over $\Theta(p^{2s})$ active parameter subsets. To avoid the combinatorial issues resulting from an L_0 penalty, recent works (Bien et al., 2013; Lim & Hastie, 2015) have instead used L_1 penalties to encourage parameter sparsity for interaction models; L_1 penalties have a long history in high-dimensional linear regression (Candes & Tao, 2007; Chen et al., 1998; Tibshirani, 1994),

Maximizing the likelihood with an added L_1 penalty is a convex problem. But each iteration of a state-of-the-art solver for methods given by Bien et al. (2013) and Lim & Hastie (2015) still takes $O(Np^2)$ time. To handle larger p, Nakagawa et al. (2016); Shah (2016); Wu et al. (2009) have proposed various pruning heuristics for finding locally optimal solutions. However, since these methods do not provide an exact solution to the optimization problem, any statistical guarantees (such as the statistical rate at which these estimators converge to the true parameter as a function of N and p) are weaker than those for exact methods.

L_1-based methods face a number of additional challenges: constructing valid confidence intervals, incorporating background knowledge, and controlling for the issue of multiple comparisons when testing many parameters for statistical significance. In many applications such as genome-wide association studies, controlling for multiplicity is critical to prevent wasting resources on false discoveries. Moreover, since $\dim(\Phi_2) = p(p+1)/2$, θ can be very high dimensional even when p is moderately large. Hence, there will typically be nontrivial uncertainty when attempting to estimate θ. Fortunately, hierarchical Bayesian methods have (1) a natural shrinkage or regularization effect such that mul-

tiple testing corrections are no longer necessary, (2) better statistical power than using multiple comparison correction terms such as Bonferroni (Gelman et al., 2012), and (3) naturally provide calibrated uncertainties. Bayesian methods can also incorporate expert information.

Though they offer desirable statistical properties, Bayesian approaches are computationally expensive. Previous efforts (Chipman, 1996; Griffin & Brown, 2017) have focused on developing *hierarchical sparsity priors* that promote strong hierarchy, analogous to the LASSO-based approaches (Bien et al., 2013; Lim & Hastie, 2015; Nakagawa et al., 2016; Wu et al., 2009). But these methods do not address the computational intractability of inference for even moderate-dimensional problems.

We address the computational challenges of inference by developing the *kernel interaction trick* (Section 5), which allows us to access posterior marginals of θ without ever representing θ explicitly. Note that while some previous works have used a degree-two polynomial kernel to implicitly generate all pairwise interactions (Morota & Gianola, 2014; Su et al., 2012; Weissbrod et al., 2016), those works have focused on prediction or effect-size estimation rather than our present focus on inference.

3. Bayesian Models with Interactions

Our goal is to estimate and provide uncertainties for the parameter $\theta \in \mathbb{R}^{\dim(\Phi_2)}$. To take a Bayesian approach, we encode the state of knowledge before observing the data D in a *prior* $\pi_0(\theta)$. We express the *likelihood* as $\mathcal{L}(Y \mid \theta, X) = \prod_{n=1}^{N} \mathcal{L}(y^{(n)} \mid \theta, x^{(n)})$. Applying Bayes' theorem yields the *posterior* distribution $\pi(\theta \mid D) \propto \mathcal{L}(Y \mid \theta, X)\pi_0(\theta)$, which describes the state of knowledge about θ after observing the data D. For a function f of interest, we wish to compute the posterior expectation

$$\mathbb{E}_{\pi(\theta|D)}[f(\theta)] = \int f(\theta)\pi(\theta \mid D)d\theta. \qquad (2)$$

Typically, $f(\theta) = \theta_j$ or $f(\theta) = \theta_j^2$, which together allow us to compute the posterior mean and variance of each θ_j.

Generative model. Going forward, we model θ as being drawn from a *Gaussian scale mixture* prior to encode desirable properties such as sparsity and strong hierarchy (cf. Carvalho et al., 2009; Chipman, 1996; George & McCulloch, 1993; Griffin & Brown, 2017; Piironen & Vehtari, 2017). These priors have also been used beyond sparse Bayesian regression (cf. Choy & Chan, 2003; Hamdan et al., 2005; Wainwright & Simoncelli, 1999). A Gaussian scale mixture is equivalent to assuming that there exists an auxiliary random variable $\tau \sim p(\tau)$ such that θ is conditionally Gaussian given τ. Let Σ_τ denote the covariance matrix for $p(\theta \mid \tau)$. Also, let σ^2 be the latent noise variance in the likelihood;

Table 1. Per-iteration MCMC runtime and memory scaling of methods for sampling two-way interactions. NAIVE refers to explicitly factorizing $\Sigma_{N,\tau}$ to compute $p(D \mid \tau, \sigma^2)$, WOODBURY refers to using the Woodbury identity and matrix determinant lemma to compute $p(D \mid \tau, \sigma^2)$, and FULL refers to jointly sampling θ and τ. The third column provides the number of parameters sampled.

METHOD	TIME	MEMORY	#
OUR METHOD	$O(pN^2 + N^3)$	$O(pN + N^2)$	$O(p)$
NAIVE	$O(p^6 + p^2N)$	$O(p^4 + p^2N)$	$O(p)$
WOODBURY	$O(p^2N^2 + N^3)$	$O(p^2N + N^2)$	$O(p)$
FULL	$O(p^2N)$	$O(p^2N)$	$\Theta(p^2)$

since it is typically unknown, we treat it as random and put a prior on it as well. Hence, the full generative model can be written

$$\begin{aligned} \tau &\sim p(\tau) \\ \sigma^2 &\sim p(\sigma^2) \\ \theta \mid \tau &\sim \mathcal{N}(0, \Sigma_\tau) \\ y^{(n)} \mid x^{(n)}, \theta, \sigma^2 &\sim \mathcal{N}(\theta^T \Phi_2(x^{(n)}), \sigma^2). \end{aligned} \qquad (3)$$

Computational challenges of inference. Again, our main goal is to tractably compute expectations of functions under the posterior $\pi(\theta \mid D) \propto \mathcal{L}(Y \mid \theta, X)\pi_0(\theta)$. Since there are $\Theta(p^2)$ parameter components, direct numerical integration over each of these components is only feasible when p is at most 3 or 4. As a result we turn to Monte Carlo integration. Two natural Monte Carlo estimators one might use to approximate $\mathbb{E}_{\pi(\theta|D)}[f(\theta)]$ are

1. $\frac{1}{T}\sum_{t=1}^{T} f(\theta^{(t)})$ with $\theta^{(t)} \stackrel{\text{iid}}{\sim} \pi(\theta \mid D)$ or

2. $\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{\pi(\theta|D,\tau^{(t)})}[f(\theta)]$ with $\tau^{(t)} \stackrel{\text{iid}}{\sim} p(\tau \mid D)$.

For the first estimator, we can use Markov chain Monte Carlo (MCMC) techniques to sample each $\theta^{(t)}$ approximately independently from $\pi(\theta \mid D)$ since the posterior is available up to a multiplicative normalizing constant. Computing the prior $p(\theta)$, however, may be analytically intractable because it requires marginalizing out τ. We could instead additionally sample τ. To use gradient-based MCMC samplers, sampling τ would require computing the pdfs (and gradients) of the likelihood terms $\mathcal{L}(y^{(n)} \mid x^{(n)}, \theta, \sigma^2)$ and the prior terms $p(\theta \mid \tau)$ and $p(\tau)$. So the cost would be $O(p^2N + \dim(\tau))$ time per iteration. Even for p moderately large, the $\Theta(p^2 + \dim(\tau))$ number of parameters might require many MCMC iterations to properly explore such a large space (Beskos et al., 2013; MacKay, 1998; Pillai et al., 2012); see also Fig. 2 for an empirical demonstration.

To explore a smaller space, and hence potentially reduce the number of MCMC iterations required for the chains to mix, we might take the second approach: sampling from $p(\tau \mid D)$ by marginalizing out the high-dimensional parameter θ. Sampling each τ requires computing

$$p(D \mid \tau, \sigma^2) = \int p(D \mid \theta, \sigma^2) dp(\theta \mid \tau). \qquad (4)$$

Since $p(\theta \mid \tau)$ is a multivariate Gaussian density function, $p(D \mid \tau, \sigma^2)$ equals

$$\frac{(1/\sqrt{2\pi\sigma^2})^N \det(2\pi\Sigma_{N,\tau})^{\frac{1}{2}} \exp\left(-\frac{1}{2\sigma^2}Y^TY\right)}{\det(2\pi\Sigma_\tau)^{\frac{1}{2}} \exp\left(-\frac{1}{2\sigma^4}Y^T\Phi_2(X)\Sigma_{N,\tau}\Phi_2(X)^TY\right)} \qquad (5)$$

where $\Sigma_{N,\tau}^{-1} := \Sigma_\tau^{-1} + \frac{1}{\sigma^2}\Phi_2(X)^T\Phi_2(X)$. Unfortunately, computing Eq. (5) naively takes prohibitive $O(p^6 + p^2N)$ time – or $O(p^2N^2 + N^3)$ time when using linear algebra identities; see Table 1 and Appendix E for details.

4. The Kernel Interaction Sampler

Our *kernel interaction sampler* (KIS) provides a recipe for efficiently sampling from $p(\tau, \sigma^2 \mid D)$ using MCMC. Recall from the last section that the computational bottleneck for sampling τ was computing $p(D \mid \tau, \sigma^2)$, so we focus on that problem here. We achieve large computational gains (Table 1) by using the special model structure and a kernel trick to avoid the factorization of $\Sigma_{N,\tau}$ in Eq. (5). To that end, KIS has three main parts: (1) we re-parameterize the generative model given in Eq. (3) using a Gaussian process (GP); (2) we show how to cheaply compute the GP kernel; and (3) we show how these steps translate into computation of $p(D \mid \tau, \sigma^2)$ in time linear in p. In Appendix A we extend to the case of higher-order interactions.

For the moment, suppose that we could construct a covariance function k_τ such that the generative model in Eq. (3) could be re-parameterized as:

$$\begin{aligned} \tau &\sim p(\tau) \\ g \mid \tau &\sim GP(0, k_\tau) \\ \sigma^2 &\sim p(\sigma^2) \\ y^{(n)} \mid g, x^{(n)}, \sigma^2 &\sim \mathcal{N}(g(x^{(n)}), \sigma^2), \end{aligned} \qquad (6)$$

where $GP(0, k_\tau)$ denotes a Gaussian process (GP) with mean function zero. Defining the kernel matrix $(K_\tau)_{ij} := k_\tau(x^{(i)}, x^{(j)})$, we can conclude that (see Rasmussen & Williams, 2006, Eq. 2.30)

$$\log p(D \mid \tau, \sigma^2) = -\frac{1}{2}Y^T\alpha - \text{trace}(\log L) - \frac{N}{2}\log 2\pi, \qquad (7)$$

where L equals the Cholesky decomposition of $K_\tau + \sigma^2 I_N$ and $\alpha := L^T(L^{-1}Y)^{-1}$. Let T_k denote the time it takes

to evaluate k_τ on a pair of points. The computational bottleneck of Eq. (7) is computing and factorizing K_τ, which take $O(N^2T_k)$ and $O(N^3)$ time, respectively. Hence, as long as T_k is $O(p)$, we can compute $p(D \mid \tau, \sigma^2)$ in time *linear* in p. To achieve this scaling, we first show (in the next result) that *any* generative model in the form of Eq. (3) can be rewritten in the form of Eq. (6). We then show how k_τ can be evaluated in $O(p)$ time for the models of interest.

Proposition 4.1. *(Gaussian process representation) Let Y and $\tilde{Y}$ be response vectors generated according to the models in Eq. (3) and Eq. (6) respectively for design matrix $X \in \mathbb{R}^{N \times p}$. Let $k_\tau(x^{(i)}, x^{(j)}) = \Phi_2(x^{(i)})^\top \Sigma_\tau \Phi_2(x^{(j)})$. Then, $Y \mid X \overset{d}{=} \tilde{Y} \mid X$, where $\overset{d}{=}$ denotes equality in distribution. Moreover, for every draw $g \mid \tau \sim \mathcal{N}(0, k_\tau)$, there exists some $\theta \in \mathbb{R}^{\dim(\Phi_2)}$ such that $g(\cdot) = \theta^T\Phi_2(\cdot)$.*

The proof follows directly by considering the weight-space view of a GP (Rasmussen & Williams, 2006, Chapter 2); see Appendix B for details.

Next, we need to show that k_τ can be evaluated in $O(p)$ time for models of interest. This fact is not obvious; computing k_τ on a pair of points naively still requires explicitly computing the high-dimensional feature maps Φ_2 and prior covariance matrix Σ_τ. To compute k_τ efficiently, we rewrite it as a weighted sum of polynomial kernels of the form

$$k_{\text{poly},d}^c(x, \tilde{x}) := \left(x^T\tilde{x} + c\right)^d,$$

which each take $O(p)$ time to compute. Below we define *two-way interaction kernels* as particular linear combinations of these polynomial kernels. Then we provide a result motivating this class; namely, we show that any diagonal Σ_τ prior can be written as a two-way interaction kernel. Fortunately, to the best of our knowledge, all previous high-dimensional Bayesian regression models assume Σ_τ is diagonal (cf. Carvalho et al., 2009; Chipman, 1996; George & McCulloch, 1993; Griffin & Brown, 2017; Piironen & Vehtari, 2017). Hence, this restriction on Σ_τ is mild.

Definition 4.2. (Two-way interaction kernel) We call the kernel k a *two-way interaction kernel* if for some choice of $M_1, M_2 \in \mathbb{N}$, $\alpha, \psi, \lambda^{(m)} \in \mathbb{R}_+^p$ ($m = 1, \ldots, M_1$), $\nu^{(m)} \in \mathbb{R}_+$ ($m = 1, \ldots, M_2$), $1 \le i_m < j_m \le p$ ($m = 1, \ldots, M_2$), and $A \in \mathbb{R}$, the kernel $k(x, \tilde{x})$ is equal to

$$\sum_{m=1}^{M_1} k_{poly,2}^1(\lambda^{(m)} \odot x, \lambda^{(m)} \odot \tilde{x}) + \sum_{m=1}^{M_2} \nu^{(m)} x_{i_m} x_{j_m} \tilde{x}_{i_m} \tilde{x}_{j_m}$$

$$+ k_{poly,1}^A(\alpha \odot x, \alpha \odot \tilde{x}) + k_{poly,1}^0(\psi \odot x \odot x, \psi \odot \tilde{x} \odot \tilde{x}),$$

where $\odot$ is the entrywise product.

Theorem 4.3 (1-to-1 correspondence with diagonal Σ_τ). *Suppose k is a two-way interaction kernel. Then*

$$k(x, \tilde{x}) = \Phi_2(x)^\top S\Phi_2(\tilde{x}), \qquad (8)$$

where the induced prior covariance matrix S is diagonal. The entries of S are given by

$$\text{diag}(S)_{(i)} = \alpha_i^2 + 2\sum_{m=1}^{M_1}\left[\lambda_i^{(m)}\right]^2$$

$$\text{diag}(S)_{(ij)} = 2\sum_{m=1}^{M_1}\left[\lambda_i^{(m)}\lambda_j^{(m)}\right]^2 + \sum_{k:i_k=i,j_k=j}^{M_2}\nu^{(m)}$$

$$\text{diag}(S)_{(ii)} = \psi_i^2 + \sum_{m=1}^{M_1}\left[\lambda_i^{(m)}\right]^4$$

$$\text{diag}(S)_{(0)} = M_1 + A,$$

where $\text{diag}(S)_{(i)}$, $\text{diag}(S)_{(ij)}$, $\text{diag}(S)_{(ii)}$, and $\text{diag}(S)_{(0)}$ denote the prior variances of the main effect θ_{x_i}, interaction effect $\theta_{x_i x_j}$, quadratic effect $\theta_{x_i^2}$, and intercept θ_0, respectively. Furthermore, for any diagonal covariance matrix $S \in \mathbb{R}^{\dim(\Phi_2) \times \dim(\Phi_2)}$, there exists a two-way interaction kernel that induces S as a prior covariance matrix.

Theorem 4.3 (proof in Appendix B.2) and Proposition 4.1 imply that two-way interaction kernels induce a space of models in 1-to-1 correspondence with models in the form of Eq. (3) when Σ_τ is constrained to be diagonal. Since most models of practical interest have Σ_τ diagonal, we can readily construct the two-way interaction kernel corresponding to Σ_τ by solving the system of equations

$$\begin{aligned}\text{diag}(S)_{(i)} &= \text{diag}(\Sigma_\tau)_{(i)} \quad \text{diag}(S)_{(ij)} = \text{diag}(\Sigma_\tau)_{(ij)}\\ \text{diag}(S)_{(ii)} &= \text{diag}(\Sigma_\tau)_{(ii)} \quad \text{diag}(S)_{(0)} = \text{diag}(\Sigma_\tau)_{(0)}\end{aligned}$$

$$(9)$$

Each of the $M_1 + 2$ polynomial kernels takes $O(p)$ time to compute, and each of the M_2 product terms takes $O(1)$ time. Therefore, we want to select M_1 and M_2 small so that k_τ can be computed quickly. Since there are more degrees of freedom (i.e., free variables) available to solve Eq. (9) as M_1 and M_2 increase, eventually a solution will exist as we show in Appendix B.2. But Theorem 4.3 does not tell us how large M_1 and M_2 have to be for an arbitrary model. In Appendix C, we solve Eq. (9) for a variety of models of practical interest and show that in these cases, M_1 and M_2 can be set very small (between one and three). Thus k_τ can be computed in $O(p)$ time, and so the kernel matrix K_τ can be computed in $O(N^2 p)$ time. Finally, then, we may compute the likelihood $p(D \mid \tau, \sigma^2)$ in $O(N^2 p + N^3)$ time.

5. The Kernel Interaction Trick: Recovering Posterior Marginals

Even if we are able to sample τ much faster using KIS, the problem of computing $\mathbb{E}_{p(\theta|D,\tau)}[f(\theta)]$ remains unresolved. In this section, we show that, given K_τ, any such expectation can be recovered in $O(1)$ time by evaluating the GP posterior at certain test points.

To provide the main intuition for our solution, suppose we would like to compute the posterior mean of the main effect θ_{x_i}. Let $e_i \in \mathbb{R}^p$ denote the ith unit vector. Since $g = \theta^T \Phi_2$ by Proposition 4.1, we have

$$\begin{aligned}g(e_i) &= \theta_{x_i} + \theta_{x_i^2}\\ g(-e_i) &= -\theta_{x_i} + \theta_{x_i^2}\\ \frac{g(e_i) - g(-e_i)}{2} &= \theta_{x_i}.\end{aligned}$$

$$(10)$$

Since g is a Gaussian process, the distribution of $Z_g := (g(e_i), g(-e_i)) \mid D, \tau$ is multivariate Gaussian and can be computed in closed form by appropriate matrix multiplications of the kernel matrix K_τ; see Theorem 5.1 below for details. Then, by consulting Eq. (10), one can recover $\theta_{x_i} \mid D, \tau$ as the linear combination $[1/2, -1/2]^T Z_g \mid D, \tau$, which is univariate Gaussian. While we have focused on a particular instance here, this example provides the main insight for the general formula to compute $\mathbb{E}_{p(\theta|D,\tau)}[f(\theta)]$ from K_τ.

Theorem 5.1. *(The kernel interaction trick) Let $H_\tau := (K_\tau + \sigma^2 I_N)^{-1}$ and*

$$A_{ij} := [e_i, -e_i, e_j, e_i + e_j]^T \in \mathbb{R}^{4 \times p}.$$

Let $K_\tau(A_{ij}, X) = K_\tau(X, A_{ij})^T$ be the $4 \times N$ matrix formed by taking the kernel between each row of A_{ij} with each row of X. For a row-vector $a \in \mathbb{R}^4$, define the scalars $\mu_a := aK_\tau(A_{ij}, X)H_\tau Y$ and

$$\sigma_a^2 := a\left[K_\tau(A_{ij}, A_{ij}) - K_\tau(A_{ij}, X)H_\tau K_\tau(X, A_{ij})\right]a^T.$$

Then the distributions of $\theta_{x_i} \mid \tau, D$ and $\theta_{x_i x_j} \mid \tau, D$ are given by $\mathcal{N}(\mu_a, \sigma_a^2)$ with, respectively, $a = (1/2, -1/2, 0, 0)$ and $a = (-1/2, 1/2, -1, 1)$.

Corollary 5.2. *Given K_τ, the distributions of θ_{x_i} and $\theta_{x_i x_j}$ take $O(1)$ additional time and memory to compute.*

We prove Theorem 5.1 and Corollary 5.2 in Appendix B. In Appendix B.5, we generalize Theorem 5.1 by showing how to obtain the joint posterior distribution of any subset of parameters contained in θ. Hence, we can compute $\mathbb{E}_{p(\theta|D,\tau)}[f(\theta)]$ for an arbitrary f using the kernel interaction trick.

Note that if we would like to obtain the posterior mean of all $\Theta(p^2)$ parameters, then clearly a linear time algorithm in p is impossible. Instead, we can adopt a lazy evaluation strategy where we compute the posterior of one of the $\Theta(p^2)$ parameters only when it is needed. This approach is effective in the many applications where we do not need to look at all the interactions. In particular, we might first find the top k main effects. After selecting these variables, we could examine their interactions. The number of interactions among the main effects (which is $\Theta(k^2)$) is much

smaller than the total number of possible interactions (which is $\Theta(p^2)$) if $k \ll p$. Such a strategy is natural if we believe that θ satisfies the (commonly assumed) strong hierarchy restriction.

6. SKIM: Sparse Kernel Interaction Model

To demonstrate the utility and efficacy of the kernel interaction sampler and kernel interaction trick, we choose a particular model that we call the *sparse kernel interaction model (SKIM)*. In what follows, we first detail SKIM, which we will see promotes sparsity and strong hierarchy. Then, by observing that SKIM is a special case of the general model in Eq. (3), we can show that SKIM induces a two-way interaction kernel via Theorem 4.3 and Eq. (9). We will see that this kernel has only 3 components and thus takes only $O(p)$ time to evaluate. By Corollary B.3, we can compute the distribution of interaction terms from SKIM in $O(1)$ time once we have computed the kernel matrix. Hence, the final computation time for discovering main effects and interaction effects with SKIM will be $O(N^2 p + N^3)$ by the discussion at the end of Section 5.

SKIM is given in full detail, together with discussion of hyperparameter selection and intepretation, in Appendix D.1. It is a particular instance of a general class of hierarchical sparsity priors (cf. Chipman, 1996; George & McCulloch, 1993; Griffin & Brown, 2017) that have the following form:
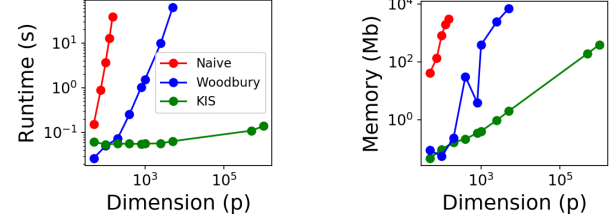
$$
\begin{aligned}
\kappa \sim p(\kappa) \quad \eta \sim p(\eta) \quad c^2 &\sim p(c^2) \\
\theta_{x_i} \mid \kappa, \eta &\sim \mathcal{N}(0, \eta_1^2 \kappa_i^2) \\
\theta_{x_i x_j} \mid \kappa, \eta &\sim \mathcal{N}(0, \eta_2^2 \kappa_i^2 \kappa_j^2) \\
\theta_{x_i^2} \mid \kappa, \eta &\sim \mathcal{N}(0, \eta_3^2 \kappa_i^4) \\
\theta_0 \mid c^2 &\sim \mathcal{N}(0, c^2),
\end{aligned}
\tag{11}
$$

where θ_0 is the intercept term and every η_i or κ_j is a scalar.

We next show that any prior in the form of Eq. (11) induces a $O(p)$ two-way interaction kernel. The proof is in Appendix B.6.

Proposition 6.1. *Taking $\tau := (\eta, \kappa, c^2)$, the generative model in Eq. (11) is equivalent to using the following kernel in Eq. (6):*

$$
\begin{aligned}
k_\tau(x, \tilde{x}) = \frac{\eta_2^2}{2} k_{poly,2}^1(\kappa \odot x, \kappa \odot \tilde{x}) \\
(\eta_3^2 - \frac{\eta_2^2}{2}) k_{poly,1}^0(\kappa \odot x \odot x, \kappa \odot \tilde{x} \odot \tilde{x}) \\
+ \left(\eta_1^2 - \eta_2^2\right) k_{poly,1}^0(\kappa \odot x, \kappa \odot \tilde{x}) + c^2 - \frac{\eta_2^2}{2}.
\end{aligned}
$$

Figure 1. Empirical evaluation of (a) time and (b) memory scaling with dimension of marginal likelihood computation. Woodbury and Naive refer to the baselines in Section 3.

7. Experiments

Time and memory cost versus Bayesian baselines. We first assess the computational advantages of our kernel interaction sampler (KIS) by comparing it with each baseline Bayesian method in Table 1. We start by profiling the time and memory cost of computing $p(D \mid \tau, \sigma^2)$, which we have seen is a computational bottleneck for sampler option 2 in Section 3. In Fig. 1, we depict the time and memory cost of $p(D \mid \tau, \sigma^2)$ computation for conjugate linear regression with an isotropic Gaussian prior on synthetic datasets with $N = 50$. We vary p but not N because we are interested primarily in the high-dimensional case when p is large relative to N. Fig. 1 shows that KIS yields orders-of-magnitude speed and memory improvements over the baseline methods for computing $p(D \mid \tau, \sigma^2)$.

We next compare inference for SKIM using KIS, which marginalizes out θ and samples τ, to jointly sampling (θ, τ) (denoted FULL).[2] We implemented KIS and FULL in `Stan` (Carpenter et al., 2019) and used the NUTS algorithm (Hoffman & Gelman, 2014) for sampling (4 chains with 1,000 iterations per chain). As shown in Fig. 2(a), KIS is orders of magnitude faster even for lower values of p. In Section 3 we remarked that since FULL explores a much higher-dimensional space, there might be issues with mixing. To explore this possibility empirically, we check the Gelman–Rubin statistic ($\hat{R}$) values of the output from both KIS and FULL. We found that, for FULL, the $\hat{R}$ values were greater than 1.05, with some reaching as high as 1.5 (indicating poor mixing), while for KIS all $\hat{R}$ values were less than 1.05 (suggesting good mixing).

Comparison to LASSO: synthetic data. Having demonstrated the considerable computational savings over baseline Bayesian approaches, we next demonstrate the advantage of our method over frequentist approaches such as the LASSO. In particular, we consider the common case when the true high-dimensional parameter θ is assumed to be sparse and satisfies the requirement of strong hierarchy. To the best of our knowledge, there has not been an extensive empiri-

---

[2]See the discussion of sampler option 1 in Section 3.

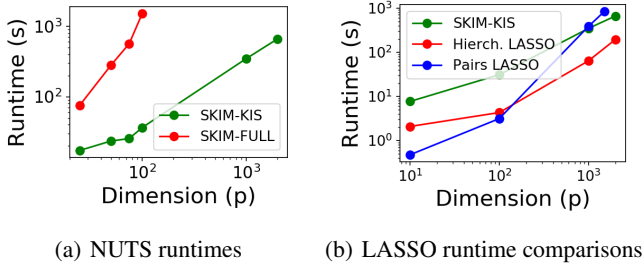(a) NUTS runtimes   (b) LASSO runtime comparisons

Figure 2. The left-hand figure indicates the time to complete four parallel chains of 1000 iterations of NUTS for the SKIM model proposed in Section 6 using KIS (denoted as SKIM-KIS) and FULL. For each point, KIS had $\hat{R} < 1.05$ while FULL always had $\hat{R} > 1.05$. The right-hand figure compares the runtime of inference for SKIM-KIS versus fitting LASSO-based methods.

cal comparison between sparse Bayesian interaction models and sparse frequentist interaction models. The likely reason is that each MCMC iteration for sampling τ takes $O(N^2p^2 + N^3)$ time using the Woodbury matrix method. The per-iteration cost of the iterative optimization solver for the LASSO and the hierarchical LASSO, on the other hand, is $O(Np^2)$, which is much faster when N is even moderately large. Fortunately, SKIM admits a cheap-to-compute kernel function such that each MCMC iteration takes $O(N^2p + N^3)$ time, which is *faster* than the LASSO-style approaches in cases when p is large relative to N.

We benchmark SKIM against generating all pairwise interactions and running the LASSO (denoted pairs LASSO) and the hierarchical LASSO (Lim & Hastie, 2015), which constrains the fitted parameters to satisfy strong hierarchy. We generate 36 different synthetic datasets, which differ in the number of observations, dimension, and signal-to-noise ratio. The covariates X are drawn from $\mathcal{N}(0, \lambda^2 I_p)$ for different choices of λ. Here, λ controls the signal-to-noise ratio; when λ is larger, the signal is stronger. We consider $N \in \{50, 100, 200\}$ observations, $p \in \{50, 100, 200, 500\}$ dimensions (which translates into between roughly 1.25×10^3 and 1.25×10^5 total interaction parameters), and $\lambda \in \{1, 2, 5\}$. In each dataset, we select five variables (and their pairwise interactions) to affect y, and we allow the rest of the variables to lead to spurious correlations with the response y. We set the magnitudes of all non-zero effects to 1. Finally, $y \mid x, \theta^* \sim \mathcal{N}(0, \sigma^2)$, where the noise variance σ^2 equals the largest λ^2 value, namely 25, to mimic the realistic case when the noise variance is large relative to the signal. We compare each method in terms of variable selection quality and mean-squared error (MSE) between the fitted and true parameter. For variable selection, we select a parameter only if the 99% posterior credible interval does not contain zero. For the hierarchical LASSO and pairs LASSO, the variables selected are those with non-zero coefficients, and we use 5-fold cross-validation to find the strength of the L_1 penalties. We fit
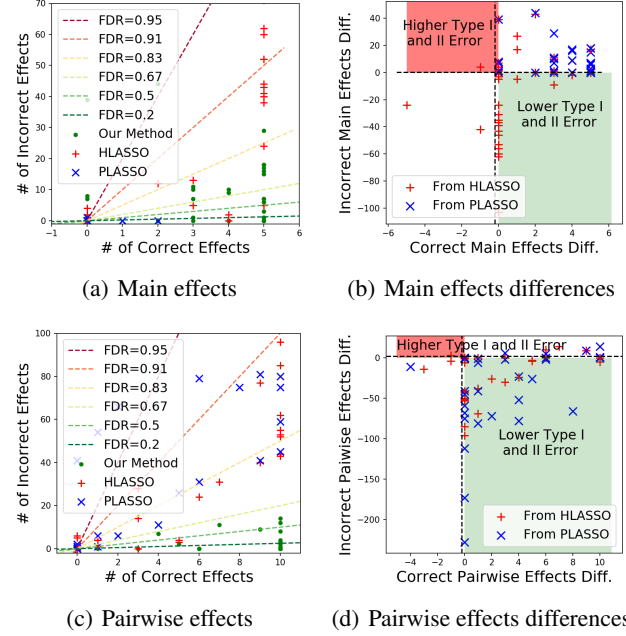

(a) Main effects   (b) Main effects differences

(c) Pairwise effects   (d) Pairwise effects differences

Figure 3. Variable selection performance of each method for the 36 synthetic datasets. Each point in each plot indicates one of these datasets for a particular method. The green regions in the second and last plot indicate where our method in strictly better than the other two in terms of variable selection, while the red region indicates the datasets for which our method is strictly worse. In the first and last figures, better performance occurs when moving right and/or down.

the hierarchical LASSO using the `glinternet` package in R and pairs LASSO using `sklearn` in `python`. We implemented KIS is `Stan` (4 chains with 1,000 iterations each). The $\hat{R}$ values for each dataset were less than 1.05.

First, we examine how well each method selects main effects and pairwise effects. Each point in Fig. 3(a) shows the number of main effects selected and number of incorrect main effects selected for a given synthetic dataset. In this plot, it is clear that our method has better false discovery rate (FDR) control over the other two methods on average. Fig. 3(c) shows the FDR performance for pairwise effects. To compare the methods at the dataset level, in Fig. 3(b,d) we consider the difference in the number of correct and incorrect main effects selected by our method and the LASSO methods for each dataset. The green shaded regions indicate the datasets for which our method simultaneously selects more correct main effects and has fewer incorrect main effects, i.e., is strictly better than the other two methods for any variable selection metric. Finally, in Fig. 4 we look at the difference in MSE to θ^*, broken down in terms of the error for estimating main and pairwise effects. Again, we see for the great majority of the datasets, KIS outperforms the LASSO based approaches. In Fig. 2(b) we see that SKIM-KIS has competitive runtimes relative to pairs LASSO and hierarchical LASSO.
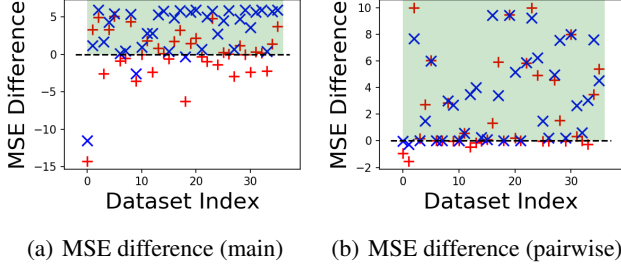
(a) MSE difference (main)   (b) MSE difference (pairwise)

Figure 4. Each red cross denotes the difference in MSE of the hierarchical LASSO and KIS from the true main effects (left) and pairwise effects (right) for a given synthetic dataset. When the MSE difference is larger than 0 (i.e., the green shaded region), then our method is closer to the true effect sizes in terms of Euclidean distance. Similarly, each blue x equals the difference in MSE of all-pairs LASSO and our method.

Table 2. Building dataset results. MAIN (PAIR) MSE refers to total error in estimating main (pairwise) effects. The main and pairwise MSE added together yield the total MSE. The second and fourth columns show (# of effects correctly selected) : (# of incorrect effects selected) for main and pairwise effects, respectively. Larger green values are better while larger purple values are worse.

METHOD	MAIN MSE	# MAIN	PAIR MSE	# PAIR
SKIM	0.1	3 : 0	7.0	3 : 0
PLASSO	5.0	2 : 5	9.3	3 : 21
HLASSO	1.5	3 : 19	7.8	3 : 18

Comparison to LASSO: synthetic data, real covariates.
To understand the impact of the geometry of the covariates on performance, we took the Residential Building Data Set from the UCI Machine Learning Repository and simulated responses as in our previous synthetic experimental setup. In particular, we randomly chose 5 variables and their 10 pairwise interactions to have non-zero effects. In this case, the covariates are highly correlated (the first 20 out of 109 principal components capture over 99% of the variance in the data). In Table 2, we see that SKIM significantly outperforms the LASSO-based methods for recovering main and pairwise effects.

Comparison to LASSO: cars miles per gallon dataset.
We conclude by comparing the methods on the Auto MPG dataset, from the UCI Machine Learning Repository, which contains $N = 398$ samples and $p = 8$ variables. We consider only the 6 numerical variables (cylinders, displacement, horsepower, weight, acceleration, model year) and standardize the data by subtracting the mean and dividing by the standard deviation. To compare the methods, we first fit SKIM and the LASSO-based methods (via 5-fold cross-validation) on these 6 features. Our method selects

Table 3. Auto MPG dataset results. Each column represents the (# of original effects selected) : (# of fake effects selected). A selected main (pairwise) effect is an "original" effect if it corresponds to one of the original 6 features (15 interactions). Main100 (Pairwise100) and Main200 (Pairwise200) denote when 100 and 200 random noise covariates are added to the original 6 features, respectively. Larger purple values are worse.

METHOD	MAIN100	MAIN200	PAIR100	PAIR200
SKIM	3 : 0	3 : 0	1 : 0	1 : 0
PLASSO	4 : 1	4 : 0	4 : 99	2 : 78
HLASSO	5 : 4	6 : 46	5 : 2	4 : 38

three main effects (weight, horsepower, acceleration) and one interaction (weight × horsepower). The hierarchical LASSO selects all six main effects and 8 out of the 15 possible pairwise interactions. Pairs LASSO selects 5 main effects and 8 interactions.

Since there is no ground truth, and all of the main and pairwise interactions could a priori affect miles per gallon, it is difficult to compare the methods. To better assess the methods, we instead append random noise covariates and refit each model. In particular, we draw additional covariates from a $\mathcal{N}(0, I_m)$, for $m = 100, 200$ and add these noise variables to the original 6 features. The total number of main and pairwise regression coefficients grows to 5,671 and 21,321 for $m = 100, 200$ respectively, making the regression task very high-dimensional. The results are summarized in Table 3. All methods are able to pick up some main effects and pairwise effects from the original dataset. Beyond that observation, we cannot compare which main and interaction effects from the original data are real. However, we do know that all noise effects are fake. We see that even with more noise directions, our method selects the same main effects and pairwise effects as the noiseless covariate case; that is, it does not pick up any fake effects. The two LASSO-based methods, on the other hand, incorrectly select many noise variables as interactions.

Conclusion. Through our kernel interaction sampler we have demonstrated that Bayesian interaction models can offer both competitive computational scaling relative to LASSO-based methods and improved Type I and II error rates. While our method runs in time linear in p per iteration, the cubic dependence on N still makes inference computationally challenging. Fortunately, there is a wide GP literature that deals precisely with reducing this cubic timing dependence through inducing points (Quiñonero Candela & Rasmussen, 2005; Titsias, 2009) or novel conjugate-gradient techniques (Gardner et al., 2018). An interesting future direction will be to empirically and theoretically understand the statistical penalty of using these inducing point methods to scale SKIM to the setting of both large N *and* large p.

Acknowledgements

References

Aschard, H. A perspective on interaction effects in genetic association studies. *Genetic Epidemiology*, 2016.

Barber, R. F. and Candès, E. J. Controlling the false discovery rate via knockoffs. *The Annals of Statistics*, 43(5): 2055–2085, 10 2015.

Benjamini, Y. and Speed, T. P. Summarizing and correcting the GC content bias in high-throughput sequencing. *Nucleic Acids Research*, 40(10), 2012.

Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J., and Stuart, A. Optimal tuning of the hybrid Monte Carlo algorithm. *Bernoulli*, 19(5A):1501–1534, 11 2013.

Bien, J., Taylor, J., and Tibshirani, R. A Lasso for hierarchical interactions. *The Annals of Statistics*, 41(3): 1111–1141, 2013.

Candes, E. and Tao, T. The Dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, pp. 2313–2351, 2007.

Carpenter, B., Lee, D., Brubaker, M. A., Riddell, A., Gelman, A., Goodrich, B., Guo, J., Hoffman, M., Betancourt, M., and Li, P. Stan: A probabilistic programming language, 2019.

Carvalho, C., Polson, N., and Scott, J. Handling sparsity via the horseshoe. In *International Conference on Artificial Intelligence and Statistics*, 2009.

Chen, S., Donoho, D., and Saunders, M. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, pp. 33–61, 1998.

Chipman, H. Bayesian variable selection with related predictors. *The Canadian Journal of Statistics*, 24(1):17–36, 1996.

Choy, S. and Chan, C. Scale mixtures distributions in insurance applications. *Journal of the IAA*, 33:93–104, 2003.

Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D., and Wilson, A. G. GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *Advances in Neural Information Processing Systems*. 2018.

Gelman, A., Hill, J., and Yajimam, M. Why we (usually) don't have to worry about multiple comparisons. *Journal of Research on Educational Effectiveness*, pp. 189–211, 2012.

George, E. and McCulloch, R. Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.

Griffin, J. and Brown, P. Hierarchical shrinkage priors for regression models. *Bayesian Analysis*, 12:135–159, 2017.

Hamdan, H., Nolan, J., Wilson, M., and Dardia, K. Using scale mixtures of normals to model continuously compounded returns. *Journal of Modern Applied Statistical Methods*, 2005.

Hoffman, M. and Gelman, A. The No-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1): 1593–1623, 2014.

Lim, M. and Hastie, T. Learning interactions via hierarchical group-lasso regularization. *Journal of Computational and Graphical Statistics*, 24(3):627–654, 2015.

Lipkovich, I., Dmitrienko, A., and D'Agostino, R. Tutorial in biostatistics: data-driven subgroup identification and analysis in clinical trials. *Statistics in Medicine*, 36:136–196, 2017.

MacKay, D. *Introduction to Monte Carlo Methods*, pp. 175–204. Springer Netherlands, 1998.

Morota, G. and Gianola, D. Kernel-based whole-genome prediction of complex traits: a review. *Frontiers in Genetics*, 5:363, 2014.

Nakagawa, K., Suzumura, S., Karasuyama, M., Tsuda, K., and Takeuchi, I. Safe pattern pruning: An efficient approach for predictive pattern mining. In *International Conference on Knowledge Discovery and Data Mining*, 2016.

Piironen, J. and Vehtari, A. Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electronic Journal of Statistics*, 11:5018–5051, 2017.

Pillai, N. S., Stuart, A. M., and Thiéry, A. H. Optimal scaling and diffusion limits for the Langevin algorithm in high dimensions. *The Annals of Applied Probability*, 22 (6):2320–2356, 12 2012.

Quiñonero Candela, J. and Rasmussen, C. E. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2006.

Risso, D., Schwartz, K., Sherlock, G., and Dudoit, S. Gc-content normalization for rna-seq data. *BMC Bioinformatics*, 12(1):480, Dec 2011.

Shah, R. Modelling interactions in high-dimensional data with backtracking. *Journal of Machine Learning Research*, 17(207):1–31, 2016.

Slim, L., Chatelain, C., Azencott, C., and Vert, J. Novel methods for epistasis detection in genome-wide association studies. *bioRxiv:325993*, 2018.

Su, G., Christensen, O. F., Ostersen, T., Henryon, M., and Lund, M. S. Estimating additive and non-additive genetic variances and predicting genetic merits using genome-wide dense single nucleotide polymorphism markers. *PLOS ONE*, 7(9):1–7, 09 2012.

Tibshirani, R. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, pp. 267–288, 1994.

Titsias, M. K. Variational learning of inducing variables in sparse gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 567–574, 2009.

Wainwright, M. and Simoncelli, E. Scale mixtures of Gaussians and the statistics of natural images. In *International Conference on Neural Information Processing Systems*, 1999.

Weissbrod, O., Geiger, D., and Rosset, S. Multikernel linear mixed models for complex phenotype prediction. *Genome Research*, 2016.

Wu, T., Chen, Y., Hastie, T., Sobel, E., and Lange, K. Genome-wide association analysis by Lasso penalized logistic regression. *Bioinformatics*, 25(6):714–721, 2009.

A. Modeling Multi-Way Interactions

In certain applications, we might expect that there are interactions of order greater than two. For example, suppose we are trying to predict college admissions. Then, we might expect a three-way interaction between a candidate's SAT score, GPA, and extracurricular involvement. Individually, these variables might only exhibit moderate association but together they could have a multiplicative effect. For example, we might expect that candidates who have high SAT scores, high GPAs, and excellent extracurricular activities will be accepted with near certainty, while candidates who only possess one/two of these qualities are borderline applicants.

We now show how to extend our results to handle such three-way, or more generally, *r-way interactions*.

Definition A.1. (*r*-way interactions) The *r-way interactions* of a covariate vector $x \in \mathbb{R}^p$ are generated from the feature map

$$\Phi_r(x) := \bigoplus_{d=1}^{r} \bigoplus_{k:k_1+\cdots+k_p=d} \prod_{j=1}^{p} x_j^{k_j}, \quad k \in \mathbb{N}^p,$$

where $\bigoplus_{j=1}^{m} a_j := (a_{11}, \cdots, a_{1k_1}, \cdots, a_{m1}, \cdots, a_{mk_m})$ denotes the concatenation of vectors $a_j \in \mathbb{R}^{k_j}$.

To model *r*-way interactions, we must use degree r polynomial kernels to generate all the necessary interactions. Hence, we recommend using the following generalized two-way interaction kernel, which we call the *r*-way interaction kernel.

Definition A.2. (*r*-way interaction kernel) A kernel k is called an *r-way interaction kernel* if for some choice of $M_1, M_2, M_3 \in \mathbb{N}, \alpha, \psi, \lambda^{(m)} \in \mathbb{R}_+^p$ $(m = 1, \ldots, M_1)$, $\nu^{(m)} \in \mathbb{R}_+$ $(m = 1, \ldots, M_2)$, and $\nabla^{(m)} \in \mathbb{R}_+^p$ $(k = 1, \ldots, M_3)$ it can be re-expressed as

$$\sum_{m=1}^{M_1} k_{poly,r}^1(\lambda^{(m)} \odot x, \lambda^{(m)} \odot y) + \sum_{m=1}^{M_2} \nu^{(m)} \left[\prod_{s=1}^{r} x_{i_{s_m}} \prod_{s=1}^{r} y_{i_{s_m}} \right] + \sum_{m=1}^{M_3} k_{r-1}(\nabla^{(m)} \odot x, \nabla^{(m)} \odot y),$$

where $\odot$ is the Hadamard product and k_{r-1} is an $r-1$ degree interaction kernel. The base case kernel (i.e., when $r = 2$) is provided in Definition 4.2.

To select the weights for an *r*-way interaction kernel, we must solve a system of equations similar to Eq. (9), except for a target prior covariance matrix $\Sigma_\tau \in \mathbb{R}^{\dim(\Phi_r) \times \dim(\Phi_r)}$.

B. Proofs

B.1. Proof of Proposition 4.1

Let $g(\cdot) = \theta^T \Phi_2(\cdot)$ and $\theta \mid \tau \sim \mathcal{N}(0, \Sigma_\tau)$. Then, $y^{(n)} = g(x^{(n)}) + \epsilon^{(n)}$. The first claim follows by taking $\phi = \Phi_2$ and $f = g$ in Rasmussen & Williams (2006, Equation 2.12).

The second claim follows directly from the duality between the weight-space and function-space view of a GP (Rasmussen & Williams, 2006, Chapter 2).

B.2. Proof of Theorem 4.3

The proof of Theorem 4.3 depends critically on Lemma B.1 below, which characterizes the relation between adding two kernels and the resulting induced prior covariance matrix.

Lemma B.1. *Let k_1 and k_2 be two kernels such that there exists vectors $a^{(1)}, a^{(2)} \in \mathbb{R}^{dim(\Phi_2)}$ for which $k_i(x,y) = \langle a^{(i)} \odot \Phi_2(x), a^{(i)} \odot \Phi_2(y) \rangle$. Let $k_3(x,y) = k_1(x,y) + k_2(x,y)$. Then,*

$$k_3(x,y) = \langle \Sigma_3^{\frac{1}{2}} \Phi_2(x), \Sigma_3^{\frac{1}{2}} \Phi_2(y) \rangle \quad s.t. \quad \Sigma_3 = diag(a^{(1)} \odot a^{(1)} + a^{(2)} \odot a^{(2)}). \tag{12}$$

Proof. By the sum property of kernels,

$$
\begin{aligned}
k_1(x,y) + k_2(x,y) &= \langle [a_1 \; a_2] \odot [\Phi_2(x) \; \Phi_2(x)], [a_1 \; a_2] \odot [\Phi_2(y) \; \Phi_2(y)] \rangle \\
&= \langle a^{(1)} \odot \Phi_2(x), a^{(1)} \odot \Phi_2(y) \rangle + \langle a^{(2)} \odot \Phi_2(x), a^{(2)} \odot \Phi_2(y) \rangle \\
&= \langle a^{(1)} \odot a^{(1)} \odot \Phi_2(x), \Phi_2(y) \rangle + \langle a^{(2)} \odot a^{(2)} \odot \Phi_2(x), \Phi_2(y) \rangle \\
&= \langle a^{(1)} \odot a^{(1)} \odot \Phi_2(x) + a^{(2)} \odot a^{(2)} \odot \Phi_2(x), \Phi_2(y) \rangle \\
&= \langle (a^{(1)} \odot a^{(1)} + a^{(2)} \odot a^{(2)}) \odot \Phi_2(x), \Phi_2(y) \rangle \\
&= \Phi_2^T(x) \operatorname{diag}((a^{(1)} \odot a^{(1)} + a^{(2)} \odot a^{(2)}) \; \Phi_2(y) \\
&= k_3(x,y).
\end{aligned}
\tag{13}
$$

$\square$

By Lemma B.1, it suffices to write out the feature map of each kernel in Definition 4.2. The induced feature maps of each respective kernel term in Definition 4.2 are given by $a_i \odot \Phi_2(x), 1 \le i \le 4$ for

$$
\begin{aligned}
a_1 &:= ((\lambda_1^{(m)})^2, \cdots, (\lambda_p^{(m)})^2, \sqrt{2}\lambda_1^{(m)}\lambda_2^{(m)}, \cdots, \sqrt{2}\lambda_{p-1}^{(m)}\lambda_p^{(m)}, \sqrt{2}\lambda_1^{(m)}, \cdots, \sqrt{2}\lambda_p^{(m)}, 1) \\
a_2 &:= (0, \cdots, 0, 0, \cdots, 0, \alpha_1, \cdots, \alpha_p, \sqrt{A}) \\
a_3 &:= (\psi_1, \cdots, \psi_p, 0, \cdots, 0, 0, \cdots, 0, 0) \\
a_4 &:= (0, \cdots, 0, 0, \cdots, 0, \sqrt{\nu^{(m)}}, 0, \cdots, 0, 0, \cdots, 0, 0)
\end{aligned}
\tag{14}
$$

The first claim follows from Eq. (14) and Lemma B.1.

To prove the second claim, take an arbitrary diagonal prior covariance matrix $S \in \mathbb{R}^{\dim(\Phi_2) \times \dim(\Phi_2)}$. It suffices to show that there exists a solution of,

$$
\operatorname{diag}(S)_{(i)} = \alpha_i^2 + 2 \sum_{m=1}^{M_1} \left[\lambda_i^{(m)} \right]^2
$$

$$
\operatorname{diag}(S)_{(ij)} = 2 \sum_{m=1}^{M_1} \left[\lambda_i^{(m)} \lambda_j^{(m)} \right]^2 + \sum_{m: i_m=i, j_m=j}^{K_2} \nu^{(m)}
$$

$$
\operatorname{diag}(S)_{(ii)} = \psi_i^2 + \sum_{m=1}^{M_1} \left[\lambda_i^{(m)} \right]^4
$$

$$
\operatorname{diag}(S)_{(0)} = M_2 + A.
$$

for some choice of $M_1, M_2 \in \mathbb{N}$, $\alpha, \psi, \lambda^{(m)} \in \mathbb{R}_+^p$ ($m = 1, \ldots, M_1$), $\nu^{(m)} \in \mathbb{R}_+$ ($m = 1, \ldots, M_2$), and $A \in \mathbb{R}$. Take $\alpha_i^2 = \operatorname{diag}(S)_{(i)}$ and $\psi_i^2 = \operatorname{diag}(S)_{(ii)}$, for $i = 1, \cdots, p$. Take $\lambda^{(m)} = 0$. Let $M_2 = \frac{p(p-1)}{2}$ and $\nu^{(1)} = \operatorname{diag}(S)_{(12)}, \cdots, \nu^{(M_2)} = \operatorname{diag}(S)_{((p-1)p)}$. Finally, letting $A = \operatorname{diag}(S)_{(0)} - M_2$ solves the system.

Remark. While we have shown one of the *many* ways to solve the above system for an arbitrary S, the strategy taken above is not practically useful; computing the kernel in this fashion will take $\Theta(p^2)$ time because $M_2 = \Theta(p^2)$. In practice, we must leverage the polynomial kernels (i.e., those in the M_1 sum) to avoid making M_2 large. We show how such a strategy works in Appendix C.

B.3. Proof of Theorem 5.1

Define $g(A^{ij}) := (g(e_i), g(-e_i), g(e_j), g(e_{ij}))$. Then,

$$
\begin{aligned}
g(A^{ij}) \mid D, \tau &\sim \mathcal{N}(\mu_{g_{ij}}, \Sigma_{ij}) \quad \text{s.t.} \quad \mu_{g_{ij}} := K_\tau(A^{ij}, X) H_\tau Y, \\
\Sigma_{ij} &:= \left[K_\tau(A^{ij}, A^{ij}) - K_\tau(A^{ij}, X) H_\tau K_\tau(X, A^{ij}) \right],
\end{aligned}
\tag{15}
$$

which follows directly from Rasmussen & Williams (2006, Equation 2.21). Notice that,

$$
\theta_{x_i} = \frac{g(e_1)}{2} - \frac{g(-e_1)}{2} = a_i^T g(A^{ij}) \quad \text{and} \quad \theta_{x_i x_j} = \frac{g(e_1)}{2} - \frac{g(-e_1)}{2} - g(e_j) + g(e_{ij}) = a_{ij}^T g(A^{ij}),
\tag{16}
$$

where $a_i = (1/2, -1/2, 0, 0)$ and $a_{ij} = (-1/2, 1/2, -1, 1)$. The proof follows from Eq. (15), Eq. (16), and recalling that an affine transformation $h : x \mapsto Ax$ of a multivariate Gaussian distribution $Z \sim \mathcal{N}(\mu, \Sigma)$ is given by $h(Z) \sim \mathcal{N}(A\mu, A\Sigma A^T)$.

B.4. Proof of Corollary 5.2

Corollary 5.2 follows immediately once we can show that $K_\tau(A_{ij}, X)$ takes $O(1)$ time. It suffices to show $k_\tau(x^{(n)}, e_i)$ and $k_\tau(x^{(n)}, e_i + e_j)$ take $O(1)$ time. Since k_τ is a sum of polynomial kernels, $k_\tau(x, y)$ only depends on $x, y \in \mathbb{R}^p$ through the inner product $x^T y$. Hence, for vectors $\tilde{x}, \tilde{y} \in \mathbb{R}^M$, $k_\tau(\tilde{x}, \tilde{y})$ is well-defined and just depends on $\tilde{x}^T \tilde{y}$. Now, $k_\tau(x^{(n)}, e_i) = k_\tau(x_i^{(n)}, 1)$ and $k_\tau(x^{(n)}, e_i + e_j) = k_\tau((x_i^{(n)}, x_j^{(n)}), (1, 1))$. Since $k_\tau(x_i^{(n)}, 1)$ and $k_\tau((x_i^{(n)}, x_j^{(n)}), (1, 1))$ do not depend on p, these terms each take $O(1)$ time to compute.

B.5. The General Kernel Interaction Trick

In this section, we generalize the kernel interaction trick, namely show how to access the distribution of arbitrary components of θ. First, we require some new notation. For $E \subseteq \{1, \cdots, p\}$, $|E| = M$, define

$$\theta_E := (\theta_{x_{i_1}}, \cdots, \theta_{x_{i_M}}, \theta_{x_{i_1} x_{i_2}}, \cdots, \theta_{x_{i_{M-1}} x_{i_M}}), \quad i_j \in E. \tag{17}$$

We show how to compute $\theta_E \mid \tau, D$ from the GP posterior predictive distribution. Without any lost of generality, we may assume $E = \{1, \cdots, M\}$ by relabeling the covariates.

Theorem B.2. *(General kernel interaction trick) Let* $H_\tau := (K_\tau + \sigma^2 I_N)^{-1}$ *and*

$$A_M := [e_1, -e_1, \cdots e_M, -e_M, e_1 + e_2, \cdots, e_{M-1} + e_M]^T.$$

Let $K_\tau(A_M, X) = K_\tau(X, A_M)^T$ *be the matrix formed by taking the kernel between each row of A_M with each row of X. Let*

$$a_i := (0, 0, \cdots, 1/2, -1/2, \cdots, 0, 0, \cdots, 0) \in \mathbb{R}^{2M + \frac{M(M-1)}{2}}$$
$$a_{ij} := (0, 0, \cdots, 1/2, -1/2, \cdots, -1, \cdots, 0, 0, \cdots, 1, \cdots, 0) \in \mathbb{R}^{2M + \frac{M(M-1)}{2}} \tag{18}$$

for $i < j$. That is, a_i has non-zero entries at e_i and $-e_i$ and a_{ij} has non-zero entries at e_i, $-e_i$, $-e_j$, and $e_i + e_j$. Let

$$R_M := [a_1 \cdots a_M \ a_{12} \cdots a_{(M-1)M}]^T. \tag{19}$$

Then, $\theta_E \mid \tau, D$ is a multivariate Gaussian distribution with mean $R_M K_\tau(A_M, X) H_\tau Y$ and covariance matrix

$$R_M \left[K_\tau(A_{ij}, A_{ij}) - K_\tau(A_{ij}, X) H_\tau K_\tau(X, A_{ij}) \right] R_M^T.$$

Proof. Following the proof of Theorem 5.1,

$$g(A^M) \mid D, \tau \sim \mathcal{N}(\mu_{g_M}, \Sigma_M) \quad \text{s.t.} \quad \mu_{g_M} := K_\tau(A^M, X) H_\tau Y,$$
$$\Sigma_M := \left[K_\tau(A^M, A^M) - K_\tau(A^M, X) H_\tau K_\tau(X, A^M) \right]. \tag{20}$$

Similar to Eq. (16),

$$\theta_{x_i} = \frac{g(e_1)}{2} - \frac{g(-e_1)}{2} = a_i^T g(A^M) \quad \text{and} \quad \theta_{x_i x_j} = \frac{g(e_1)}{2} - \frac{g(-e_1)}{2} - g(e_j) + g(e_{ij}) = a_{ij}^T g(A^M). \tag{21}$$

The proof follows from Eq. (20), Eq. (21), and recalling that an affine transformation $h : x \mapsto R_M^T x$ of a multivariate Gaussian distribution $Z \sim \mathcal{N}(\mu, \Sigma)$ is given by $h(Z) \sim \mathcal{N}(R_M \mu, R_M \Sigma R_M^T)$.

$\square$

Corollary B.3. *Given K_τ, the distribution $\theta_E \mid \tau, D$ takes $O(M^2)$ time and memory to compute.*

Proof. The proof is identical to the one provided in Appendix B.4.

$\square$

B.6. Proof of Proposition 6.1

See Appendix C.2.

C. Example Bayesian Interaction Models

In the following subsections, we show how to solve Eq. (9) for several classes of models.

C.1. Block-Degree Priors

Suppose we would like to set the prior variance of all terms with the same degree equal. That is, we would like to use a prior of the form,

$$
\begin{aligned}
\eta \in \mathbb{R}^3 &\sim p(\eta) \\
\theta_{x_i} \mid \eta &\sim \mathcal{N}(0, \eta_1^2) \\
\theta_{x_i x_j} \mid \eta &\sim \mathcal{N}(0, \eta_2^2) \\
\theta_{x_i^2} \mid \eta &\sim \mathcal{N}(0, \eta_3^2) \\
\theta_0 \mid \eta &\sim \mathcal{N}(0, c^2)
\end{aligned}
\tag{22}
$$

To find the corresponding kernel, let $\lambda = (\frac{1}{\sqrt[4]{2}}\sqrt{\eta_2}, \cdots, \frac{1}{\sqrt[4]{2}}\sqrt{\eta_2})$, $M_1 = 1$ and $M_2 = 0$. Then, $\operatorname{diag}(S)_{(ij)} = \eta_2^2$. Setting $\psi_i^2 = \eta_3^2 - \frac{1}{2}\eta_2^2$, implies that $\operatorname{diag}(S)_{(ii)} = \eta_3^2$. Finally, letting $\alpha_i^2 = \tau_1^2 - \frac{2\eta_2}{\sqrt{2}}$ and $A = c^2 - 1$ implies that $\operatorname{diag}(S)_{(i)} = \eta_1^2$ and $\operatorname{diag}(S)_{(0)} = c^2$ as desired. We may equivalently re-write the induced kernel as,

$$
k_{block,\eta}(x,y) = \frac{\eta_2^2}{2} k_{\text{poly},2}^1(x,y) + (\eta_3^2 - \frac{\eta_2^2}{2}) k_{\text{poly},1}^0(x \odot x, y \odot y) + \left(\eta_1^2 - \eta_2^2\right) k_{\text{poly},1}^0(x,y) + c^2 - \frac{\eta_2^2}{2}.
\tag{23}
$$

Hence, Eq. (22) admits a kernel that only takes $O(p)$ time to compute.

C.2. Sparsity Priors

By Lemma B.1, the sparsity prior model provided in Eq. (11) equals $k_{block,\eta}(\kappa \odot x, \kappa \odot y)$.

D. SKIM Model Details

The full hierarchal form of SKIM is provided below, which is based closely on the *regularized horseshoe prior* (Piironen & Vehtari, 2017) and the model proposed in Griffin & Brown (2017):

$$
\begin{aligned}
m^2 &\sim \text{InvGamma}(\alpha_1, \beta_1) & \xi^2 &\sim \text{InvGamma}(\alpha_2, \beta_2) \\
\phi &:= \frac{s}{p-s}\frac{\sigma}{\sqrt{N}} & \sigma &\sim N^+(0, \alpha_3) \\
\kappa_i &= \frac{m\lambda_i}{\sqrt{m^2 + \eta_1^2\lambda_i^2}} & \lambda_i &\sim C^+(0,1) \\
\eta_1 &\sim C^+(0, \phi) & \eta_2 &\sim \frac{\eta_1^2}{m^2}\xi \\
\theta_{x_i} \mid \eta, \kappa &\sim \mathcal{N}(0, \eta_1^2\kappa_i^2) \\
\theta_{x_j} \mid \eta, \kappa &\sim \mathcal{N}(0, \eta_1^2\kappa_j^2) \\
\theta_{x_i x_j} \mid \eta, \kappa &\sim \mathcal{N}(0, \eta_2^2\kappa_i^2\kappa_j^2) \\
\theta_0 \mid \eta &\sim \mathcal{N}(0, c^2)
\end{aligned}
$$

where s, α_i, and β_i are user-specified hyperparameters, $C^+(0,1)$ is a half-Cauchy distribution, and N^+ is a half-normal distribution. More details, such as selecting the hyperparameters, desirable properties, and interpretations of SKIM, are provided below.

D.1. SKIM Details

Recall that we are primarily interested in the case when θ is sparse and satisfies strong-hierarchy. In order to promote sparsity in the main effects, we require two ingredients: (1) a prior on the *global shrinkage* parameter η_1 and (2) a prior on the *local shrinkage* parameters $\kappa \in \mathbb{R}^p$, which together express the prior variance of θ_{x_i} as (Carvalho et al., 2009; Piironen & Vehtari, 2017):

$$\theta_{x_i} \mid \kappa, \eta_1 \sim \mathcal{N}(0, \eta_1^2 \kappa_i^2), \quad i = 1, \cdots, p. \tag{24}$$

η_1 controls the overall sparsity level of the model; in particular, the model becomes more sparse as η_1 decreases. If we expect s non-zero main effects, then setting $\eta_1 = \frac{s}{p-s} \frac{\sigma}{\sqrt{N}}$ will yield an expected prior sparsity level of s by Piironen & Vehtari (2017, Equation 3.12). However, we often do not know exactly how to select s. Hence, Piironen & Vehtari (2017) instead suggest drawing,

$$\phi := \frac{s}{p-s} \frac{\sigma}{\sqrt{N}} \qquad \eta_1 \sim C^+(0, \phi), \tag{25}$$

to express our uncertainty of not knowing the true main effect sparsity level.

The prior variance of θ_{x_i} is non-negligible only when κ_i is large enough to escape the global shrinkage of η_1. Hence, we want to draw κ_i from a heavy-tailed distribution so that certain main effects can escape global shrinkage. Carvalho et al. (2009) suggest drawing κ_i from a half-Cauchy distribution since this distribution has fat tails. However, such a prior often leads to undesirable numerical stability issues when using NUTS (Piironen & Vehtari, 2017). As a result, Piironen & Vehtari (2017) instead propose using a *regularized horseshoe* prior which truncates the half-Cauchy distribution to have support only on $[0, m)$ instead of $[0, \infty)$. This truncation (empirically) turns out to lead to better mixing properties, and is equivalent to the following generative mechanism:

$$\kappa_i = \frac{m\lambda_i}{\sqrt{m^2 + \eta_1^2 \lambda_i^2}} \qquad \lambda_i \sim C^+(0, 1) \tag{26}$$

As $\lambda_i \to \infty$, $\kappa_i \to \frac{m}{\eta_1}$. Hence, as $\lambda_i \to \infty$, the prior variance of θ_{x_i} equals m. Since we might not know the scale m of the non-zero main effects, we place a prior on m, namely,

$$m^2 \sim \text{InvGamma}(\alpha_1, \beta_1) \tag{27}$$

for hyperparameters α_1 and α_2.

Next, we model the interactions. If strong-hierarchy holds, sparsity comes for free; if there are only $s \ll p$ non-zero main effects, then there are at most $\frac{s(s-1)}{2} \ll p^2$ possible pairwise interactions. We must be careful, however, because strong-hierarchy trivially holds; our main effect estimates will, with probability one, never equal zero because the prior variances of the main effects are greater than 0 with probability one by Eq. (24) and our choice of priors. Instead, we aim for a relaxed version of strong-hierarchy. Namely, that the prior variance of an interaction $\theta_{x_i x_j}$ is large only if θ_{x_i} and θ_{x_j} are both large. θ_{x_i} and θ_{x_j} are large only when κ_i and κ_j are large. Hence, it suffices to make the prior variance of $\theta_{x_i x_j}$ large only when κ_i and κ_j are both large. Let $\tilde{\kappa}_i^2 = \frac{\eta_1^2}{m^2} \kappa_i^2$. Then, $0 \le \tilde{\kappa}_i^2 \le 1$ and $\tilde{\kappa}_i$ approaches 1 as $\lambda_i \to \infty$. Since, $\tilde{\kappa}_i^2$ and $\tilde{\kappa}_i^2$ are bounded by 1, $\tilde{\kappa}_i^2 \tilde{\kappa}_i^2$ will only be close to 1 when each term is close to one. That is, when both λ_i *and* λ_j are large, or equivalently when κ_i and κ_j are both large. Hence, it suffices to let

$$\theta_{x_i x_j} \mid \eta_1, \kappa \sim \mathcal{N}(0, \xi^2 \tilde{\kappa}_j^2 \tilde{\kappa}_j^2)$$
$$= \mathcal{N}(0, \eta_2^2 \kappa_i^2 \kappa_j^2) \quad \text{for} \quad \eta_2 := \frac{\eta_1^2}{m^2} \xi, \tag{28}$$

to promote strong-hierarchy, where ξ has the interpretation of the scale of the non-zero interaction effects; as λ_i and λ_j tend to infinity, the prior variance of $\theta_{x_i x_j}$ approaches ξ^2. Again, since we might not now this scale, we draw

$$\xi^2 \sim \text{InvGamma}(\alpha_2, \beta_2), \tag{29}$$

for some choice of hyperparameters α_2 and β_2.

E. Woodbury Identity and the Matrix Determinant Lemma

To compute the determinant in Eq. (5), one can perform a Cholesky decomposition of $\Sigma_{N,\tau} \in \mathbb{R}^{\dim(\Phi_2) \times \dim(\Phi_2)}$, which costs $O(Np^2 + p^6)$ time and $O(Np^2 + p^4)$ memory. This factorization can be avoided through the Woodbury matrix lemma and matrix determinant lemma.

The Woodbury matrix identity implies that,

$$(A^{-1} + UU^T)^{-1} = A - AU(I_K + U^T AU)^{-1}U^T A, \tag{30}$$

where $A \in \mathbb{R}^{M \times M}$, $U \in \mathbb{R}^{M \times K}$, and I_K is the $K \times K$ identity matrix. The matrix determinant lemma implies that,

$$\det(A^{-1} + UU^T) = \det(I + U^T AU)\det(A^{-1}) \tag{31}$$

Then, by the Woodbury identity,

$$\Sigma_{\tau,N} = (\Sigma_\tau^{-1} + \frac{1}{\sigma^2}\Phi_2(X)^T\Phi_2(X))^{-1} = \Sigma_\tau - \Sigma_\tau\Phi_2(X)^T(I_N + \Phi_2(X)\Sigma_\tau\Phi_2(X)^T)^{-1}\Phi_2(X)\Sigma_\tau. \tag{32}$$

Computing $p(D \mid \tau)$ requires computing $\det(\Sigma_{\tau,N})$. By the matrix determinant lemma,

$$\det(\Sigma_{\tau,N}) = (\det(I_N + \Phi_2(X)\Sigma_\tau\Phi_2(X)^T)\det(\Sigma_\tau^{-1}))^{-1}. \tag{33}$$

When Σ_τ is diagonal, the determinant equals the product of the diagonal, and its inverse equals one over the diagonal. Both of these quantities can be computed in $O(p^2)$ time. Hence, the time complexity for computing $\det(\Sigma_{\tau,N})$ is dominated by computing $\det(I_N + \Phi_2(X)\Sigma_\tau\Phi_2(X)^T)$, which takes $O(N^2p^2 + N^3)$ time and $O(Np^2)$ memory to store $\Phi_2(X)$.

F. Standard Polynomial Kernel

The feature map induced by the standard degree two polynomial kernel is given by

$$\begin{aligned}\Phi_{\text{poly},2}^c(x) &:= (x_1^2, \cdots, x_p^2, \sqrt{2}x_1x_2, \cdots, \sqrt{2}x_{p-1}x_p, \sqrt{2c}x_1, \cdots, \sqrt{2c}x_p, c) \\ &= a_{poly,2} \odot \Phi_2(x), \ a_{poly,2} := (1, \cdots, 1, \sqrt{2}, \cdots, \sqrt{2}, \sqrt{2c}, \cdots, \sqrt{2c}, c).\end{aligned} \tag{34}$$

Hence, Eq. (34) implies that

$$\text{diag}(\Sigma_{poly,2}) = a_{poly,2} \odot a_{poly,2}. \tag{35}$$

Eq. (35) shows that the prior covariance of the interaction terms are given higher prior variance than the main effects when $c \leq 1$, which is often undesirable. Furthermore, this prior does not promote sparsity, which is typically expected in high-dimensional problems.