Diverse Representative Rashomon Sets for Sparse Generalized Additive Models

Varun Babbar*
Department of Computer Science
Duke University
Durham, NC 27705

Christopher Li
Department of Computer Science
Duke University
Durham, NC 27705

Chudi Zhong
Department of Computer Science
University of North Carolina, Chapel Hill
Chapel Hill, NC 27705

Cynthia Rudin
Department of Computer Science
Duke University
Durham, NC 27705

Abstract

The Rashomon set paradigm seeks to uncover the full collection of near-optimal predictive models from a given function class. Access to this set allows users to better understand predictive multiplicity, interact with models, and select those that best satisfy domain-specific constraints. For sparse generalized additive models (GAMs), however, existing approaches that approximate the Rashomon set only apply to restricted subsets of features, overlooking the fact that equally accurate models can rely on many different feature subsets. We address this gap by representing the Rashomon set through a maximally diverse collection of models. We define a Metropolis-Hastings style approach for sampling maximally diverse GAMs from the Rashomon set. Experiments show that our methods can produce substantially more diverse solutions across 3 out of 4 tested diversity metrics, providing users with actionable alternatives that maintain similar predictive performance.

1 Introduction

Interpretability is important to build trust when deploying machine learning models to high-stakes decision-making problems. However, interpretability itself is often not enough. Domain experts may inspect an interpretable model, identify undesirable behavior, and request changes. Because most learning algorithms are designed to return exactly one model, each such request triggers a costly retraining cycle, creating an *interaction bottleneck* [31].

The Rashomon set paradigm offers a fundamentally different approach. Instead of producing a single model, one constructs or approximates the *Rashomon set*, the set of all models whose performance is close to optimal [7, 32]. Studying the Rashomon set provides a richer understanding of the prediction problem than examining only one fitted model [31]: it can support model selection [20, 21], reveal predictive multiplicity [27, 22, 34, 5], assess the stability of variable importance [16, 10, 12], and help practitioners understand which conclusions are robust across many well-performing models [33]. On the algorithmic side, recent work has developed methods to construct Rashomon sets for several model classes, including decision trees [35, 18], linear models [36, 24], and neural networks [23, 11, 13, 14]. These methods allow users to directly explore, search, and select from many near-optimal models within a given model class.

*Corresponding author: vb97@duke.edu

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

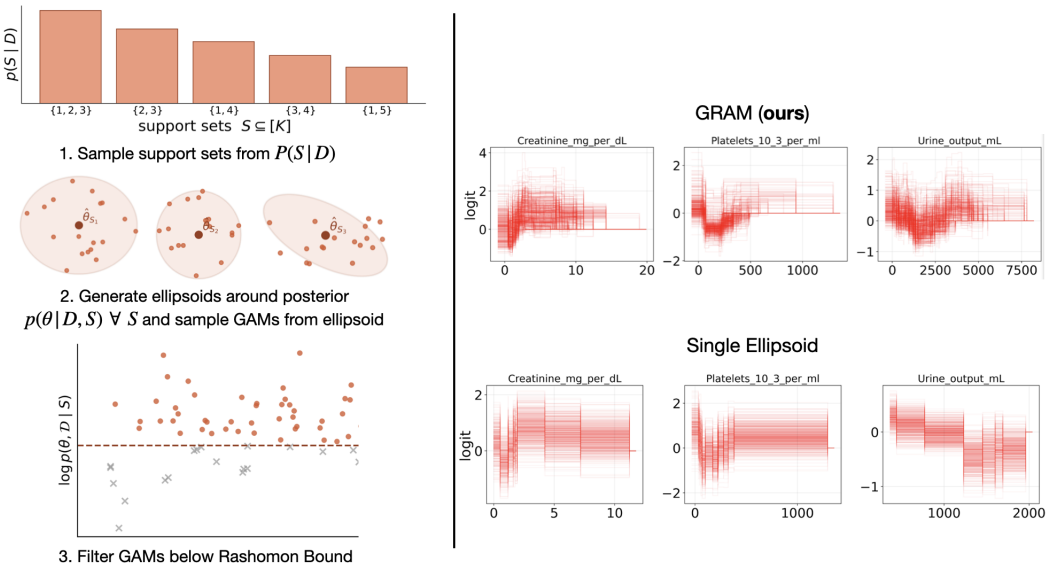


Figure 1: **(Left)** Illustration of our method. We use a Metropolis–Hastings sampler with some novel diversity-related adjustments over support sets $S \subseteq [m]$ to draw from the posterior $p(S \mid D)$, where S is the set of active features. For each sampled support, we find the MAP parameters $\hat{\boldsymbol{\omega}}_S = \arg\max_{\boldsymbol{\omega}} \log p(\boldsymbol{\omega} \mid D, S)$ and, following [36], grow an ellipsoid $R(S) = \{\boldsymbol{\omega} : \log p(\boldsymbol{\omega}, D \mid S) \geq B\}$ around $\hat{\boldsymbol{\omega}}_S$ in parameter space. We then sample models from each ellipsoid and discard any whose log-joint falls below the Rashomon bound B. The resulting Rashomon set is the union $R = \bigcup_S R(S)$. **(Right)** Our method versus [36] on three features from the MGH dataset. The baseline (bottom) fixes a single support set, so all shape functions share the same step locations. Our method (top) explores many support sets, varying both step locations and coefficients, revealing greater structural diversity at matched performance.

While the Rashomon set paradigm can theoretically eliminate the interaction bottleneck, it simultaneously leads to computational challenges, particularly with generalized additive models (GAMs). GAMs are one of the most widely used forms of interpretable predictive models [17]. By decomposing a prediction into a sum of univariate shape functions, a GAM can visualize the contribution of each feature directly via a 2D plot. Modern GAM implementations, such as Explainable Boosting Machines [26], Neural Additive Models [2], and sparse GAMs with ℓ_0 regularization [25], have demonstrated performance competitive with black-box models on tabular datasets. [36] provided the first method for constructing Rashomon sets for sparse GAMs. They use a maximum-volume inscribed ellipsoid to approximate the Rashomon set of sparse GAMs for a fixed support set, namely, a fixed set of selected bin indicators or step functions. This provides an effective building block for exploring coefficient variation within a single support set. They further use a blocking method to analytically derive ellipsoids for sub-support sets contained in the original support. However, this approach has a fundamental limitation: it captures only a single "slice" of the full Rashomon set. The complete Rashomon set of sparse GAMs can span a vast number of different support sets. Because the blocking method can only enumerate sub-support sets of the original support, users may be limited to a narrow region of the Rashomon set.

Exploring a more comprehensive landscape of the Rashomon set for GAMs is very challenging. The number of possible support sets is combinatorial, making exhaustive enumeration infeasible. Moreover, many candidate support sets are either uninformative, such as those that differ only by a minor shift in a bin boundary (e.g., from age < 30 to age < 31), or noncompetitive because no model with that support can achieve near-optimal performance. Enumerating all of them is therefore neither possible nor meaningful. What practitioners need is a collection of diverse support sets, each verified to admit well-performing GAMs. Each such support set represents a structurally distinct way of modeling the data. This requires a method that can (1) search the exponentially large space of support sets efficiently, (2) identify support sets that admit near-optimal models, and (3) promote structural diversity among the selected support sets.

We propose a novel method, named *GAM Rashomon-Sets via Adaptive MCMC* (GRAM), to address these requirements. We introduce a Markov-Chain Monte Carlo style approach that samples over the space of support sets, guided by a diversity objective. This avoids brute-force enumeration and directs the search toward structurally distinct support sets. For each selected support set, we can further construct an ellipsoidal approximation of the corresponding Rashomon set region following [36]. The result covers structurally distinct regions of the full Rashomon set, enabling a broader exploration than was previously possible. Our empirical results show that this approach produces a more diverse and representative Rashomon set of GAMs compared to several baselines across 4 diversity metrics on 10 datasets. Importantly, we also show that GRAM is able to empirically form an effective cover of the Rashomon set.

With the proposed method, we broaden the practical utility of GAM Rashomon sets. Because we cover structurally distinct regions of the Rashomon set, sampling from our representation yields GAMs with greater diversity than sampling from a single ellipsoid, giving practitioners a more representative view of all near-optimal models. Furthermore, by considering many different support sets rather than only sub-support sets of a single starting point, we obtain wider and more informative variable importance ranges: a feature that appears indispensable under one support set may be entirely absent under another, and our method captures this variation. For the full range of experiments, see Appendix C.

2 Background: Sparse GAMs and their Rashomon Sets

We consider a supervised learning dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^p$ is a feature vector and y_i is the corresponding label. A generalized additive model (GAM) has the predictive form

$$g(\mathbb{E}(y)) = \boldsymbol{\omega}_0 + f_1(\mathbf{x}_{\cdot,1}) + f_2(\mathbf{x}_{\cdot,2}) + ... + f_p(\mathbf{x}_{\cdot,p}) \tag{1}$$

where $g(\cdot)$ is a link function, e.g., the identity function for regression or the logistic function for classification. Each shape function f_j is a univariate function of feature j and can be completely visualized as a 2D curve (feature value vs. contribution). This makes the model inherently interpretable: a practitioner can inspect each shape function to see exactly how any feature value maps to its effect on the prediction. For simplicity, we write $\mathbf{x}_j$ instead of $\mathbf{x}_{\cdot,j}$ when the context is clear (the first index is over observations, the second over features).

In practice, continuous features are often discretized into bins [26].

Following [26, 36], the shape function for feature j can be written as

$$f_j(x_j) = \sum_{k=1}^{B_j - 1} \omega_{j,k} \mathbf{1}[b_{j,k} \leq x_j < b_{j,k+1}], \tag{2}$$

where B_j is the number of bin boundaries and $\omega_{j,k}$ is the contribution associated with bin k. Binning does not lose information when bin boundaries are created between every two adjacent feature values in the dataset. After binning, fitting a GAM is equivalent to learning a weight vector $\boldsymbol{\omega} = \{\omega_{j,0}, \omega_{j,1}, ..., \omega_{j,B_j-1}\}_{j=1}^p$. We can use m to denote the total number of bin boundaries across all features, i.e., $m = \sum_{j=1}^p (B_j - 1)$. Each bin boundary corresponds to a potential step (jump) in a shape function.

A sparse GAM for classification is usually trained by minimizing the following loss [25, 36]:

$$\mathcal{L}(\boldsymbol{\omega}, \omega_0, \mathcal{D}) = \mathcal{L}_c(\boldsymbol{\omega}, \omega_0, \mathcal{D}) + \lambda_2 \mathcal{L}_2(\boldsymbol{\omega}, \mathcal{D}) + \lambda_0 \mathcal{L}_0(\boldsymbol{\omega}, \mathcal{D}), \tag{3}$$

where $\mathcal{L}_c$ is the logistic loss, $\mathcal{L}_2$ is the ℓ_2 penalty on the coefficients, and $\mathcal{L}_0$ is the ℓ_0 penalty on the total number of steps in all shape functions. λ_2 and λ_0 control the strength of these penalties.

Following [36], we define the Rashomon set as follows:

Definition 2.1 ($\epsilon-$Rashomon set)**.** For a binned dataset $\mathcal{D}$ with n samples and m binary features, where $\boldsymbol{\omega} \in \mathbb{R}^m$ defines a generalized additive model, the ϵ-Rashomon set is

$$\mathcal{R}(\epsilon, \mathcal{D}) = \{\boldsymbol{\omega} \in \mathbb{R}^m, \omega_0 \in \mathbb{R} : \mathcal{L}(\boldsymbol{\omega}, \omega_0, \mathcal{D}) \leq (1+\epsilon)\mathcal{L}(\boldsymbol{\omega}^*, \omega_0^*, \mathcal{D})\}, \tag{4}$$

where $\boldsymbol{\omega}^*$ is an optimal solution to Eq. (3) which can be obtained by [25], and ϵ controls the tolerance for near-optimality.

Unfortunately, the Rashomon set of GAM is difficult to characterize exactly because GAM coefficients are continuous, and the sparsity penalty induces a non-convex search over support sets. A *support set* $S \subseteq [m]$ specifies which bin boundaries are active as steps in the shape functions; bin boundaries not in S are effectively merged with their neighbors, producing a coarser shape function. The ℓ_0 penalty in Eq. 3 encourages $|S|$ to be small. Prior work, therefore, studies a restricted version of this set. Specifically, [36] first obtains a sparse GAM, fixes its selected support set S with $|S| < m$, and approximates the near-optimal coefficient region within this fixed support by an ellipsoid:

$$\hat{R} := \{\tilde{\boldsymbol{\omega}} \in \mathbb{R}^{|S|+1} : (\tilde{\boldsymbol{\omega}} - \tilde{\boldsymbol{\omega}}_c)^T Q (\tilde{\boldsymbol{\omega}} - \tilde{\boldsymbol{\omega}}_c) \leq 1\} \tag{5}$$

where $\tilde{\boldsymbol{\omega}} = (\omega_0, \boldsymbol{\omega})$, and Q and $\tilde{\boldsymbol{\omega}}_c$ are parameters of the ellipsoid learned by solving a convex optimization problem [36, Eq. 6].

Although [36] can approximate Rashomon sets for smaller support sets after obtaining an ellipsoidal approximation for a larger support set, this procedure only considers subsets of the fixed initial support. As a result, it may leave many well-performing models unexplored, particularly those that rely on different features or bin indicators outside the initial S. This motivates our central question: *How can we efficiently find a diverse collection of well-performing sparse GAMs over different support sets in high-dimensional feature spaces?*

Exhaustively training GAMs over all possible support sets $S \subseteq [m]$ is computationally infeasible due to the combinatorial number of candidates. We therefore need a method that can efficiently identify promising support sets likely to produce near-optimal sparse GAMs, and use the resulting collection of models to better approximate the broader GAM Rashomon set.

3 Methodology

3.1 A Bayesian View of GAM Rashomon Sets

We approach the problem of finding diverse, well-performing support sets through a Bayesian lens, which asks: *given the data, which support sets are most likely to yield GAMs in the Rashomon set?* Let $S \subseteq [m]$ be a support set. We model the posterior over support sets as:

$$p(S|\mathcal{D}) \propto p(\mathcal{D}|S)p(S) \tag{6}$$

The marginal likelihood $p(\mathcal{D}|S)$ integrates over all possible weight vectors associated with the active bin boundaries in S:

$$p(\mathcal{D}|S) = \int p(\mathcal{D}|S, \boldsymbol{\omega}) p(\boldsymbol{\omega}|S) d\boldsymbol{\omega}, \tag{7}$$

where $\boldsymbol{\omega}$ are the parameters of a GAM. And therefore, $p(S|\mathcal{D}) \propto p(S) \int p(\mathcal{D}|S, \boldsymbol{\omega}) p(\boldsymbol{\omega}|S) d\boldsymbol{\omega}$.

To connect this Bayesian formulation to the GAM training objective in Eq. 3, we specify the likelihood and prior as follows: For a GAM trained using logistic regression, the likelihood function $p(\mathcal{D}|S, \boldsymbol{\omega})$ is given by $\exp(-\mathcal{L}_c(\boldsymbol{\omega}, \mathcal{D}))$, where $\mathcal{L}_c$ is the logistic loss. Given the vector π of bin-proportions (where $\pi_{i,j}$ represents the proportion of samples in bin j of feature i), we impose a Gaussian prior over the parameters:

$$p(\boldsymbol{\omega}|S) = \mathcal{N}\left(0, \frac{1}{2\lambda_2} \text{diag}(\pi)^{-1}\right) \tag{8}$$

Under this prior, we have $-\log p(\boldsymbol{\omega}|S) = \lambda_2 \sum_{j,k} \pi_{j,k}\, \omega_{j,k}^2 + \text{const} = \lambda_2\, \mathcal{L}_2(\boldsymbol{\omega}) + \text{const}$, which recovers the weighted $\mathcal{L}_2$ penalty in Eq 3. Combining the log-likelihood and the log-prior, we obtain:

$$-\log p(\mathcal{D}|S, \boldsymbol{\omega}) p(\boldsymbol{\omega}|S) = \mathcal{L}_c(\boldsymbol{\omega}, \mathcal{D}) + \lambda_2 \mathcal{L}_2(\boldsymbol{\omega}) + \text{const} \tag{9}$$

which recovers the loss function in Eq. 3 without the ℓ_0 term. This means that the Rashomon set definition in Eq. 4 can be equivalently expressed as a lower bound on the log-posterior. Specifically, for a fixed support set S, we define the Rashomon set for a single support set as the set of all parameters whose unnormalized log-posterior exceeds a threshold B:

$$R(S) := \{\boldsymbol{\omega} \in \mathbb{R}^{|S|} : \log p(\mathcal{D}|S, \boldsymbol{\omega}) + \log p(\boldsymbol{\omega}|S) \geq B\}. \tag{10}$$

The full Rashomon set is the union over all support sets, i.e., $R := \bigcup_{S \subseteq [m]} R(S)$. Since exhaustive enumeration is infeasible, we need an efficient method to identify promising support sets.

3.2 Sampling Diverse Support Sets

GRAM uses a Metropolis-Hastings (MH) style algorithm to sample support sets from $p(S|\mathcal{D})$, augmented with a diversity term that biases the chain toward structurally distinct support sets. At each step, we propose a new support set S', evaluate its posterior score, and accept or reject based on a ratio that balances posterior quality with diversity relative to the already-collected Rashomon set. For each accepted support set, we grow an ellipsoidal approximation of $R(S)$ following [36] and sample GAMs from it. The full procedure is given in Algorithm 1.

Evaluating the posterior score. Computing the posterior $p(S|\mathcal{D})$ requires evaluating the likelihood $p(\mathcal{D}|S)$ in Eq. 7, which involves an intractable integral. We use the Laplace approximation from [30] to approximate $p(\mathcal{D}|S)$. Letting $\hat{\boldsymbol{\omega}}_S$ denote the MAP estimate of the weights for support S and

$$A_S \;=\; -\nabla^2 \log p(\boldsymbol{\omega}|\mathcal{D}, S)\Big|_{\boldsymbol{\omega}=\hat{\boldsymbol{\omega}}_S} \;=\; X_S^\top \operatorname{diag}(\pi_i(1-\pi_i))\, X_S + \tfrac{1}{\sigma^2}\mathbf{I} \tag{11}$$

denote the Hessian of the negative log posterior at the MAP, we obtain

$$\log p(\mathcal{D}|S) \;\approx\; \log p(\mathcal{D}|\hat{\boldsymbol{\omega}}_S, S) \;+\; \log p(\hat{\boldsymbol{\omega}}_S|S) \;+\; \tfrac{|S|}{2}\log(2\pi) \;-\; \tfrac{1}{2}\log\det(A_S). \tag{12}$$

The first two terms represent the unnormalized log-posterior at its mode, while the remaining terms account for the local Gaussian volume around the MAP. Note that the prior variance σ^2 is equivalent to $\frac{1}{\lambda_2}$ from Equation 8.

For the support set prior $p(S)$, to encourage diversity, we assign each bin boundary an equal probability of inclusion p_{feat}, hence $p(S) = p_{\text{feat}}^{|S|}(1-p_{\text{feat}})^{m-|S|}$. This leads to the score function:

$$J(S) = \log p(S|\mathcal{D}) = \log p(\mathcal{D}|S) + |S|\log p_{\text{feat}} + (m-|S|)\log(1-p_{\text{feat}}) + \text{constant} \tag{13}$$

Acceptance ratio. At each MH step, we propose S' from a proposal distribution $q(S'|S^{(t-1)})$ and accept with the log-ratio:

$$\begin{aligned}\log\alpha =& J(S') - J(S^{(t-1)}) + \log q(S^{(t-1)}|S') - \log q(S'|S^{(t-1)}) \\ &+ \lambda_r\big(\log\phi(S', R) - \log\phi(S^{(t-1)}, R)\big).\end{aligned} \tag{14}$$

The first two terms form the standard Metropolis–Hastings ratio targeting $p(S|\mathcal{D})$. The last term, controlled by $\lambda_r \geq 0$, favors support sets that are diverse with respect to the current support set. The full procedure is given in Algorithm 1.

Line 6 in Algorithm 1 relies on fast computation of the score function (shown in Algorithm 2) to evaluate the utility of a proposed support set. Given a candidate support set S, the procedure first computes the MAP coefficients $\hat{\boldsymbol{\omega}}_S$ under the (smoothness-augmented) Gaussian prior via regularized logistic regression, then forms the Hessian A_S at the MAP, applies the Laplace approximation to estimate the marginal likelihood $\log p(D \mid S)$, and finally adds the Bernoulli sparsity prior over feature inclusion to produce the score $J(S) = \log p(S \mid D) + \text{const}$. This score is the quantity compared in the Metropolis-Hastings acceptance ratio (Eq. 14).

Proposal Distribution An important aspect of our algorithm is the proposal distribution $q(S'|S^{(t-1)})$. In this paper, we use an adaptive proposal distribution that is allowed to make one of 3 actions $a_t \in \{\text{swap, add, remove}\}$. We initialize these actions to be equiprobable. For the add/remove action, we sample up to $k_{max} = 5$ features to be added/removed. We keep track of the exponential moving average of the action probabilities based on past acceptance history – if a given action often results in acceptance, its average probability increases over time. Let the vector $\mathbf{v}_t$ represent the unnormalized action potential at time t. We perform the following update at time t, given proposed support set S' sampled from $q(S'|S^{(t-1)})$:

$$\mathbf{v}_t[a_t] = (1-\alpha)\mathbf{v}_{t-1}[a_t] + \alpha(J(S') - J(S^{(t-1)}))) \tag{15}$$

$$\mathbf{p}_t = \text{softmax}\Big(\frac{\mathbf{v}_t}{T}\Big). \tag{16}$$

An example proposal probability for adding 2 features to the support set would look like:

$$\log q(S'|S^{(t-1)}) = \log \mathbf{p}_{(t-1)}[a_t] - \log k_{max} - \log\binom{|m|}{2}. \tag{17}$$

For completeness, we perform experiments with a few possible proposal distributions in the appendix.

Algorithm 1 GRAM: GAM Rashomon sets via Adaptive MCMC

Input: Data $D = \{(x_i, y_i)\}$, prior variance σ^2, sparsity p (prior probability of including a feature in support set), steps T_{total}, Rashomon bound B, diversity metric ϕ, diversity weight λ_r.

Initialize support set $S^{(0)}$.
$J^{(0)} \leftarrow \text{EVALUATESCORE}(S^{(0)})$
$R \leftarrow \emptyset$ ▷ Initialize the Rashomon set
for $t = 1$ **to** T_{total} **do**
  Propose new support set $S' \sim q(S' \mid S^{(t-1)})$. ▷ Require a proposal function
  $J' \leftarrow \text{EVALUATESCORE}(S')$ ▷ Score for proposed support set- Algorithm 2
  Compute the weights of the optimal GAM $\hat{\boldsymbol{\omega}}_{S'}$ and the corresponding posterior $\log p(D|\hat{\boldsymbol{\omega}}_{S'}, S') + \log p(\hat{\boldsymbol{\omega}}_{S'}|S')$ based on Equation 9
  Compute $\log \alpha$ based on Eq 14 ▷ Acceptance probability α is high if proposal is well-performing and diverse w.r.t. R
  Sample $u \sim \text{Uniform}(0, 1)$.
  if $\log u < \min(0, \log \alpha)$ **then**
    $S^{(t)} \leftarrow S'$, $J^{(t)} \leftarrow J'$ ▷ Accept
    if $\log p(D|\hat{\boldsymbol{\omega}}_{S'}, S') + \log p(\hat{\boldsymbol{\omega}}_{S'}|S') \geq B$ **then** ▷ Check if the best model for the proposed support set is within the Rashomon bound
      Add the model $\hat{\boldsymbol{\omega}}_{S'}$ to R.
    end if
  else
    $S^{(t)} \leftarrow S^{(t-1)}$, $J^{(t)} \leftarrow J^{(t-1)}$ ▷ Reject
  end if
end for
for $S, \log p(\hat{\boldsymbol{\omega}}_S \mid D) \in R$ **do**
  $\varepsilon = \log p(\hat{\boldsymbol{\omega}}_S \mid D) - B$ ▷ Residual budget for support set
  Grow an ellipsoid ball E_S around $\hat{\boldsymbol{\omega}}_S$ with tolerance ε using the method of [36].
  Sample K models from E_S and add to R if their weights satisfy the Rashomon bound B.
end for

Output: Rashomon set of GAMs R.

Diversity Penalty The diversity term $\lambda_r \big(\log \phi(S', R) - \log \phi(S^{(t-1)}, R)\big)$ in the acceptance ratio (see Eq 25) biases the chain toward proposals that are dissimilar from already-accepted members of the Rashomon set. In this work, we experiment with 4 diversity metrics, described below.

- **Support Set (Jaccard) Diversity:** Given k support sets $\Gamma = \{S_1, ..S_k\}$ over which the overall Rashomon set is defined, we compute the Jaccard diversity for an incoming support set S as:

$$\phi_{JD}(S, R) = 1 - \frac{1}{|R|} \sum_{\boldsymbol{\omega} \in R} \frac{|\text{support}(\boldsymbol{\omega}) \cap S|}{|\text{support}(\boldsymbol{\omega}) \cup S|}. \tag{18}$$

- **Monotonicity Entropy:** Given the overall Rashomon set R collected so far across all support sets, for each feature $j \in \{1, ..p\}$, let $p_j^{\uparrow}(R), p_j^{\downarrow}(R)$, and $p_j^{-}(R)$ be the proportion of GAMs in R with monotonically increasing shape functions for feature j (resp. decreasing and neither). Given p features, we define the monotonicity entropy as:

$$\phi_{ME}(S, R) = -\frac{1}{p} \sum_{j=1}^{p} \sum_{u \in \{\uparrow, \downarrow, -\}} p_j^u(R \cup \boldsymbol{\omega}) \log p_j^u(R \cup \boldsymbol{\omega}). \tag{19}$$

  GAMs that disagree on whether a feature's effect is increasing, decreasing, or non-monotonic represent fundamentally different inferences about the data.
- **Prediction Hamming Distance**: The average pairwise Hamming distance between the prediction vectors of GAMs in the set.
- **L1**: The average pairwise L_1 distance between GAMs in the Rashomon set. To remove arbitrary scale fluctuations, this is computed by first normalizing the GAMs such that the weight vectors have unit norm.

The hyperparameter $\lambda_r \geq 0$ controls the diversity–fidelity trade-off. Setting $\lambda_r = 0$ recovers the standard Metropolis–Hastings sampler targeting $p(S \mid D)$. For $\lambda_r > 0$, the effective target at a

given instance becomes $p(S \mid D)\,\phi(S,R)^{\lambda_r}$, which is no longer detail-balanced with respect to the posterior alone; we view this as a deliberate design choice rather than a defect, since our goal is to enumerate *diverse* near-optimal models, not necessarily to reproduce posterior mass faithfully. In particular, the Rashomon-bound check on Line 12 of Algorithm 1 is performed against the unmodified log-posterior, so the diversity penalty influences *which* good support sets are visited but never permits the inclusion of a poor support set in R.

Smoothness guidance. An important aspect of interpretable GAMs is that the shape function should not contain many jumps, i.e. there should not be a large discontinuity between age $= 30$ and age $= 31$. To encourage such smoothness, we propose an additional term to the prior $p(\hat{\boldsymbol{\omega}}|S)$ in Equation 7 that penalizes such jumps, i.e.

$$\log p(\hat{\boldsymbol{\omega}}|S) = -\frac{1}{2\sigma^2}\Big(\sum_{j=1}^{p}\sum_{k=1}^{B_j-1}|\omega_{j,k}-\omega_{j,k+1}|^2 + \|\boldsymbol{\omega}\|_2^2\Big) + \text{constant} \tag{20}$$

This term is only used in the acceptance probability computation to guide the chain towards support sets that result in smooth, well performing GAMs; it is not part of the Rashomon objective.

From Support Sets to the Rashomon set. Given a collection of sampled support sets $\Lambda = \{S_1, \ldots, S_n\} \sim p(S|\mathcal{D})$, we approximate the full Rashomon set as

$$\hat{R} \;\approx\; \bigcup_{S\in\Lambda} R(S). \tag{21}$$

For each accepted support set S, we grow an ellipsoidal approximation of $\mathcal{R}(S)$ following [36], sample K GAMs from the ellipsoid, and discard any whose log-posterior falls below the Rashomon bound B. The full procedure is given in Algorithm 1.

Algorithm 2 EvaluateScore(S)

1: **1. Find MAP parameters** ($\hat{\boldsymbol{\omega}}_S$):

$$\hat{\boldsymbol{\omega}}_S = \arg\max_{\boldsymbol{\omega}} \underbrace{\sum_{i=1}^{N}\log p(y_i \mid x_{i,S},\boldsymbol{\omega})}_{\text{log-likelihood}} - \underbrace{\tfrac{1}{2\sigma^2}\|\boldsymbol{\omega}\|^2}_{\text{log-prior}}.$$

2: **2. Compute Hessian** (A_S) at $\hat{\boldsymbol{\omega}}_S$:

$$A_S = \underbrace{X_S^\top \operatorname{diag}(\pi_i(1-\pi_i))\,X_S}_{\text{Fisher info}} + \underbrace{\tfrac{1}{\sigma^2}I}_{\text{prior precision}}, \qquad \pi_i = \sigma(\mathbf{x}_i^\top\hat{\boldsymbol{\omega}}_S).$$

3: **3. Laplace approximation** ($\log p(D \mid S)$):

$$L(S) \approx \log p(D \mid \hat{\boldsymbol{\omega}}_S) + \log p(\hat{\boldsymbol{\omega}}_S) - \tfrac{1}{2}\log\det(A_S) + \tfrac{|S|}{2}\log(2\pi).$$

4: **4. Compute smoothness penalty** at $\hat{\boldsymbol{\omega}}_S$:

$$\Omega(\hat{\boldsymbol{\omega}}_S) = \sum_{j=1}^{p}\sum_{k=1}^{B_j-1}(\hat{\omega}_{j,k}-\hat{\omega}_{j,k+1})^2.$$

5: **5. Add sparsity prior and smoothness guide:**
6: **return** $J(S) = L(S) + |S|\log p + (m-|S|)\log(1-p) - \frac{1}{2\sigma^2}\,\Omega(\hat{\boldsymbol{\omega}}_S)$.

3.3 Optimizations

Fine tuning vs full resolving: Each step of Algorithm 1 requires scoring a proposed support S', which in turn requires solving a regularized logistic regression to obtain $\hat{\boldsymbol{\omega}}_{S'}$. When S' differs from $S^{(t-1)}$ only by swapping feature i for a strongly correlated feature j, the previous solution $\hat{\boldsymbol{\omega}}_{S^{(t-1)}}$

is already nearly optimal, and a single coordinate-descent step on the new coefficient suffices. We show in Theorem 3.1 that the suboptimality of the coordinate-refit relative to the full refit scales as $(1-\rho^2)$, where ρ is the correlation between x_i and x_j. We therefore fine-tune whenever $|\rho| \geq \tau$ and trigger a full resolve every 50 steps to prevent drift. More details, including fine tuning when adding / removing a feature, are in Appendix B.

Theorem 3.1 (Swap gap bound via correlation for logistic regression)**.** *Let $y_l \in \{\pm 1\}$, and suppose all feature columns are centered and standardized ($\|\mathbf{x}_k\|_2^2 = n$). Consider swapping binned feature i for feature j in support S, where $\frac{1}{n}\mathbf{x}_i^\top \mathbf{x}_j = \rho$. Let $\boldsymbol{\omega}_{\mathrm{cd}}$ denote the solution obtained by fixing all existing coefficients and optimizing only the new coefficient for feature j (coordinate refit), and let $\boldsymbol{\omega}_{\mathrm{full}}$ denote the fully re-optimized solution. The suboptimality of the coordinate refit on the ridge-regularized logistic loss $\mathcal{L}$ satisfies*

$$\mathcal{L}(\boldsymbol{\omega}_{\mathrm{cd}}) - \mathcal{L}(\boldsymbol{\omega}_{\mathrm{full}}) \;\leq\; \tfrac{\sigma^2 \kappa_S}{32}\Big(|\hat{\omega}_i| + \sigma^2\big(1 + \tfrac{|\hat{\omega}_i|}{4}\big)\Big)^2 (1-\rho^2),$$

where σ^2 is the ridge parameter, $\kappa_S = \lambda_{\max}\big(\frac{1}{n}\mathbf{X}_S^\top \mathbf{X}_S\big)$, and $\hat{\omega}_i$ is the coefficient of feature i in the original solution. Thus, the coordinate-refit gap vanishes as $|\rho| \to 1$.

Initialization. Given a large set of potential binary features, we initialize the algorithm with a support chosen using ℓ_2 feature selection. We run ℓ_2 logistic regression on the entire feature set with a large regularization value of $\frac{1}{p_{\text{feat}}}$ for fast convergence. During sampling, the algorithm automatically adapts towards choosing support sets with the appropriate degree of sparsity (i.e. sets around the high density posterior region).

Efficient diversity computation. We make three optimizations. Firstly, the monotonicity entropy requires computing the proportion of models which are monotonic / non-monotonic. To avoid redundant computation, we keep a running tab of these statistics and simply update them when a new model is added. Secondly, if the Rashomon set becomes large (e.g., greater than 200 models), we compute the diversity term associated with a random sample of the Rashomon set. Lastly, to ensure that diversity term scores support sets according to all well performing models that can be formed from that support set, we sample a few GAMs associated with that support set and compute the diversity term with respect to these GAMs.

4 Experiments

We experiment with our method on 10 datasets. To produce the large initial support pool m, we first binarize the data using a combination of the threshold guessing in [28] alongside random threshold selection. For every continuous / categorical feature in the dataset, we aim to ensure that at least 50% of the empirical thresholds observed in the dataset are represented in the support pool – if threshold guessing produces binary features fewer than this, we sample randomly from the remaining binary thresholds to make up for this gap. By default, we include all originally binary features in the support pool. This binarization makes the support set pool very large, often in thousands of binary features.

4.1 GRAM provides more diverse GAM Rashomon sets

We compare our method against five baselines. **Swapping** is an extension of the FasterRisk algorithm [24]. It uses FasterRisk's swap-based beam search to generate a diverse set of sparse linear (risk-score) models. This is performed up to a depth of 2, effectively creating a candidate support swap tree. All models found in this swap tree are filtered to see if they satisfy the Rashomon bound (see Appendix for a more detailed description). **Ellipsoid** applies L_1 feature selection to match GRAM's average support size, then uses [36] to fit the optimal L_2-regularized GAM, build an ellipsoid around it, and sample the same number of models as GRAM. **Bootstrap** resamples the training set n times, fitting an optimal L_2-regularized GAM with L_1-selected support size 100 on each. The final two baselines sample random support sets of size equal to GRAM's average, fit the optimal L_2 GAM on each, and draw 20 models per support via the ellipsoid method, either **uniformly** or with **rejection** at minimum distance $d = 0.01\times$ the major axis length from the center. For these baselines, the number of support sets matches the average support set size of GRAM. For fairness, we filter all models output by all baselines by the Rashomon bound set by GRAM's optimal solution.

From Table 1, we see that GRAM outputs GAMs that are very often the most structurally diverse across 3 out of the 4 metrics tested, with L_1 coming in second place. Of particular interest is the monotonicity entropy metric – a larger diversity here implies that our method often provides GAMs that have a mix of monotonically increasing / decreasing shape functions.

Table 1: A comparison of our approach vs baseline methods. For every diversity metric in $\{L_1$, Jaccard, Monotonicity Entropy, Prediction Hamming$\}$, we count the number of (dataset, ϵ) pairs where GRAM produced more diverse solutions according to that metric. The coverage indicates the percentage of (dataset, ϵ, diversity metric) runs which produced > 1 model in the Rashomon set. We ran the methods on 10 datasets for $\epsilon \in \{0.005, 0.01, 0.015, 0.025\}$. Values are mean $\pm$ standard error over 5 trials. More complete results can be seen in the appendix.

	Diversity Metrics				
Method	L_1	**Jaccard**	**Pred. Hamming**	**Mono. Entropy**	**Coverage (%)**
GRAM (**ours**)	9.4 ± 2.3	$\mathbf{29.4 \pm 3.6}$	$\mathbf{16.4 \pm 2.7}$	$\mathbf{31.6 \pm 1.9}$	$\mathbf{100.0 \pm 0.0}$
Swapping	$\mathbf{19.0 \pm 1.0}$	8.8 ± 2.5	4.0 ± 1.4	3.6 ± 2.2	$\mathbf{100.0 \pm 0.0}$
Single Ellipsoid	11.2 ± 1.9	0.0 ± 0.0	10.8 ± 3.4	4.0 ± 2.9	$\mathbf{100.0 \pm 0.0}$
Bootstrapped GAMs	0.4 ± 0.5	0.0 ± 0.0	6.8 ± 1.9	0.4 ± 0.5	79.9 ± 0.8
Random Ellipsoids	0.0 ± 0.0	1.2 ± 1.6	0.8 ± 0.8	0.0 ± 0.0	5.6 ± 2.4
Random Ellipsoids (rej.)	0.0 ± 0.0	0.6 ± 0.5	1.2 ± 1.6	0.4 ± 0.9	5.8 ± 2.3

4.2 Performing single coordinate fine-tuning more often improves GRAM's runtime

Our next experiment involves assessing the fine-tuning optimization trick of our method. In particular, we vary the minimum correlation threshold τ required to activate fine-tuning from 0 to 0.5, running GRAM for each value. From Figure 2, we see that at lower values of the threshold (corresponding to a larger rate of fine tuning), we obtain much lower runtimes for the algorithm, often making it $2 - 3\times$ faster than the no-finetune baseline. In the appendix, we show that this runtime improvement is subject to some tradeoffs. In particular, fine tuning more often can reduce the size of the output Rashomon set as well as the efficiency of the MCMC sampler.

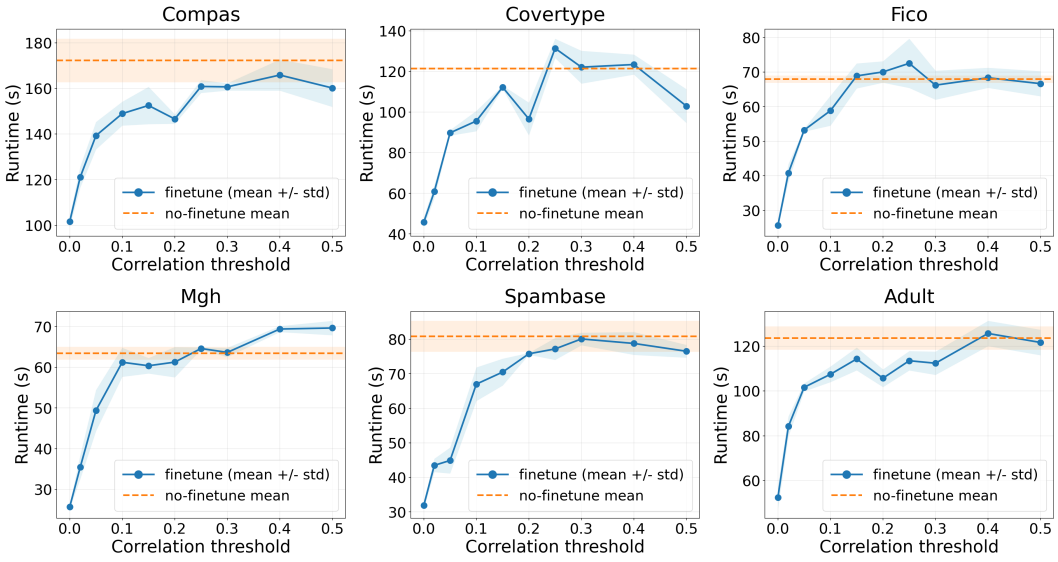


Figure 2: An illustration of the degree of single-coordinate finetuning vs runtime of GRAM. Lower correlation thresholds imply more frequent finetuning as new support sets are chosen. In this regime, GRAM can be $2 - 3\times$ faster than if we perform a logistic regression re-solve on the new support set. For these experiments, we set $\epsilon = 0.02$, number of iterations $= 5000$, $\sigma^2 = 0.01$, $p_{feat} = 0.01$.

4.3 The set of models form an empirical cover for the true, intractable Rashomon set

In this experiment, we run GRAM until 10^5 models are collected (25 models sampled per accepted support set), and cluster the resulting models (i.e. their weight vectors) via K-means for $K \in \{5, 20, 50, 100, 500\}$. For each K, we report the fraction of clusters containing at least one of the first 5000 sampled models. As shown in Table 2, the first 5% of sampled models already cover 75–90$\%$ of the clusters even at $K = 100$, indicating rapid coverage of the Rashomon set.

Table 2: $\%$ of k-means clusters of a large Rashomon set covered by the first 5000 models sampled by GRAM. Entries are the mean $\pm$ sample standard error over 5 K-means initializations. $\epsilon = 0.05$, $\sigma^2 = 0.01$, $p_{\text{feat}} = 0.001$.

Dataset	**K=5**	**K=20**	**K=50**	**K=100**	**K=500**
Compas	100.00 ± 0.00	100.00 ± 0.00	87.20 ± 1.74	74.60 ± 1.21	44.12 ± 0.60
FICO	100.00 ± 0.00	100.00 ± 0.00	91.20 ± 1.36	80.20 ± 0.66	41.28 ± 0.36
NIJ Recidivism	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	92.76 ± 0.29
Covertype	100.00 ± 0.00	100.00 ± 0.00	97.20 ± 1.02	92.00 ± 0.71	57.36 ± 0.95
MGH	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	99.00 ± 0.45	74.28 ± 0.97
Student	100.00 ± 0.00	100.00 ± 0.00	97.20 ± 1.50	91.40 ± 0.98	83.40 ± 0.54
Wine	100.00 ± 0.00	100.00 ± 0.00	99.60 ± 0.40	99.20 ± 0.20	74.28 ± 0.81
Heart	100.00 ± 0.00	100.00 ± 0.00	98.80 ± 0.80	97.60 ± 0.24	91.96 ± 0.57
Adult	100.00 ± 0.00	100.00 ± 0.00	99.60 ± 0.40	98.60 ± 0.51	74.76 ± 1.31
Spambase	100.00 ± 0.00	100.00 ± 0.00	98.80 ± 0.80	94.40 ± 0.93	89.72 ± 0.44

5 Conclusion

In this paper, we introduced GRAM, a Metropolis–Hastings-style algorithm for sampling diverse Rashomon sets of sparse generalized additive models. By recasting the search for near-optimal GAMs as posterior inference over support sets, GRAM moves beyond the fixed-support ellipsoidal approximation of [36] and instead aggregates ellipsoids across many structurally distinct supports. Using several optimization tricks such as single coordinate fine tuning for correlated support sets, a diversity aware acceptance ratio, and adaptive proposal distributions, GRAM performs a meaningful and computationally tractable exploration of the search space, even with 1000s of binarized features. Empirically, GRAM produces Rashomon sets that are substantially more diverse than single-ellipsoid, swap-based, bootstrapped, and random-support baselines across 3 out of the 4 diversity metrics and 10 datasets, while a small fraction of sampled models is able to form an effective cover for a much larger Rashomon set if the algorithm is run for a long time.

We hope that future work will study the impact of this newfound diversity on downstream model selection and human-AI collaboration in real world, high stakes decision making settings.

Broader Impacts This work aims to enhance model selection for interpretable predictive models by providing a larger, more diverse set of model choices. This may benefit a wide range of high-stakes tasks. We are unaware of any negative potential social impacts of these ideas.

Acknowledgements This work relates to Department of Navy award N000142612253 issued by the Office of Naval Research. We acknowledge funding from the National Institutes of Health (NIH) under grant R01-DA054994.

References

[1] Stefan Aeberhard and M. Forina. Wine. UCI Machine Learning Repository, 1992. DOI: https://doi.org/10.24432/C5PC7J.

[2] Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. *Advances in neural information processing systems*, 34:4699–4711, 2021.

[3] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. ProPublica, 2016. `https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing`.

[4] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[5] Emily Black, Manish Raghavan, and Solon Barocas. Model multiplicity: Opportunities, concerns, and solutions. In *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*, pages 850–863, 2022.

[6] Jock Blackard. Covertype. UCI Machine Learning Repository, 1998. DOI: https://doi.org/10.24432/C50K5N.

[7] Leo Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, 16(3):199–231, 2001.

[8] Paulo Cortez and Alice Maria Gonçalves Silva. Using data mining to predict secondary school student performance. In *Proc. 5th Future Business Technology Conference (FUBUTEC)*, pages 5–12, 2008.

[9] Robert Detrano, Andras Janosi, Walter Steinbrunn, Matthias Pfisterer, Johann-Jakob Schmid, Sarbjit Sandhu, Kern H. Guppy, Stella Lee, and Victor Froelicher. International application of a new probability algorithm for the diagnosis of coronary artery disease. *The American Journal of Cardiology*, 64(5):304–310, 1989.

[10] Jiayun Dong and Cynthia Rudin. Exploring the cloud of variable importance for the set of all good models. *Nature Machine Intelligence*, 2(12):810–824, 2020.

[11] Jon Donnelly, Zhicheng Guo, Alina Jade Barnett, Hayden McTavish, Chaofan Chen, and Cynthia Rudin. Rashomon sets for prototypical-part networks: Editing interpretable models in real-time. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4528–4538, 2025.

[12] Jon Donnelly, Srikar Katta, Cynthia Rudin, and Edward P. Browne. The rashomon importance distribution: Getting rid of unstable, single model-based variable importance. In *Proceedings of Neural Information Processing Systems (NeurIPS)*, 2023.

[13] Gilles Eerlings, Brent Zoomers, Jori Liesenborgs, Gustavo Rovelo Ruiz, and Kris Luyten. Diverse: Disagreement-inducing vector evolution for rashomon set exploration. *arXiv preprint arXiv:2601.20627*, 2026.

[14] Shihan Feng, Cheng Zhang, Michael Xi, Ethan Hsu, Lesia Semenova, and Chudi Zhong. Parameter-efficient construction of the rashomon slice for concept bottleneck models. *Transactions on Machine Learning Research*, 2026.

[15] FICO. Explainable machine learning challenge. `https://community.fico.com/s/explainable-machine-learning-challenge`, 2018.

[16] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019.

[17] Trevor Hastie and Robert Tibshirani. Generalized additive models. *Statistical science*, 1(3):297–310, 1986.

[18] Zakk Heile, Hayden Mctavish, Varun Babbar, Margo Seltzer, and Cynthia Rudin. From rashomon theory to PRAXIS: Efficient decision tree rashomon sets. In *Proceedings of the 43rd International Conference on Machine Learning*, pages 42550–42662. PMLR, 2026.

[19] Mark Hopkins, Erik Reeber, George Forman, and Jaap Suermondt. Spambase data set. UCI Machine Learning Repository, 1999.

[20] Ethan Hsu, Tony Cao, Lesia Semenova, and Chudi Zhong. The rashomon set has it all: Analyzing trustworthiness of trees under multiplicity. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025.

[21] Ethan Hsu, Harry Chen, Chudi Zhong, and Lesia Semenova. The double-edged nature of the rashomon set for trustworthy machine learning. In *Proceedings of the 43rd International Conference on Machine Learning*, pages 44452–44490. PMLR, 2026.

[22] Hsiang Hsu and Flavio Calmon. Rashomon capacity: A metric for predictive multiplicity in classification. *Advances in Neural Information Processing Systems*, 35:28988–29000, 2022.

[23] Hsiang Hsu, Guihong Li, Shaohan Hu, and Chun-Fu Chen. Dropout-based rashomon set exploration for efficient predictive multiplicity estimation. In *The Twelfth International Conference on Learning Representations, 2024*.

[24] Jiachang Liu, Chudi Zhong, Boxuan Li, Margo Seltzer, and Cynthia Rudin. Fasterrisk: fast and accurate interpretable risk scores. *Advances in Neural Information Processing Systems*, 35:17760–17773, 2022.

[25] Jiachang Liu, Chudi Zhong, Margo Seltzer, and Cynthia Rudin. Fast sparse classification for generalized linear and additive models. In *AISTATS*, pages 9304–9333, 2022.

[26] Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 623–631, 2013.

[27] Charles Marx, Flavio Calmon, and Berk Ustun. Predictive multiplicity in classification. In *International conference on machine learning*, pages 6765–6774. PMLR, 2020.

[28] Hayden McTavish, Chudi Zhong, Reto Achermann, Ilias Karimalis, Jacques Chen, Cynthia Rudin, and Margo Seltzer. Fast sparse decision tree optimization via reference ensembles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9604–9613, 2022.

[29] NIJ. NIJ's Recidivism Challenge Full Dataset | Office of Justice Programs, 2021.

[30] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

[31] Cynthia Rudin, Chudi Zhong, Lesia Semenova, Margo Seltzer, Ronald Parr, Jiachang Liu, Srikar Katta, Jon Donnelly, Harry Chen, and Zachery Boner. Amazing things come from having many good models. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

[32] Lesia Semenova, Cynthia Rudin, and Ronald Parr. On the existence of simpler machine learning models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1827–1858, 2022.

[33] Bohdan Turbal, Iryna Voitsitska, and Lesia Semenova. Ellice: Efficient and provably robust algorithmic recourse via the rashomon sets. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[34] Jamelle Watson-Daniels, David C Parkes, and Berk Ustun. Predictive multiplicity in probabilistic classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10306–10314, 2023.

[35] Rui Xin, Chudi Zhong, Zhi Chen, Takuya Takagi, Margo Seltzer, and Cynthia Rudin. Exploring the whole rashomon set of sparse decision trees. In *Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS 2022)*. NeurIPS, 2022. Oral presentation.

[36] Chudi Zhong, Zhi Chen, Jiachang Liu, Margo Seltzer, and Cynthia Rudin. Exploring and interacting with the set of good sparse generalized additive models. In *Advances in Neural Information Processing Systems*, 2023.

A Proofs

Theorem 3.1. *Let $y_l \in \{\pm 1\}$, and suppose all feature columns are centered and standardized ($\|\mathbf{x}_k\|_2^2 = n$). Consider swapping binned feature i for feature j in support S, where $\frac{1}{n}\mathbf{x}_i^\top \mathbf{x}_j = \rho$. Let $\boldsymbol{\omega}_{\mathrm{cd}}$ denote the solution obtained by fixing all existing coefficients and optimizing only the new coefficient for feature j (coordinate refit), and let $\boldsymbol{\omega}_{\mathrm{full}}$ denote the fully re-optimized solution. The suboptimality of the coordinate refit on the ridge-regularized logistic loss $\mathcal{L}$ satisfies*

$$\mathcal{L}(\boldsymbol{\omega}_{\mathrm{cd}}) - \mathcal{L}(\boldsymbol{\omega}_{\mathrm{full}}) \;\le\; \tfrac{\sigma^2 \kappa_S}{32}\Big(|\hat{\omega}_i| + \sigma^2\big(1 + \tfrac{|\hat{\omega}_i|}{4}\big)\Big)^2 (1-\rho^2),$$

where σ^2 is the ridge parameter, $\kappa_S = \lambda_{\max}\big(\frac{1}{n}\mathbf{X}_S^\top \mathbf{X}_S\big)$, and $\hat{\omega}_i$ is the coefficient of feature i in the original solution. Thus, the coordinate-refit gap vanishes as $|\rho| \to 1$.

Proof. The L_2 regularized logistic loss objective $\mathcal{L}$ contains the regularizer $\frac{1}{2\sigma^2}\|\boldsymbol{\omega}\|_2^2$, and is therefore $\frac{1}{\sigma^2}$-strongly convex in the active coefficients. The logistic loss is convex, while the ridge term contributes a Hessian of $\frac{1}{\sigma^2}\mathbf{I}$, giving a uniform lower bound $\frac{1}{\sigma^2}\mathbf{I}$ on the Hessian of $\mathcal{L}$.

Let $\phi(z) := 1/(1+e^z)$, so that $\frac{d}{dz}\log(1+e^{-z}) = -\phi(z)$ and $\phi'(z) = -\phi(z)(1-\phi(z)) \in [-1/4, 0]$. In particular, ϕ is monotonically decreasing and $\frac{1}{4}$-Lipschitz; we will use both properties repeatedly.

Let $S \subseteq [m]$ denote the active support before the swap, and write $\hat{\boldsymbol{\omega}}_S$ for the optimal coefficients on the features in S and $\hat{\omega}_i$ for the optimal coefficient for feature i. Define the linear predictors for the l^{th} example at the original optimum and at the coordinate refit:

$$\hat{\eta}_l := \mathbf{x}_{l,S}^\top \hat{\boldsymbol{\omega}}_S + \hat{\omega}_i\, \mathbf{x}_{l,i}, \qquad \eta_l^{\mathrm{cd}} := \mathbf{x}_{l,S}^\top \hat{\boldsymbol{\omega}}_S + \omega_j^{\mathrm{cd}}\, \mathbf{x}_{l,j}.$$

The two predictors share the same component $\mathbf{x}_{l,S}^\top \hat{\boldsymbol{\omega}}_S$ on the unchanged support S, and differ only in the contribution from the swapped feature. We write $\boldsymbol{\omega}_{\mathrm{cd}} := (\hat{\boldsymbol{\omega}}_S, \omega_j^{\mathrm{cd}})$ and $\boldsymbol{\omega}_{\mathrm{full}} := (\boldsymbol{\omega}_S^{\mathrm{full}}, \omega_j^{\mathrm{full}})$, and let $\mathcal{L}(\boldsymbol{\omega})$ denote the loss in Eq. 3 restricted to support $S \cup \{j\}$ (the ℓ_0 term is constant on this support and may therefore be dropped).

Step 1: Reduction to a gradient bound. A standard consequence of μ-strong convexity is that, for any $\boldsymbol{\omega}$, the suboptimality at $\boldsymbol{\omega}$ is controlled by the squared gradient norm at $\boldsymbol{\omega}$: $f(\boldsymbol{\omega}) - \min f \le \frac{1}{2\mu}\|\nabla f(\boldsymbol{\omega})\|^2$. Applying this with $\mu = \frac{1}{\sigma^2}$ gives

$$\mathcal{L}(\boldsymbol{\omega}_{\mathrm{cd}}) - \mathcal{L}(\boldsymbol{\omega}_{\mathrm{full}}) \;\le\; \tfrac{\sigma^2}{2}\|\nabla \mathcal{L}(\boldsymbol{\omega}_{\mathrm{cd}})\|^2.$$

Now, $\boldsymbol{\omega}_{\mathrm{cd}}$ is constructed by minimizing $\mathcal{L}$ along the ω_j-direction with $\boldsymbol{\omega}_S$ fixed at $\hat{\boldsymbol{\omega}}_S$, so the partial derivative in that direction vanishes: $\partial_{\omega_j}\mathcal{L}(\boldsymbol{\omega}_{\mathrm{cd}}) = 0$. The full gradient $\nabla\mathcal{L}(\boldsymbol{\omega}_{\mathrm{cd}})$ is therefore supported entirely on the $\boldsymbol{\omega}_S$-block:

$$\|\nabla \mathcal{L}(\boldsymbol{\omega}_{\mathrm{cd}})\|^2 = \|\nabla_{\boldsymbol{\omega}_S}\mathcal{L}(\boldsymbol{\omega}_{\mathrm{cd}})\|^2.$$

The remainder of the proof bounds this single block.

Step 2: Bounding $\nabla_{\boldsymbol{\omega}_S}\mathcal{L}(\boldsymbol{\omega}_{\mathrm{cd}})$. Differentiating $\mathcal{L}$ with respect to the $\boldsymbol{\omega}_S$ coordinates at $\boldsymbol{\omega}_{\mathrm{cd}} = (\hat{\boldsymbol{\omega}}_S, \omega_j^{\mathrm{cd}})$,

$$\nabla_{\boldsymbol{\omega}_S}\mathcal{L}(\boldsymbol{\omega}_{\mathrm{cd}}) = -\tfrac{1}{n}\sum_{l=1}^{n} y_l \phi(y_l \eta_l^{\mathrm{cd}})\, \mathbf{x}_{l,S} \;+\; \tfrac{1}{\sigma^2}\hat{\boldsymbol{\omega}}_S,$$

which is the standard logistic-regression gradient evaluated at the predictor η_l^{cd}. To exploit that $\hat{\boldsymbol{\omega}}_S$ was optimal on the original support $S \cup \{i\}$, we recall its first-order optimality condition (with the predictor $\hat{\eta}_l$):

$$\tfrac{1}{\sigma^2}\hat{\boldsymbol{\omega}}_S = \tfrac{1}{n}\sum_{l=1}^{n} y_l \phi(y_l \hat{\eta}_l)\, \mathbf{x}_{l,S}.$$

Substituting this expression into the formula for $\nabla_{\boldsymbol{\omega}_S}\mathcal{L}(\boldsymbol{\omega}_{\mathrm{cd}})$,

$$\nabla_{\boldsymbol{\omega}_S}\mathcal{L}(\boldsymbol{\omega}_{\mathrm{cd}}) = \tfrac{1}{n}\sum_{l=1}^{n} y_l\big[\phi(y_l\hat{\eta}_l) - \phi(y_l\eta_l^{\mathrm{cd}})\big]\mathbf{x}_{l,S} = \tfrac{1}{n}\mathbf{X}_S^\top \boldsymbol{\delta},$$

where we have defined

$$\delta_l := y_l\big[\phi(y_l\hat{\eta}_l) - \phi(y_l\eta_l^{\mathrm{cd}})\big].$$

Thus the residual gradient is determined entirely by how much the predictor changes when we swap feature i for feature j.

To bound $\|\boldsymbol{\delta}\|_2$, note that $|y_l| = 1$, so by $\frac{1}{4}$-Lipschitzness of ϕ,

$$|\delta_l| = \big|\phi(y_l\hat{\eta}_l) - \phi(y_l\eta_l^{\mathrm{cd}})\big| \;\le\; \tfrac{1}{4}|y_l\hat{\eta}_l - y_l\eta_l^{\mathrm{cd}}| \;=\; \tfrac{1}{4}|\hat{\omega}_i\mathbf{x}_{l,i} - \omega_j^{\mathrm{cd}}\mathbf{x}_{l,j}|,$$

where in the last step the components on S canceled. Squaring and summing,

$$\|\boldsymbol{\delta}\|_2^2 \;\le\; \tfrac{1}{16}\|\hat{\omega}_i\,\mathbf{x}_i - \omega_j^{\mathrm{cd}}\,\mathbf{x}_j\|_2^2.$$

Since $\|\mathbf{X}_S\|_{\mathrm{op}}^2 = \lambda_{\max}(\mathbf{X}_S^\top\mathbf{X}_S) = n\kappa_S$,

$$\|\nabla_{\boldsymbol{\omega}_S}\mathcal{L}(\boldsymbol{\omega}_{\mathrm{cd}})\|^2 = \tfrac{1}{n^2}\|\mathbf{X}_S^\top\boldsymbol{\delta}\|_2^2 \;\le\; \tfrac{1}{n^2}\|\mathbf{X}_S\|_{\mathrm{op}}^2\,\|\boldsymbol{\delta}\|_2^2 \;\le\; \tfrac{\kappa_S}{16n}\|\hat{\omega}_i\,\mathbf{x}_i - \omega_j^{\mathrm{cd}}\,\mathbf{x}_j\|_2^2.$$

Combining with Step 1,

$$\mathcal{L}(\boldsymbol{\omega}_{\mathrm{cd}}) - \mathcal{L}(\boldsymbol{\omega}_{\mathrm{full}}) \;\le\; \tfrac{\sigma^2\kappa_S}{32\,n}\|\hat{\omega}_i\,\mathbf{x}_i - \omega_j^{\mathrm{cd}}\,\mathbf{x}_j\|_2^2. \tag{22}$$

The rest of the proof bounds the Euclidean distance $\|\hat{\omega}_i\,\mathbf{x}_i - \omega_j^{\mathrm{cd}}\,\mathbf{x}_j\|_2$ between the two single-feature contributions.

Step 3: Decomposition of the predictor gap. Expanding the square and using the assumed normalizations $\|\mathbf{x}_i\|_2^2 = \|\mathbf{x}_j\|_2^2 = n$ and $\mathbf{x}_i^\top\mathbf{x}_j = n\rho$,

$$\begin{aligned}\tfrac{1}{n}\|\hat{\omega}_i\,\mathbf{x}_i - \omega_j^{\mathrm{cd}}\,\mathbf{x}_j\|_2^2 &= \hat{\omega}_i^2 - 2\hat{\omega}_i\omega_j^{\mathrm{cd}}\rho + (\omega_j^{\mathrm{cd}})^2\\ &= \hat{\omega}_i^2(1-\rho^2) + (\omega_j^{\mathrm{cd}} - \hat{\omega}_i\rho)^2,\end{aligned}$$

where the second equality is obtained by completing the square in ω_j^{cd} around $\hat{\omega}_i\rho$.

Step 4: Bounding $|\omega_j^{\mathrm{cd}} - \hat{\omega}_i\rho|$. Define:

$$g(\omega) := \partial_\omega\mathcal{L}(\hat{\boldsymbol{\omega}}_S, \omega) = \tfrac{\omega}{\sigma^2} - \tfrac{1}{n}\sum_{l=1}^n y_l\phi\big(y_l(\mathbf{x}_{l,S}^\top\hat{\boldsymbol{\omega}}_S + \omega\,\mathbf{x}_{l,j})\big)\mathbf{x}_{l,j}.$$

By construction, $g(\omega_j^{\mathrm{cd}}) = 0$. Furthermore,

$$g'(\omega) = \tfrac{1}{\sigma^2} - \tfrac{1}{n}\sum_{l=1}^n y_l^2\,\phi'\big(y_l(\mathbf{x}_{l,S}^\top\hat{\boldsymbol{\omega}}_S + \omega\,\mathbf{x}_{l,j})\big)\mathbf{x}_{l,j}^2 = \tfrac{1}{\sigma^2} + \tfrac{1}{n}\sum_{l=1}^n\big(-\phi'(\cdot)\big)\mathbf{x}_{l,j}^2,$$

where we have used $y_l^2 = 1$. Since $-\phi'(\cdot) \in [0, 1/4]$, every term in the data-fit sum is non-negative, so $g'(\omega) \ge \frac{1}{\sigma^2}$. Equivalently, g is $\frac{1}{\sigma^2}$-strongly increasing, so for any $\omega^\star$,

$$|\omega_j^{\mathrm{cd}} - \omega^\star| \;\le\; \sigma^2\,|g(\omega_j^{\mathrm{cd}}) - g(\omega^\star)| \;=\; \sigma^2\,|g(\omega^\star)|.$$

Choosing $\omega^\star = \hat{\omega}_i\rho$,

$$|\omega_j^{\mathrm{cd}} - \hat{\omega}_i\rho| \;\le\; \sigma^2\,|g(\hat{\omega}_i\rho)|. \tag{23}$$

It remains to bound $|g(\hat{\omega}_i\rho)|$. Define the predictor at $\omega = \hat{\omega}_i\rho$,

$$\tilde{\eta}_l := \mathbf{x}_{l,S}^\top\hat{\boldsymbol{\omega}}_S + \hat{\omega}_i\rho\,\mathbf{x}_{l,j}.$$

The optimality of $\hat{\omega}_i$ on the original support yields

$$\tfrac{1}{\sigma^2}\hat{\omega}_i = \tfrac{1}{n}\sum_{l=1}^n y_l\phi(y_l\hat{\eta}_l)\mathbf{x}_{l,i}.$$

Multiplying both sides by ρ and substituting into the definition of $g(\hat{\omega}_i\rho)$,

$$g(\hat{\omega}_i\rho) = \tfrac{1}{n}\sum_{l=1}^n y_l\big[\rho\,\phi(y_l\hat{\eta}_l)\mathbf{x}_{l,i} - \phi(y_l\tilde{\eta}_l)\mathbf{x}_{l,j}\big].$$

Now decompose $\mathbf{x}_j$ into its projection onto $\mathbf{x}_i$ and an orthogonal residual:

$$\mathbf{x}_{l,j} = \rho\,\mathbf{x}_{l,i} + r_l, \qquad r_l := \mathbf{x}_{l,j} - \rho\,\mathbf{x}_{l,i}.$$

The residual norm satisfies

$$\tfrac{1}{n}\|\mathbf{r}\|_2^2 = \tfrac{1}{n}\big(\|\mathbf{x}_j\|_2^2 - 2\rho\,\mathbf{x}_i^\top\mathbf{x}_j + \rho^2\|\mathbf{x}_i\|_2^2\big) = 1 - 2\rho^2 + \rho^2 = 1 - \rho^2.$$

Substituting $\mathbf{x}_{l,j} = \rho\,\mathbf{x}_{l,i} + r_l$ in the second term of $g(\hat{\omega}_i\rho)$ and grouping,

$$g(\hat{\omega}_i\rho) = T_1 + T_2,$$

where

$$T_1 := \tfrac{\rho}{n}\sum_{l=1}^{n} y_l\big[\phi(y_l\hat{\eta}_l) - \phi(y_l\tilde{\eta}_l)\big]\mathbf{x}_{l,i}, \qquad T_2 := -\tfrac{1}{n}\sum_{l=1}^{n} y_l\phi(y_l\tilde{\eta}_l)\,r_l.$$

We bound each piece in turn.

Bound on T_2. Since $\phi \in (0,1)$ and $|y_l| = 1$, we have $|y_l\phi(\cdot)| \le 1$. By the Cauchy–Schwarz inequality,

$$|T_2| \;\le\; \tfrac{1}{n}\sum_{l=1}^{n}|r_l| \;\le\; \tfrac{1}{n}\cdot\sqrt{n}\,\|\mathbf{r}\|_2 \;=\; \tfrac{1}{\sqrt{n}}\|\mathbf{r}\|_2 \;=\; \sqrt{1-\rho^2}.$$

Bound on T_1. The two predictors $\hat{\eta}_l$ and $\tilde{\eta}_l$ differ only in the swapped-feature component, so

$$\hat{\eta}_l - \tilde{\eta}_l = \hat{\omega}_i\,\mathbf{x}_{l,i} - \hat{\omega}_i\rho\,\mathbf{x}_{l,j} = \hat{\omega}_i(\mathbf{x}_{l,i} - \rho\,\mathbf{x}_{l,j}).$$

By $\frac{1}{4}$-Lipschitzness of ϕ,

$$\big|\phi(y_l\hat{\eta}_l) - \phi(y_l\tilde{\eta}_l)\big| \;\le\; \tfrac{1}{4}|\hat{\eta}_l - \tilde{\eta}_l| \;=\; \tfrac{|\hat{\omega}_i|}{4}|\mathbf{x}_{l,i} - \rho\,\mathbf{x}_{l,j}|.$$

Note that $\|\mathbf{x}_i - \rho\,\mathbf{x}_j\|_2^2 = n(1 - 2\rho^2 + \rho^2) = n(1-\rho^2)$. Applying Cauchy–Schwarz,

$$|T_1| \;\le\; \tfrac{|\rho|}{n}\sum_{l=1}^{n}\tfrac{|\hat{\omega}_i|}{4}|\mathbf{x}_{l,i} - \rho\,\mathbf{x}_{l,j}|\,|\mathbf{x}_{l,i}| \;\le\; \tfrac{|\rho||\hat{\omega}_i|}{4n}\|\mathbf{x}_i - \rho\,\mathbf{x}_j\|_2\|\mathbf{x}_i\|_2 \;=\; \tfrac{|\rho||\hat{\omega}_i|}{4}\sqrt{1-\rho^2}.$$

Combining the two bounds and using $|\rho| \le 1$,

$$|g(\hat{\omega}_i\rho)| \;\le\; |T_1| + |T_2| \;\le\; \big(1 + \tfrac{|\hat{\omega}_i|}{4}\big)\sqrt{1-\rho^2}.$$

Substituting into (23),

$$|\omega_j^{\mathrm{cd}} - \hat{\omega}_i\rho| \;\le\; \sigma^2\big(1 + \tfrac{|\hat{\omega}_i|}{4}\big)\sqrt{1-\rho^2}. \tag{24}$$

Step 5: Combination. Substituting (24) into the decomposition in Step 3,

$$\tfrac{1}{n}\|\hat{\omega}_i\,\mathbf{x}_i - \omega_j^{\mathrm{cd}}\,\mathbf{x}_j\|_2^2 \;\le\; \hat{\omega}_i^2(1-\rho^2) + \sigma^4\big(1 + \tfrac{|\hat{\omega}_i|}{4}\big)^2(1-\rho^2).$$

Both terms carry a common factor of $(1-\rho^2)$, so we may write

$$\tfrac{1}{n}\|\hat{\omega}_i\,\mathbf{x}_i - \omega_j^{\mathrm{cd}}\,\mathbf{x}_j\|_2^2 \;\le\; \Big(\hat{\omega}_i^2 + \sigma^4\big(1 + \tfrac{|\hat{\omega}_i|}{4}\big)^2\Big)(1-\rho^2).$$

Applying the inequality $a^2 + b^2 \le (|a| + |b|)^2$,

$$\tfrac{1}{n}\|\hat{\omega}_i\,\mathbf{x}_i - \omega_j^{\mathrm{cd}}\,\mathbf{x}_j\|_2^2 \;\le\; \Big(|\hat{\omega}_i| + \sigma^2\big(1 + \tfrac{|\hat{\omega}_i|}{4}\big)\Big)^2(1-\rho^2).$$

Substituting this back into the gradient-norm bound (22),

$$\mathcal{L}(\boldsymbol{\omega}_{\mathrm{cd}}) - \mathcal{L}(\boldsymbol{\omega}_{\mathrm{full}}) \;\le\; \tfrac{\sigma^2\kappa_S}{32}\Big(|\hat{\omega}_i| + \sigma^2\big(1 + \tfrac{|\hat{\omega}_i|}{4}\big)\Big)^2(1-\rho^2),$$

which completes the proof. □

B Single Coordinate Fine Tuning Experiments

In this section, we describe the single coordinate fine tuning experiments from Section 4.2 in more detail. Consider line 5 in Algorithm 1. Suppose we are proposing a new support set S' given $S^{(t-1)}$). As an example, assume that feature j has been swapped with feature k to form S'. If $\rho(\mathbf{x}_j, \mathbf{x}_k) \geq \tau$, there is sufficient information shared between these swapped features that performing single coordinate descent only on feature j should not result in a much lower loss as compared to retraining the weight vector from scratch (see Theorem 3.1 for the intuition). Similarly, when adding a new feature, if j is sufficiently correlated to any feature present in $S^{(t-1)}$, we perform the same fine tuning procedure. When removing a feature, we check if the removed feature is sufficiently correlated with any feature in $S^{(t-1)}$.

For the experiments below (as well as in Section 4.2), we ran GRAM for 5000 iterations, setting $\sigma^2 = 0.01, p_{feat} = 0.01$, and $\epsilon = 0.02$. We run the experiments 5 times, computing the mean and standard error over 5 trials. From Figure 3, we see similar trends as in Figure 2 for NIJ Recidivism and Wine datasets - increasing the correlation threshold (aka reducing the rate of single coordinate fine tuning) increases the runtime. For Heart and Student this relationship is less clear - this is likely because these datasets are small enough for the runtime gain to be negligible (i.e. GRAM) finishes running in < 10 seconds.

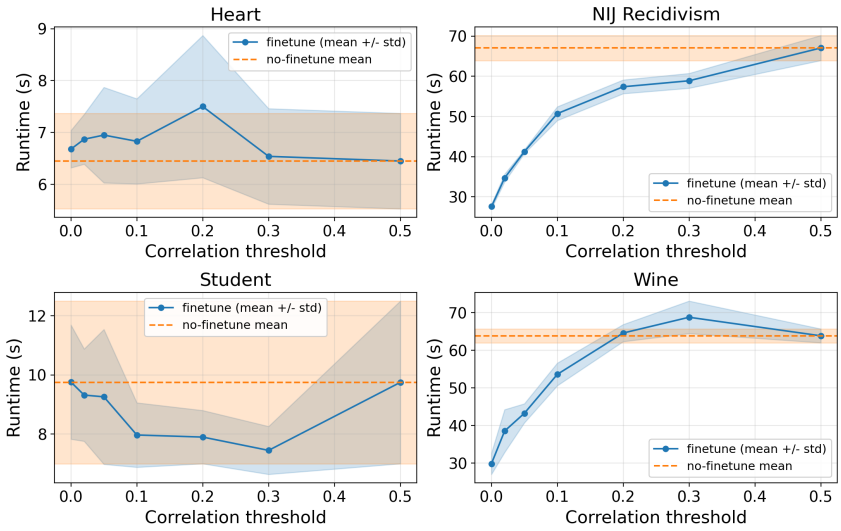


Figure 3: Runtime (seconds) across datasets at varying correlation thresholds for fine-tuning. We ran the method for 5000 iterations, with $\sigma^2 = 0.01, p_{feat} = 0.01, \epsilon = 0.02$.

From Figures 4, 5, 6, and 7, we see that the increase in runtime, however, can come with tradeoffs. In particular, this can reduce the sample efficiency of the algorithm, as evidenced by the general drop in acceptance rate and the # of models in the Rashomon set for low correlation thresholds. Luckily, there is no evident relationship between the degree of fine tuning and diversity metrics like monotonicity entropy and jaccard support diversity.

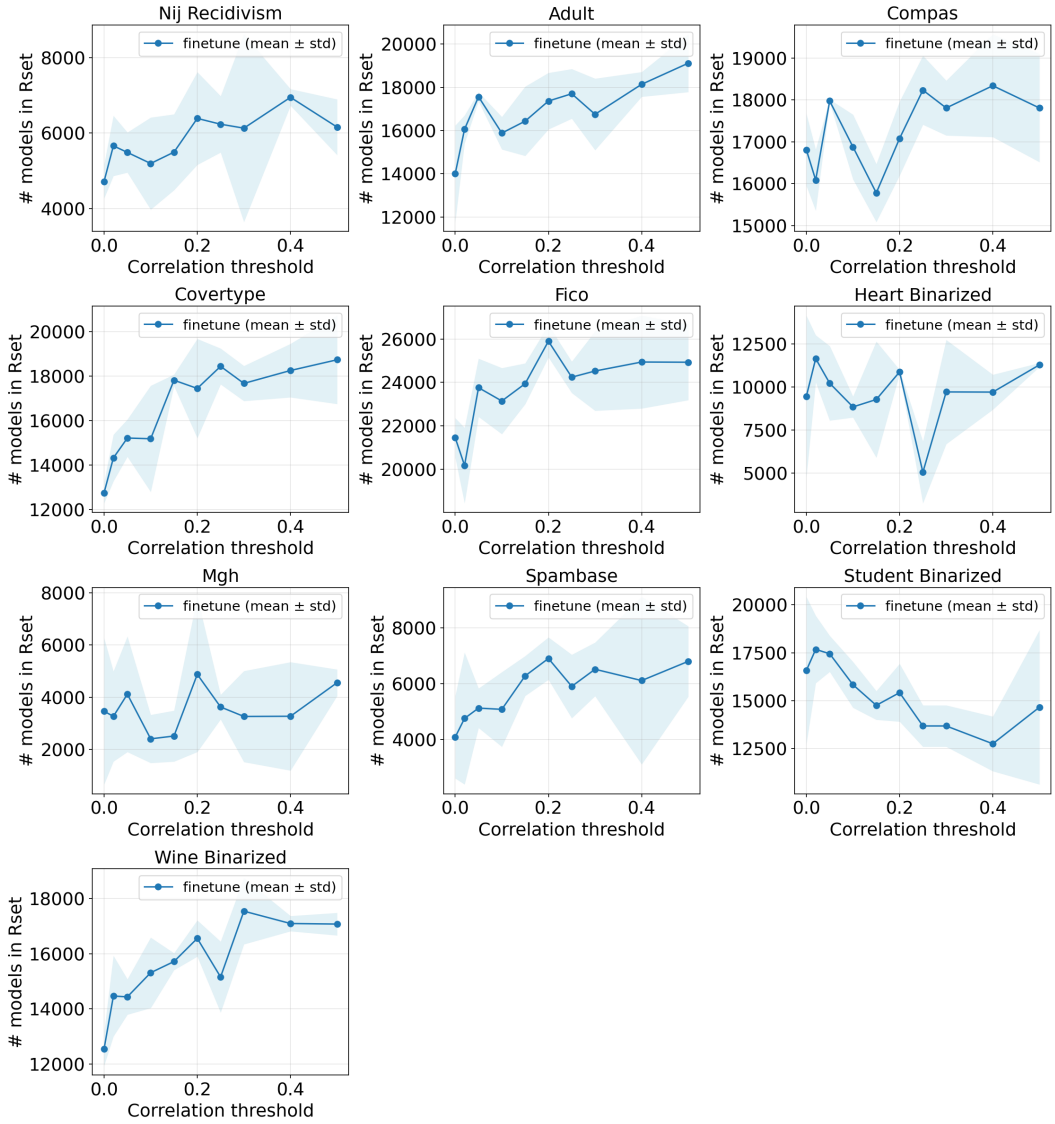


Figure 4: Number of models in the Rashomon set vs the correlation threshold τ for fine tuning.

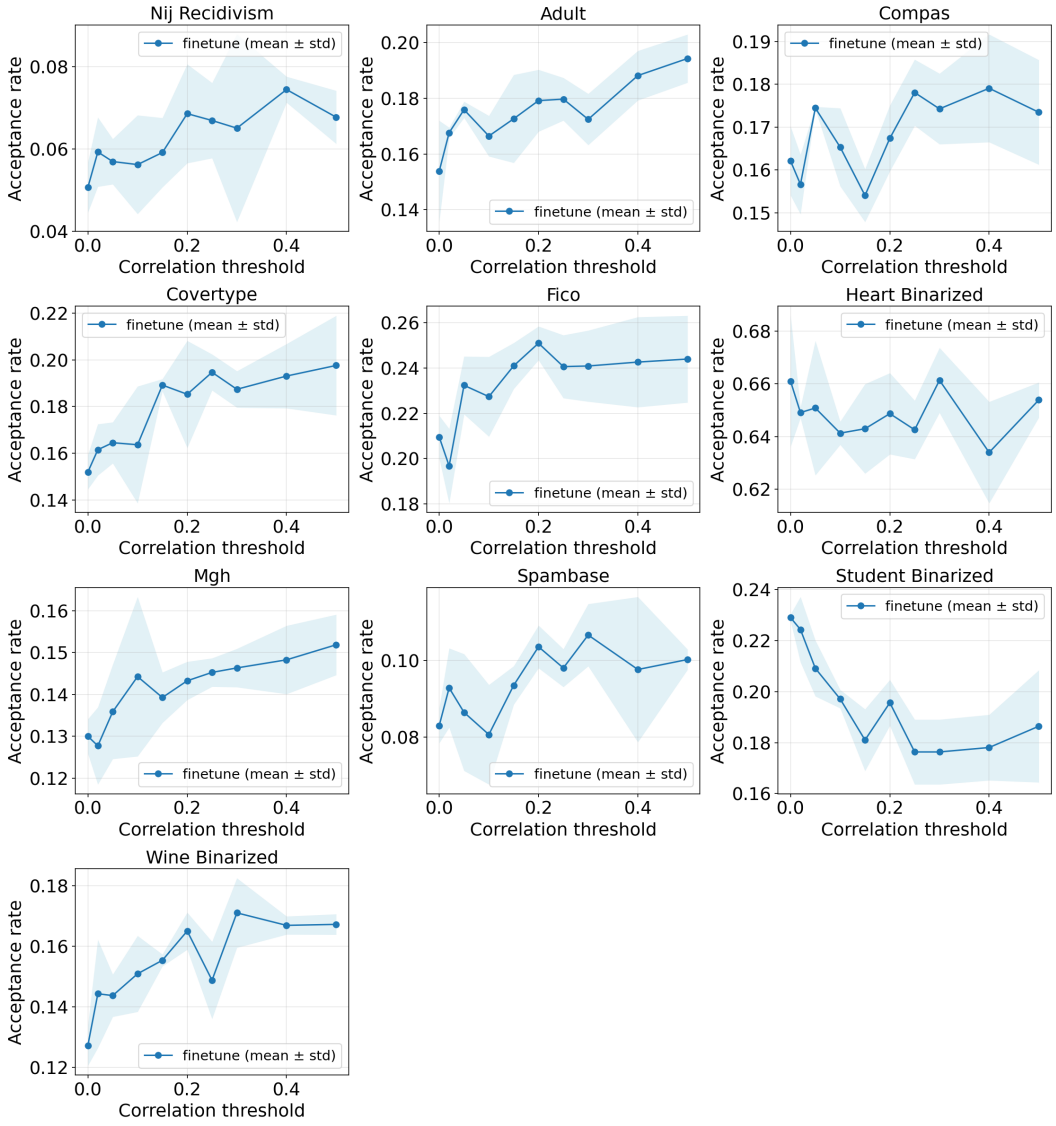


Figure 5: Acceptance probability of a support set vs correlation threshold τ for fine tuning.

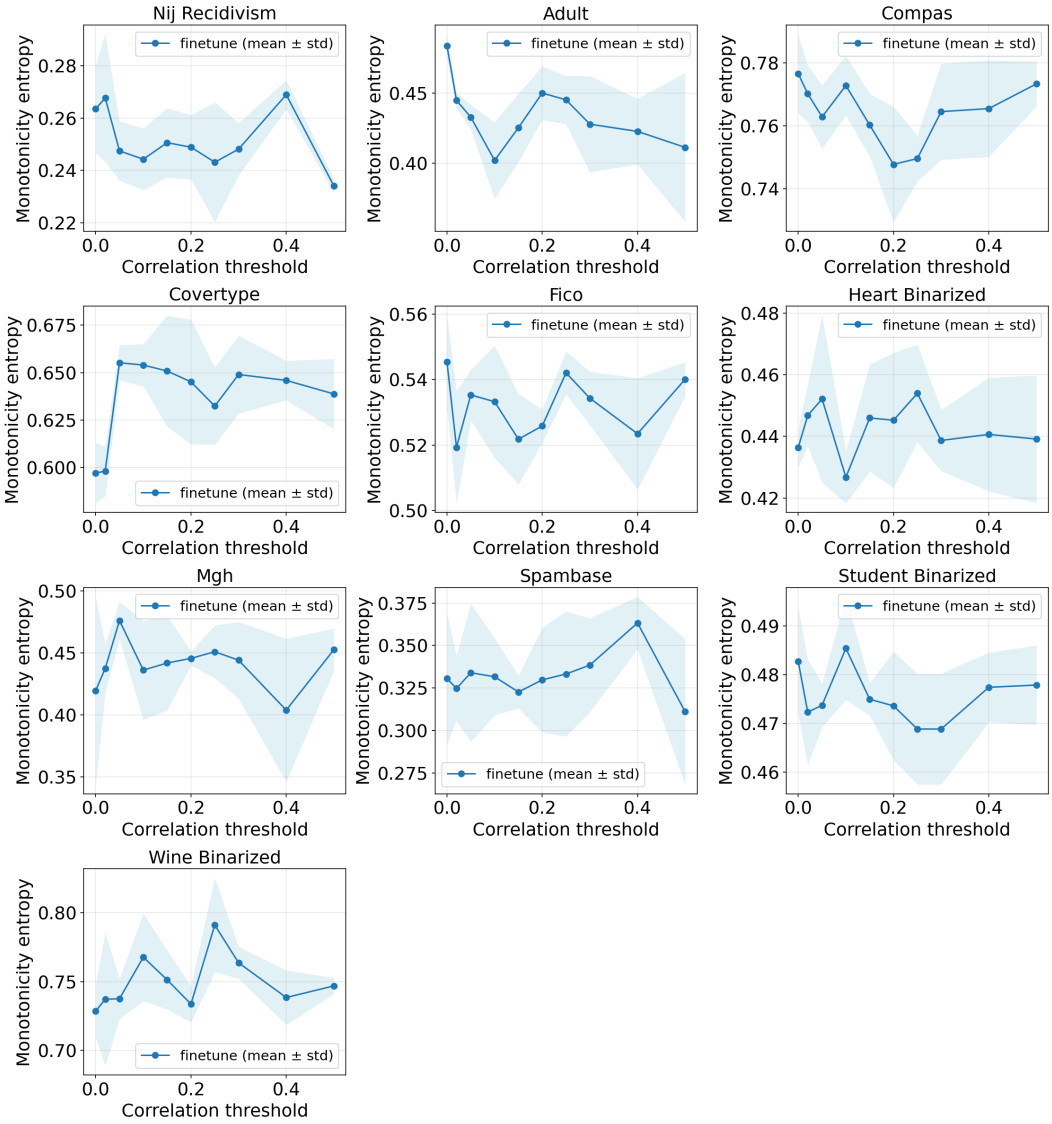


Figure 6: Monotonicity entropy vs correlation threshold τ for fine tuning.

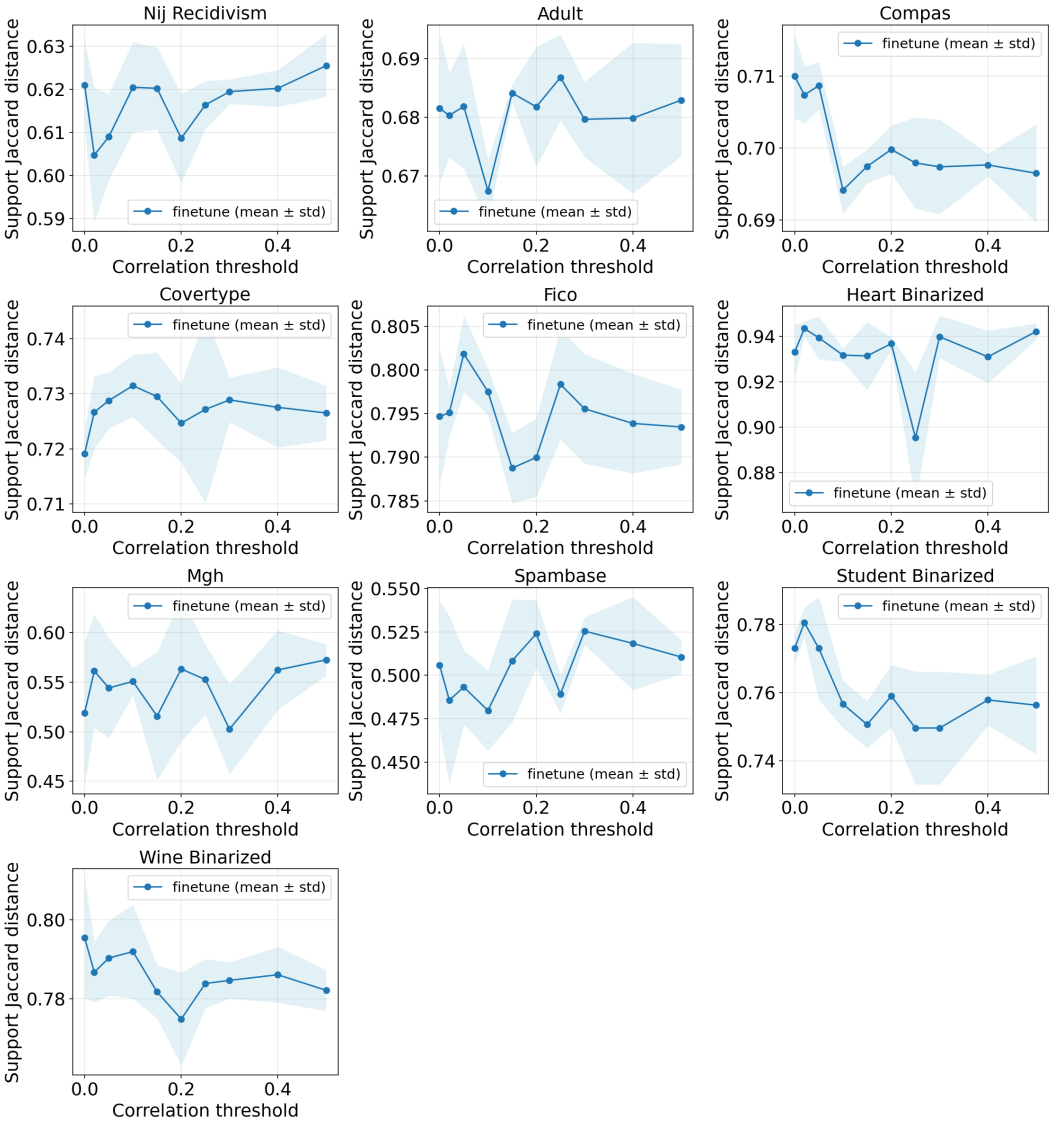


Figure 7: Jaccard support diversity vs correlation threshold τ for fine tuning.

C Variable Importance Experiments

For each dataset, we report the bootstrap mean and standard deviation of the model-reliance interval $[\min, \max]$ across the Rashomon set and its width (range), under our GRAM algorithm and the single-$\hat{w}$ ellipsoidal baseline. Each quantity is computed over five bootstrapped training datasets. The number of sampled models per method is reported as mean $\pm$ standard deviation across bootstrap trials. We show the top 5 features per dataset, ranked descending by the MCMC bootstrap-mean upper bound. Bold entries mark the method with the larger mean range for each feature; if the two methods produce identical mean ranges, both are bolded. Feature names are abbreviated for compactness.

For these experiments, we set $\epsilon = 0.02$, $\sigma^2 = 0.1$, $p_{feat} = 0.001$, number of steps $= 5000$, diversity function $\phi =$ Jaccard distance. Across all tables below, we see that GRAM consistently finds a larger range of variable importance compared to the ellipsoid method. This is useful to practitioners, who may now be able to find GAM Rashomon sets which provide more faithful notions of model class reliance, a point exemplified in [12].

Table 3: **Adult** dataset.

	GRAM (N=5,674.0 $\pm$ 803.3)		Ellipsoid (N=5,765.6 $\pm$ 799.4)	
Feature	$[\min, \max]$	Range	$[\min, \max]$	Range
edu-num	$[0.018 \pm 0.005,\ 0.053 \pm 0.004]$	$\mathbf{0.035 \pm 0.004}$	$[0.019 \pm 0.007,\ 0.049 \pm 0.008]$	0.029 ± 0.003
cap-gain	$[0.021 \pm 0.005,\ 0.035 \pm 0.005]$	0.014 ± 0.001	$[0.021 \pm 0.005,\ 0.035 \pm 0.006]$	$\mathbf{0.014 \pm 0.001}$
marital_Married	$[0.001 \pm 0.001,\ 0.022 \pm 0.003]$	0.022 ± 0.003	$[0.001 \pm 0.003,\ 0.023 \pm 0.004]$	$\mathbf{0.022 \pm 0.002}$
age	$[0.003 \pm 0.003,\ 0.019 \pm 0.003]$	$\mathbf{0.016 \pm 0.002}$	$[0.005 \pm 0.003,\ 0.020 \pm 0.003]$	0.015 ± 0.002
hrs/wk	$[0.000 \pm 0.002,\ 0.016 \pm 0.005]$	$\mathbf{0.016 \pm 0.003}$	$[0.000 \pm 0.002,\ 0.014 \pm 0.006]$	0.014 ± 0.004

Table 4: **COMPAS** dataset.

	GRAM (N=9,472.4 $\pm$ 161.9)		Ellipsoid (N=10,081.4 $\pm$ 184.5)	
Feature	$[\min, \max]$	Range	$[\min, \max]$	Range
priors	$[0.085 \pm 0.007,\ 0.152 \pm 0.012]$	$\mathbf{0.067 \pm 0.011}$	$[0.089 \pm 0.011,\ 0.150 \pm 0.011]$	0.062 ± 0.007
age	$[0.013 \pm 0.006,\ 0.078 \pm 0.006]$	$\mathbf{0.065 \pm 0.002}$	$[0.016 \pm 0.008,\ 0.079 \pm 0.006]$	0.064 ± 0.006
female	$[-0.008 \pm 0.002,\ 0.022 \pm 0.006]$	$\mathbf{0.029 \pm 0.006}$	$[-0.007 \pm 0.002,\ 0.014 \pm 0.004]$	0.020 ± 0.003
juv_crimes	$[-0.006 \pm 0.001,\ 0.017 \pm 0.002]$	$\mathbf{0.023 \pm 0.003}$	$[-0.003 \pm 0.003,\ 0.007 \pm 0.008]$	0.010 ± 0.011
charge=felony	$[-0.008 \pm 0.002,\ 0.015 \pm 0.006]$	$\mathbf{0.023 \pm 0.006}$	$[-0.005 \pm 0.005,\ 0.013 \pm 0.012]$	0.017 ± 0.017

Table 5: **Covertype** dataset.

	GRAM (N=8,093.6 $\pm$ 586.8)		Ellipsoid (N=8,140.6 $\pm$ 586.8)	
Feature	$[\min, \max]$	Range	$[\min, \max]$	Range
Elevation	$[0.196 \pm 0.008,\ 0.237 \pm 0.007]$	0.041 ± 0.003	$[0.193 \pm 0.010,\ 0.234 \pm 0.008]$	$\mathbf{0.041 \pm 0.004}$
HDist_Road	$[0.003 \pm 0.003,\ 0.026 \pm 0.004]$	$\mathbf{0.023 \pm 0.002}$	$[0.004 \pm 0.003,\ 0.026 \pm 0.004]$	0.022 ± 0.001
Hillsh_Noon	$[-0.004 \pm 0.002,\ 0.021 \pm 0.008]$	$\mathbf{0.025 \pm 0.007}$	$[-0.002 \pm 0.003,\ 0.019 \pm 0.008]$	0.022 ± 0.006
HDist_Hydro	$[-0.001 \pm 0.001,\ 0.021 \pm 0.002]$	$\mathbf{0.021 \pm 0.003}$	$[-0.001 \pm 0.002,\ 0.019 \pm 0.002]$	0.020 ± 0.002
Wild_Area3	$[-0.003 \pm 0.000,\ 0.017 \pm 0.000]$	$\mathbf{0.021 \pm 0.000}$	$[-0.002 \pm 0.000,\ 0.018 \pm 0.000]$	0.020 ± 0.000

Table 6: **FICO** dataset.

	GRAM (N=9,089.6 $\pm$ 438.6)		Ellipsoid (N=9,352.6 $\pm$ 466.3)	
Feature	$[\min, \max]$	Range	$[\min, \max]$	Range
ExtRiskEst	$[0.012 \pm 0.008,\ 0.142 \pm 0.015]$	$\mathbf{0.129 \pm 0.010}$	$[0.014 \pm 0.007,\ 0.127 \pm 0.014]$	0.112 ± 0.009
Pct_NoDelq	$[-0.005 \pm 0.003,\ 0.037 \pm 0.007]$	$\mathbf{0.042 \pm 0.005}$	$[-0.005 \pm 0.002,\ 0.031 \pm 0.010]$	0.036 ± 0.008
NetFrac_Burden	$[-0.003 \pm 0.001,\ 0.036 \pm 0.003]$	0.039 ± 0.003	$[-0.002 \pm 0.002,\ 0.039 \pm 0.004]$	$\mathbf{0.041 \pm 0.004}$
AvgMInFile	$[-0.003 \pm 0.001,\ 0.032 \pm 0.007]$	$\mathbf{0.036 \pm 0.006}$	$[-0.002 \pm 0.004,\ 0.033 \pm 0.011]$	0.035 ± 0.009
MSince_RecInq	$[-0.001 \pm 0.002,\ 0.023 \pm 0.003]$	$\mathbf{0.024 \pm 0.002}$	$[-0.000 \pm 0.001,\ 0.023 \pm 0.004]$	0.023 ± 0.004

Table 7: **Heart (binarized)** dataset.

	GRAM (N=61.2 $\pm$ 22.8)		Ellipsoid (N=68.0 $\pm$ 25.2)	
Feature	[min, max]	Range	[min, max]	Range
thal	$[0.030 \pm 0.068,\ 0.230 \pm 0.071]$	$\mathbf{0.199 \pm 0.050}$	$[0.059 \pm 0.133,\ 0.237 \pm 0.058]$	0.178 ± 0.090
cp	$[0.000 \pm 0.000,\ 0.140 \pm 0.130]$	$\mathbf{0.140 \pm 0.130}$	$[0.000 \pm 0.000,\ 0.117 \pm 0.143]$	0.117 ± 0.143
ca	$[0.007 \pm 0.015,\ 0.106 \pm 0.146]$	0.100 ± 0.137	$[0.000 \pm 0.000,\ 0.106 \pm 0.146]$	$\mathbf{0.106 \pm 0.146}$
slope	$[0.000 \pm 0.000,\ 0.018 \pm 0.041]$	$\mathbf{0.018 \pm 0.041}$	$[0.000 \pm 0.000,\ 0.000 \pm 0.000]$	0.000 ± 0.000

Table 8: **MGH** dataset.

	GRAM (N=7,663.4 $\pm$ 1,612.7)		Ellipsoid (N=7,802.6 $\pm$ 1,608.2)	
Feature	[min, max]	Range	[min, max]	Range
Bilirubin	$[-0.003 \pm 0.000,\ 0.007 \pm 0.001]$	$\mathbf{0.010 \pm 0.001}$	$[-0.002 \pm 0.001,\ 0.007 \pm 0.001]$	0.009 ± 0.001
Creatinine	$[-0.003 \pm 0.001,\ 0.005 \pm 0.001]$	0.008 ± 0.001	$[-0.003 \pm 0.001,\ 0.005 \pm 0.002]$	$\mathbf{0.008 \pm 0.002}$
Norepi	$[-0.003 \pm 0.001,\ 0.005 \pm 0.002]$	$\mathbf{0.007 \pm 0.001}$	$[-0.003 \pm 0.001,\ 0.004 \pm 0.001]$	0.006 ± 0.002
Urine_out	$[-0.003 \pm 0.001,\ 0.004 \pm 0.001]$	$\mathbf{0.007 \pm 0.001}$	$[-0.003 \pm 0.001,\ 0.004 \pm 0.001]$	$\mathbf{0.007 \pm 0.001}$
Platelets	$[-0.003 \pm 0.001,\ 0.004 \pm 0.001]$	$\mathbf{0.006 \pm 0.002}$	$[-0.002 \pm 0.001,\ 0.003 \pm 0.002]$	0.005 ± 0.002

Table 9: **NIJ Recidivism** dataset.

	GRAM (N=6,687.6 $\pm$ 503.6)		Ellipsoid (N=7,034.4 $\pm$ 544.2)	
Feature	[min, max]	Range	[min, max]	Range
Pct_Employed	$[0.056 \pm 0.005,\ 0.151 \pm 0.006]$	$\mathbf{0.095 \pm 0.007}$	$[0.055 \pm 0.008,\ 0.150 \pm 0.007]$	0.095 ± 0.013
Jobs/Yr	$[0.002 \pm 0.002,\ 0.051 \pm 0.006]$	$\mathbf{0.049 \pm 0.005}$	$[0.002 \pm 0.003,\ 0.051 \pm 0.007]$	0.048 ± 0.006
Suprv_RiskScore	$[-0.001 \pm 0.002,\ 0.034 \pm 0.009]$	$\mathbf{0.035 \pm 0.007}$	$[-0.001 \pm 0.002,\ 0.034 \pm 0.006]$	$\mathbf{0.035 \pm 0.005}$
Gang_Aff	$[-0.000 \pm 0.001,\ 0.024 \pm 0.002]$	0.024 ± 0.001	$[-0.000 \pm 0.001,\ 0.024 \pm 0.001]$	$\mathbf{0.025 \pm 0.001}$
PriorArr_PPV_0	$[-0.005 \pm 0.001,\ 0.023 \pm 0.003]$	$\mathbf{0.028 \pm 0.003}$	$[-0.005 \pm 0.001,\ 0.021 \pm 0.003]$	0.026 ± 0.002

Table 10: **Spambase** dataset.

	GRAM (N=1,347.6 $\pm$ 333.8)		Ellipsoid (N=1,378.2 $\pm$ 344.3)	
Feature	[min, max]	Range	[min, max]	Range
char_$	$[0.016 \pm 0.006,\ 0.037 \pm 0.006]$	$\mathbf{0.020 \pm 0.004}$	$[0.018 \pm 0.009,\ 0.034 \pm 0.009]$	0.016 ± 0.001
wf_hp	$[0.016 \pm 0.004,\ 0.034 \pm 0.005]$	$\mathbf{0.018 \pm 0.004}$	$[0.020 \pm 0.004,\ 0.033 \pm 0.004]$	0.013 ± 0.003
char_!	$[0.017 \pm 0.002,\ 0.032 \pm 0.001]$	$\mathbf{0.015 \pm 0.002}$	$[0.018 \pm 0.001,\ 0.030 \pm 0.002]$	0.013 ± 0.002
cap_run_avg	$[0.007 \pm 0.003,\ 0.029 \pm 0.007]$	$\mathbf{0.022 \pm 0.005}$	$[0.008 \pm 0.005,\ 0.024 \pm 0.006]$	0.016 ± 0.002
wf_remove	$[0.007 \pm 0.004,\ 0.020 \pm 0.005]$	$\mathbf{0.012 \pm 0.002}$	$[0.009 \pm 0.005,\ 0.019 \pm 0.006]$	0.010 ± 0.001

Table 11: **Student (binarized)** dataset.

	GRAM (N=1,428.6 $\pm$ 412.7)		Ellipsoid (N=1,554.6 $\pm$ 445.4)	
Feature	[min, max]	Range	[min, max]	Range
higher_no	$[-0.001 \pm 0.001,\ 0.037 \pm 0.019]$	$\mathbf{0.038 \pm 0.019}$	$[-0.007 \pm 0.002,\ 0.030 \pm 0.025]$	0.038 ± 0.022
higher_yes	$[-0.007 \pm 0.013,\ 0.030 \pm 0.017]$	$\mathbf{0.037 \pm 0.018}$	$[-0.003 \pm 0.004,\ 0.025 \pm 0.020]$	0.027 ± 0.021
failures	$[-0.003 \pm 0.002,\ 0.029 \pm 0.006]$	$\mathbf{0.032 \pm 0.008}$	$[0.000 \pm 0.000,\ 0.017 \pm 0.005]$	0.017 ± 0.005
school_GP	$[-0.002 \pm 0.000,\ 0.029 \pm 0.003]$	$\mathbf{0.030 \pm 0.003}$	$[0.000 \pm 0.000,\ 0.012 \pm 0.007]$	0.012 ± 0.007
school_MS	$[-0.003 \pm 0.002,\ 0.026 \pm 0.004]$	$\mathbf{0.029 \pm 0.003}$	$[0.000 \pm 0.000,\ 0.014 \pm 0.004]$	0.014 ± 0.004

Table 12: **Wine (binarized)** dataset.

	GRAM (N=4,923.4 $\pm$ 1,052.8)		Ellipsoid (N=5,068.2 $\pm$ 1,103.2)	
Feature	[min, max]	Range	[min, max]	Range
alcohol	$[0.004 \pm 0.005,\ 0.067 \pm 0.010]$	$\mathbf{0.064 \pm 0.006}$	$[0.008 \pm 0.008,\ 0.064 \pm 0.014]$	0.056 ± 0.011
chlor	$[-0.005 \pm 0.001,\ 0.016 \pm 0.004]$	0.021 ± 0.004	$[-0.005 \pm 0.002,\ 0.016 \pm 0.005]$	$\mathbf{0.022 \pm 0.003}$
free_SO2	$[-0.004 \pm 0.001,\ 0.016 \pm 0.002]$	$\mathbf{0.020 \pm 0.002}$	$[-0.004 \pm 0.000,\ 0.014 \pm 0.003]$	0.019 ± 0.003
sulph	$[-0.003 \pm 0.001,\ 0.015 \pm 0.002]$	$\mathbf{0.018 \pm 0.003}$	$[-0.003 \pm 0.001,\ 0.014 \pm 0.002]$	0.017 ± 0.001
density	$[-0.007 \pm 0.002,\ 0.013 \pm 0.003]$	$\mathbf{0.020 \pm 0.002}$	$[-0.005 \pm 0.004,\ 0.008 \pm 0.006]$	0.013 ± 0.009

D Diversity Optimized Sampling

In this section, we perform experiments to better understand the effect of the diversity term. In particular, we see how changing the terms in the log acceptance ratio $\log \alpha$ (from Equation 25) changes the diversity of the resulting Rashomon set.

$$\begin{aligned}\log \alpha =& J(S') - J(S^{(t-1)}) + \log q(S^{(t-1)}|S') - \log q(S'|S^{(t-1)}) \\ &+ \lambda_r \big(\log \phi(S', R) - \log \phi(S^{(t-1)}, R)\big).\end{aligned} \tag{25}$$

For the experiments below, we set $\epsilon = 0.05$, $p_{feat} = 0.02$, number of steps = 5000. For each support set, we sample 20 GAMs around the ellipsoid. We now change two parameters in the $\log \alpha$ term: the diversity penalty λ_r and ϕ. We vary λ_r to take one of the following values: $\{0, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$. The diversity function is one of $\phi(S', R) \in \{$ Monotonicity Entropy, L_1, Jaccard Distance, and Prediction Hamming Distance$\}$. For each $\{\lambda_r, \phi\}$ pair, we conduct 5 trials and report the mean and standard error.

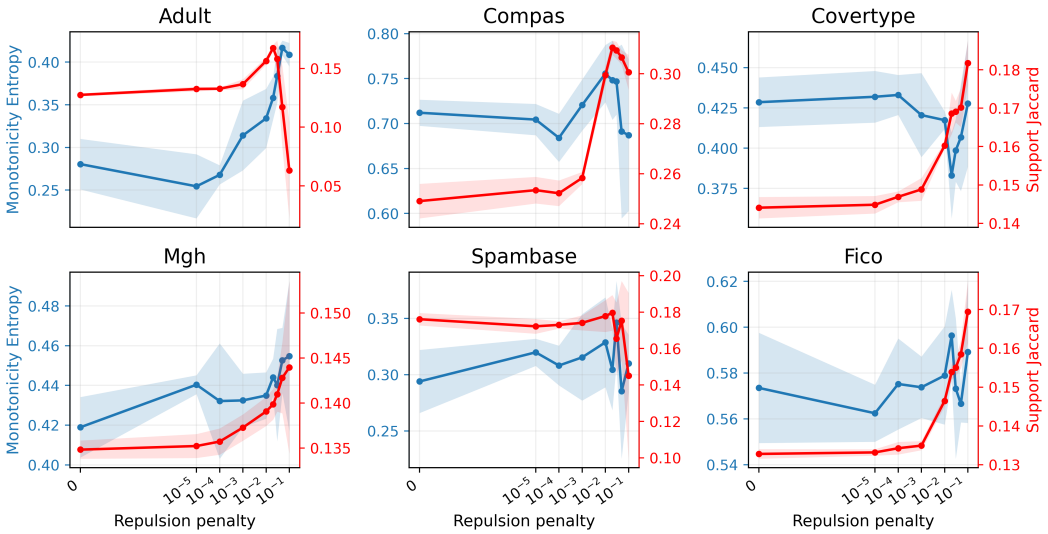


Figure 8: Repulsion penalty vs diversity measure $\in$ {Monotonicity Entropy, Support Jaccard} for 6 datasets.

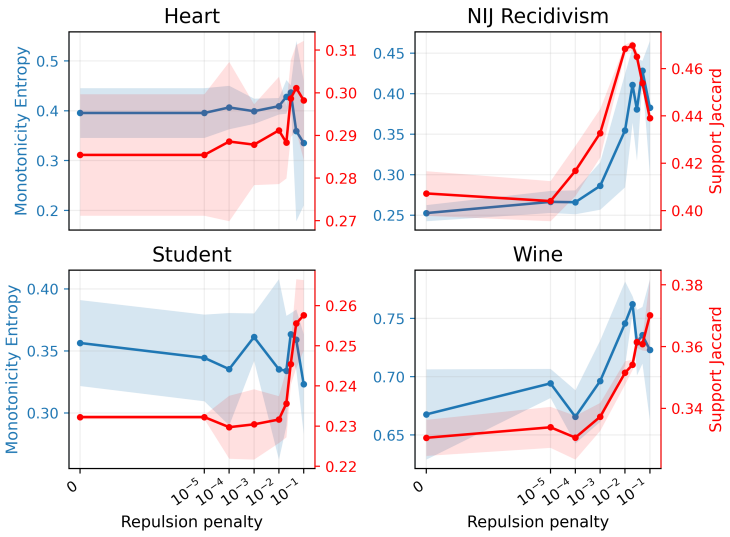


Figure 9: Repulsion penalty vs diversity measure $\in$ {Monotonicity Entropy, Support Jaccard} for 4 datasets.

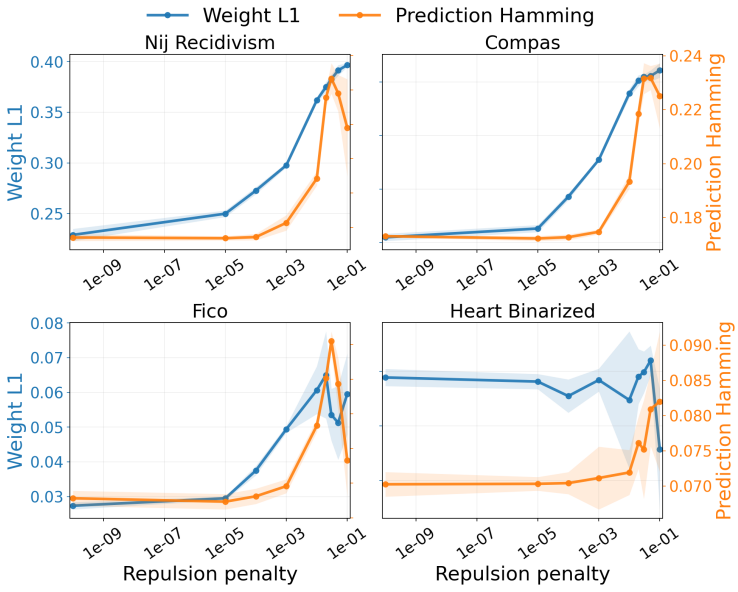


Figure 10: Repulsion penalty vs diversity measure $\in$ {Prediction Hamming, L1} for 4 datasets.

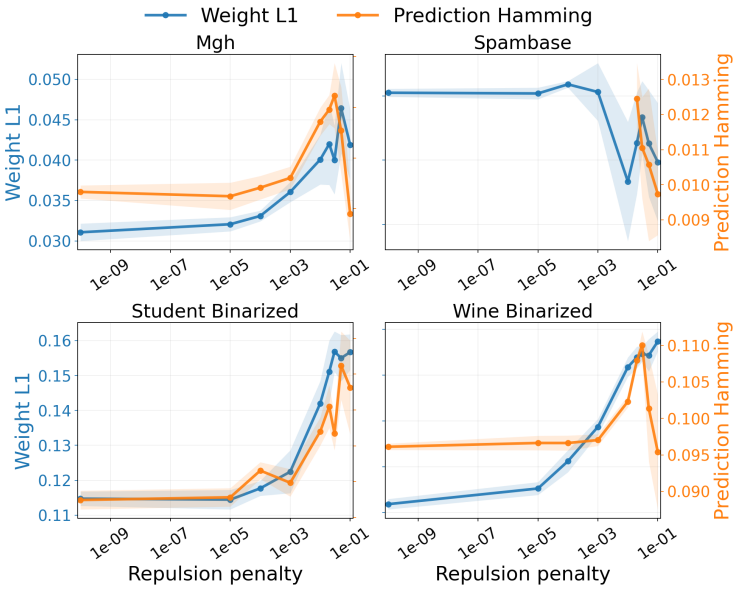


Figure 11: Repulsion penalty vs diversity measure $\in \{$Prediction Hamming, L1$\}$ for 4 datasets.

E Ablation Studies

In this section, we perform ablation studies to test various components of GRAM. These include the posterior-only sampler (i.e. without the diversity term), the smoothness term in Equation 20, and the adaptive proposal in Equation 15.

E.1 With vs Without diversity sampling

In this experiment, we disabled the diversity sampling term of GRAM, i.e. we set $\lambda_r = 0$ while keeping all other components intact (i.e. the smoothness term and adaptive proposal).

Table 13: MCMC ablation: full GRAM vs. posterior only ($\lambda_r = 0$). Top: set size, accuracy range and runtime; bottom: diversity metrics. Entries are mean $\pm$ standard error over 5 trials; in each column, the better mean of the two arms is bolded (both if tied). Ellipsoid augmentation is off ($\epsilon = 0.02$, $\sigma^2 = 0.1$, $p_{\text{feat}} = 0.01$, 1000 MH steps).

			Train acc. (%)		
Dataset	Arm	# Models ↑	Min ↑	Max ↑	Time (s) ↓
COMPAS	GRAM (full)	$\mathbf{281.6 \pm 9.5}$	66.64 ± 0.12	68.68 ± 0.03	7.3 ± 0.1
	Posterior only	191.0 ± 11.6	$\mathbf{67.56 \pm 0.07}$	$\mathbf{68.73 \pm 0.03}$	$\mathbf{6.8 \pm 0.3}$
FICO	GRAM (full)	$\mathbf{489.8 \pm 17.8}$	71.86 ± 0.10	$\mathbf{73.05 \pm 0.04}$	85.0 ± 3.1
	Posterior only	175.0 ± 80.8	$\mathbf{72.47 \pm 0.06}$	73.03 ± 0.07	$\mathbf{59.6 \pm 2.1}$
Spambase	GRAM (full)	$\mathbf{199.8 \pm 54.9}$	$\mathbf{94.99 \pm 0.03}$	$\mathbf{95.49 \pm 0.04}$	213.7 ± 2.3
	Posterior only	32.8 ± 3.5	94.66 ± 0.09	94.94 ± 0.07	$\mathbf{47.7 \pm 4.3}$
MGH	GRAM (full)	$\mathbf{116.0 \pm 43.9}$	$\mathbf{96.43 \pm 0.06}$	$\mathbf{96.70 \pm 0.09}$	146.5 ± 13.6
	Posterior only	105.6 ± 29.7	96.26 ± 0.02	96.64 ± 0.03	$\mathbf{52.2 \pm 1.9}$
Adult	GRAM (full)	93.8 ± 55.6	83.10 ± 0.06	83.40 ± 0.06	$\mathbf{6278.6 \pm 335.3}$
	Posterior only	$\mathbf{165.0 \pm 32.2}$	$\mathbf{83.26 \pm 0.02}$	$\mathbf{83.58 \pm 0.06}$	7631.6 ± 886.6

		Diversity ($\times 10^2$) ↑				
Dataset	Arm	Pred. Ham.	Supp. Ham.	Weight L_1	Shape L_1	Mono. Ent.
COMPAS	GRAM (full)	$\mathbf{9.55 \pm 0.13}$	$\mathbf{35.33 \pm 0.26}$	$\mathbf{14.12 \pm 0.09}$	$\mathbf{27.66 \pm 0.88}$	$\mathbf{47.32 \pm 1.15}$
	Posterior only	6.58 ± 0.26	27.59 ± 1.47	10.38 ± 0.62	19.75 ± 0.93	39.35 ± 3.66
FICO	GRAM (full)	$\mathbf{6.14 \pm 0.14}$	$\mathbf{13.26 \pm 0.40}$	$\mathbf{2.39 \pm 0.08}$	$\mathbf{20.15 \pm 0.70}$	$\mathbf{46.63 \pm 1.03}$
	Posterior only	2.78 ± 0.27	6.51 ± 0.71	1.05 ± 0.12	11.90 ± 1.30	36.83 ± 3.23
Spambase	GRAM (full)	$\mathbf{0.73 \pm 0.09}$	$\mathbf{3.87 \pm 0.54}$	$\mathbf{0.69 \pm 0.09}$	$\mathbf{15.19 \pm 1.78}$	$\mathbf{18.88 \pm 2.01}$
	Posterior only	0.33 ± 0.04	0.56 ± 0.06	0.15 ± 0.01	6.36 ± 0.46	5.69 ± 1.49
MGH	GRAM (full)	0.31 ± 0.07	3.33 ± 0.83	0.63 ± 0.15	$\mathbf{23.02 \pm 4.95}$	11.06 ± 3.61
	Posterior only	$\mathbf{0.38 \pm 0.02}$	$\mathbf{3.83 \pm 0.76}$	$\mathbf{0.88 \pm 0.12}$	11.15 ± 1.58	$\mathbf{16.54 \pm 2.99}$
Adult	GRAM (full)	$\mathbf{1.73 \pm 0.26}$	1.37 ± 0.55	0.38 ± 0.11	$\mathbf{33.87 \pm 11.40}$	$\mathbf{14.16 \pm 3.61}$
	Posterior only	0.95 ± 0.09	$\mathbf{2.22 \pm 0.42}$	$\mathbf{0.55 \pm 0.09}$	28.22 ± 10.79	11.11 ± 2.96

From Table 13 we see that keeping the diversity term significantly improves the diversity of GAMs in the Rashomon set whilst increasing the # models in the Rashomon set per 1000 steps. While the diversity term has mixed impact on the accuracy of models in the Rashomon set, it generally increases the runtime of the algorithm. This illustrates the tradeoffs associated with the algorithm - we can improve the diversity of the Rashomon set, but it can come at a cost of increased runtime.

E.2 With vs Without smoothness penalty

Table 14: MCMC ablation: full GRAM vs. no smoothness. Removing the smoothness guide from the search score. Smoothness is the average roughness penalty of models in the set (lower is smoother). Entries are mean ± standard error over 5 trials; in each column, the better mean of the two arms is bolded (both if tied). Ellipsoid augmentation is off ($\epsilon = 0.02$, $\sigma^2 = 0.1$, $p_{\text{feat}} = 0.01$, 1000 MH steps).

Dataset	Arm	# Models ↑	Smooth. ↓	Time (s) ↓
COMPAS	GRAM (full)	$\mathbf{281.6 \pm 9.5}$	$\mathbf{20.4 \pm 0.4}$	7.3 ± 0.1
	No smoothness	270.2 ± 5.4	23.7 ± 0.1	$\mathbf{6.8 \pm 0.1}$
FICO	GRAM (full)	$\mathbf{489.8 \pm 17.8}$	$\mathbf{42.3 \pm 1.4}$	85.0 ± 3.1
	No smoothness	472.2 ± 17.6	43.1 ± 1.3	$\mathbf{55.1 \pm 0.6}$
Spambase	GRAM (full)	199.8 ± 54.9	$\mathbf{145.5 \pm 3.0}$	213.7 ± 2.3
	No smoothness	$\mathbf{212.4 \pm 42.3}$	166.2 ± 1.6	$\mathbf{97.5 \pm 1.1}$
MGH	GRAM (full)	116.0 ± 43.9	$\mathbf{285.5 \pm 8.8}$	146.5 ± 13.6
	No smoothness	$\mathbf{213.4 \pm 53.5}$	286.3 ± 6.0	$\mathbf{77.9 \pm 4.7}$
Adult	GRAM (full)	$\mathbf{93.8 \pm 55.6}$	203.7 ± 18.8	$\mathbf{6278.6 \pm 335.3}$
	No smoothness	35.4 ± 7.6	$\mathbf{184.1 \pm 30.9}$	7813.2 ± 455.0

From Table 14 we see that keeping the smoothness term reduces the smoothness loss (in Equation 20), making sure that the GAMs have smoother shape functions. As before, adding this term causes a slight increase in runtime, mainly associated with computation of the smoothness term.

E.3 With vs Without adaptive proposal

Table 15: MCMC ablation: full GRAM vs. non-adaptive proposal. Replacing the adaptive proposal with a non-adaptive one. Best and worst are the highest and lowest training log-likelihood among models in the set (higher is better). Entries are mean ± standard error over 5 trials; in each column, the better mean of the two arms is bolded (both if tied). Ellipsoid augmentation is off ($\epsilon = 0.02$, $\sigma^2 = 0.1$, $p_{\text{feat}} = 0.01$, 1000 MH steps).

			Train acc. (%)		$\log p(D \mid \omega)$		
Dataset	Arm	# Models ↑	Min ↑	Max ↑	Best ↑	Worst ↑	Time (s) ↓
COMPAS	GRAM (full)	281.6 ± 9.5	66.64 ± 0.12	$\mathbf{68.68 \pm 0.03}$	$\mathbf{-4152.0 \pm 3.2}$	$\mathbf{-4232.2 \pm 2.8}$	$\mathbf{7.3 \pm 0.1}$
	Nonadaptive	$\mathbf{426.4 \pm 11.3}$	$\mathbf{66.76 \pm 0.10}$	68.62 ± 0.03	-4160.3 ± 2.5	-4237.6 ± 4.7	16.8 ± 0.3
FICO	GRAM (full)	489.8 ± 17.8	$\mathbf{71.86 \pm 0.10}$	$\mathbf{73.05 \pm 0.04}$	-5695.0 ± 2.5	-5802.6 ± 4.6	85.0 ± 3.1
	Nonadaptive	$\mathbf{567.2 \pm 17.5}$	71.84 ± 0.11	72.95 ± 0.05	$\mathbf{-5694.9 \pm 3.4}$	$\mathbf{-5797.3 \pm 7.4}$	$\mathbf{46.8 \pm 1.1}$
Spambase	GRAM (full)	199.8 ± 54.9	$\mathbf{94.99 \pm 0.03}$	$\mathbf{95.49 \pm 0.04}$	$\mathbf{-732.0 \pm 2.5}$	$\mathbf{-746.5 \pm 2.6}$	213.7 ± 2.3
	Nonadaptive	$\mathbf{200.4 \pm 84.0}$	94.36 ± 0.13	94.98 ± 0.07	-795.3 ± 2.8	-811.1 ± 2.9	$\mathbf{93.6 \pm 5.4}$
MGH	GRAM (full)	116.0 ± 43.9	$\mathbf{96.43 \pm 0.06}$	$\mathbf{96.70 \pm 0.09}$	$\mathbf{-1273.2 \pm 17.3}$	$\mathbf{-1293.2 \pm 14.2}$	146.5 ± 13.6
	Nonadaptive	$\mathbf{230.6 \pm 67.9}$	96.11 ± 0.05	96.50 ± 0.07	-1343.2 ± 13.2	-1363.7 ± 12.4	$\mathbf{43.3 \pm 2.4}$
Adult	GRAM (full)	$\mathbf{93.8 \pm 55.6}$	$\mathbf{83.10 \pm 0.06}$	$\mathbf{83.40 \pm 0.06}$	$\mathbf{-26721.2 \pm 46.0}$	$\mathbf{-27221.8 \pm 45.2}$	6278.6 ± 335.3
	Nonadaptive	39.4 ± 6.9	82.97 ± 0.06	83.39 ± 0.05	-26779.3 ± 31.0	-27292.6 ± 49.3	$\mathbf{2835.1 \pm 150.7}$

From Table 15 we see that keeping the adaptive proposal generally improves the accuracy of the models found in the Rashomon set, as well as the log likelihood under the modeled distribution. The effect on the number of models in the Rashomon set is mixed, while adding the adaptive proposal increases the runtime.

E.4 Swap + add + remove proposal vs Swap only proposal

Table 16: MCMC ablation: full GRAM vs. swap only. Restricting the sampler to swap moves. Best and worst are the highest and lowest training log-likelihood among models in the set (higher is better). Top: set size, accuracy, likelihood range and runtime; bottom: diversity metrics. Entries are mean $\pm$ standard error over 5 trials; in each column, the better mean of the two arms is bolded (both if tied). Ellipsoid augmentation is off ($\epsilon = 0.02$, $\sigma^2 = 0.1$, $p_{\text{feat}} = 0.01$, 1000 MH steps).

			Train acc. (%)		$\log p(D \mid \omega)$		
Dataset	Arm	# Models ↑	Min ↑	Max ↑	Best ↑	Worst ↑	Time (s) ↓
COMPAS	GRAM (full)	281.6 ± 9.5	66.64 ± 0.12	68.68 ± 0.03	-4152.0 ± 3.2	-4232.2 ± 2.8	$\mathbf{7.3 \pm 0.1}$
	Swap only	$\mathbf{490.6 \pm 4.8}$	$\mathbf{67.10 \pm 0.13}$	$\mathbf{68.72 \pm 0.02}$	$\mathbf{-4151.9 \pm 1.3}$	$\mathbf{-4211.3 \pm 2.2}$	11.5 ± 0.1
FICO	GRAM (full)	489.8 ± 17.8	71.86 ± 0.10	$\mathbf{73.05 \pm 0.04}$	-5695.0 ± 2.5	$\mathbf{-5802.6 \pm 4.6}$	85.0 ± 3.1
	Swap only	$\mathbf{497.6 \pm 31.9}$	$\mathbf{71.92 \pm 0.07}$	72.94 ± 0.03	$\mathbf{-5693.5 \pm 3.3}$	-5804.0 ± 4.3	$\mathbf{43.0 \pm 0.5}$
Spambase	GRAM (full)	$\mathbf{199.8 \pm 54.9}$	$\mathbf{94.99 \pm 0.03}$	$\mathbf{95.49 \pm 0.04}$	$\mathbf{-732.0 \pm 2.5}$	$\mathbf{-746.5 \pm 2.6}$	213.7 ± 2.3
	Swap only	112.8 ± 34.9	94.12 ± 0.04	94.74 ± 0.06	-810.1 ± 6.9	-825.8 ± 6.9	$\mathbf{29.4 \pm 3.8}$
MGH	GRAM (full)	$\mathbf{116.0 \pm 43.9}$	$\mathbf{96.43 \pm 0.06}$	$\mathbf{96.70 \pm 0.09}$	$\mathbf{-1273.2 \pm 17.3}$	$\mathbf{-1293.2 \pm 14.2}$	146.5 ± 13.6
	Swap only	20.2 ± 9.7	95.93 ± 0.04	96.20 ± 0.09	-1392.3 ± 2.9	-1411.7 ± 5.7	$\mathbf{13.5 \pm 0.1}$
Adult	GRAM (full)	$\mathbf{93.8 \pm 55.6}$	$\mathbf{83.10 \pm 0.06}$	$\mathbf{83.40 \pm 0.06}$	$\mathbf{-26721.2 \pm 46.0}$	$\mathbf{-27221.8 \pm 45.2}$	6278.6 ± 335.3
	Swap only	22.0 ± 4.2	82.92 ± 0.02	83.35 ± 0.02	-26791.4 ± 5.3	-27237.6 ± 24.6	$\mathbf{1933.8 \pm 31.1}$

		Diversity ($\times 10^2$) ↑				
Dataset	Arm	Pred. Ham.	Supp. Ham.	Weight L_1	Shape L_1	Mono. Ent.
COMPAS	GRAM (full)	$\mathbf{9.55 \pm 0.13}$	35.33 ± 0.26	14.12 ± 0.09	$\mathbf{27.66 \pm 0.88}$	47.32 ± 1.15
	Swap only	8.12 ± 0.16	$\mathbf{42.50 \pm 0.22}$	$\mathbf{14.13 \pm 0.12}$	24.44 ± 0.36	$\mathbf{53.15 \pm 0.51}$
FICO	GRAM (full)	$\mathbf{6.14 \pm 0.14}$	$\mathbf{13.26 \pm 0.40}$	$\mathbf{2.39 \pm 0.08}$	$\mathbf{20.15 \pm 0.70}$	46.63 ± 1.03
	Swap only	5.89 ± 0.17	9.81 ± 0.20	1.99 ± 0.04	18.53 ± 0.31	$\mathbf{49.44 \pm 0.93}$
Spambase	GRAM (full)	0.73 ± 0.09	$\mathbf{3.87 \pm 0.54}$	$\mathbf{0.69 \pm 0.09}$	$\mathbf{15.19 \pm 1.78}$	18.88 ± 2.01
	Swap only	$\mathbf{1.03 \pm 0.13}$	1.32 ± 0.37	0.43 ± 0.10	14.93 ± 2.56	$\mathbf{20.21 \pm 2.93}$
MGH	GRAM (full)	0.31 ± 0.07	$\mathbf{3.33 \pm 0.83}$	$\mathbf{0.63 \pm 0.15}$	$\mathbf{23.02 \pm 4.95}$	$\mathbf{11.06 \pm 3.61}$
	Swap only	$\mathbf{0.40 \pm 0.05}$	0.68 ± 0.11	0.38 ± 0.07	14.96 ± 3.47	7.26 ± 3.67
Adult	GRAM (full)	$\mathbf{1.73 \pm 0.26}$	$\mathbf{1.37 \pm 0.55}$	$\mathbf{0.38 \pm 0.11}$	$\mathbf{33.87 \pm 11.40}$	$\mathbf{14.16 \pm 3.61}$
	Swap only	$\mathbf{1.73 \pm 0.15}$	0.15 ± 0.01	0.07 ± 0.01	22.62 ± 3.36	2.51 ± 1.79

From Table 16 we see that compared to a swap proposal, using a swap+add+remove proposal has a mixed impact on the number of models in the Rashomon set. However, it generally improves the accuracy and log likelihood of models while also ensuring that the models are generally more diverse across various metrics tested.

F Visualizations of Rashomon Sets found by GRAM vs Baselines

For these visualizations, we pre-process the dataset so that the support set pool m contains at least 50% of the binary thresholds associated with each continuous feature. We also add further thresholds found using the threshold guessing algorithm from [28], setting num estimators to $= 200$. We set $\epsilon = 0.025, \sigma^2 = 0.1, p_{feat} = 1e-4$. For the ellipsoid method, we set the support set to be that of the optimal GAM found by GRAM. For a fair comparison, the upper bound for this method matches that of GRAM - this constrains the size of the ellipsoid. For the swapping method, we set the beam width $= 15$, the depth $= 2$, and increase the epsilon value to be a more generous 0.03, since there is no absolute Rashomon bound argument available in this implementation.

A small value of p_{feat} ensures that the chosen support sets are sparse - despite this sparsity, from the figures below, we see that our method provides substantially more diverse solutions compared to the ellipsoid and swapping baselines.

F.1 MGH Dataset

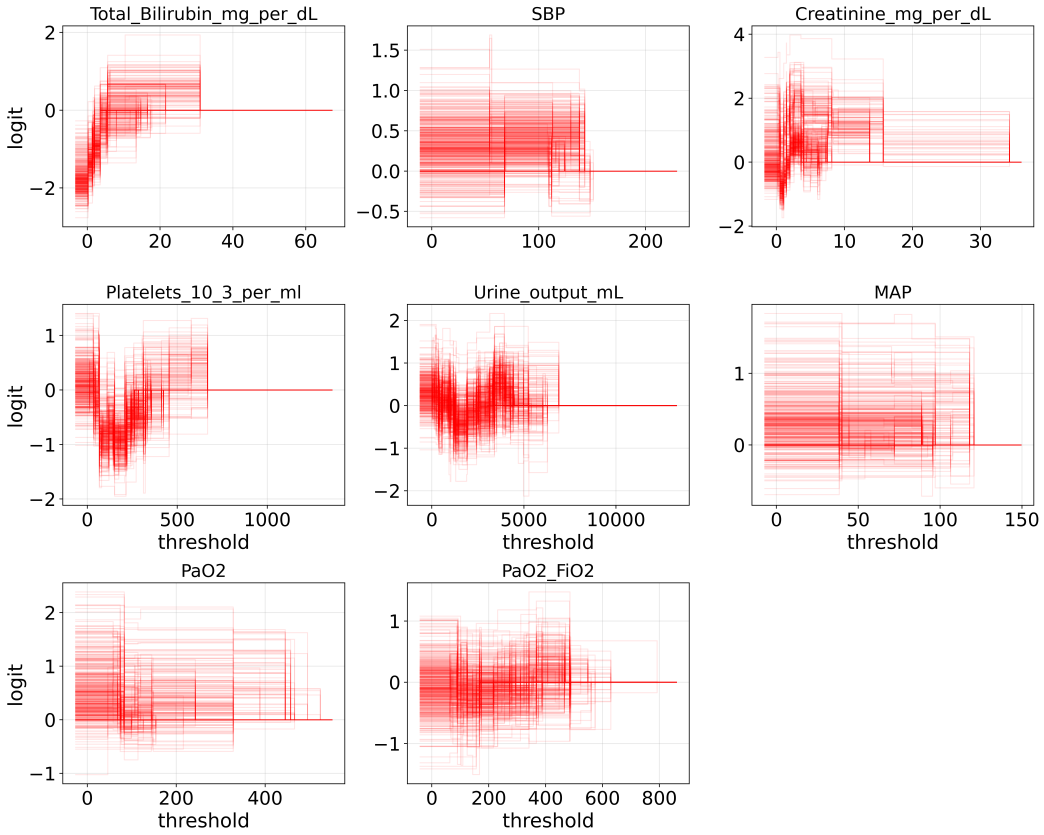


Figure 12: Illustration of GAMs output by GRAM. The average support set size is ≈ 51. We set $\epsilon = 0.025, \sigma^2 = 0.05, p_{feat} = 1e-4$.

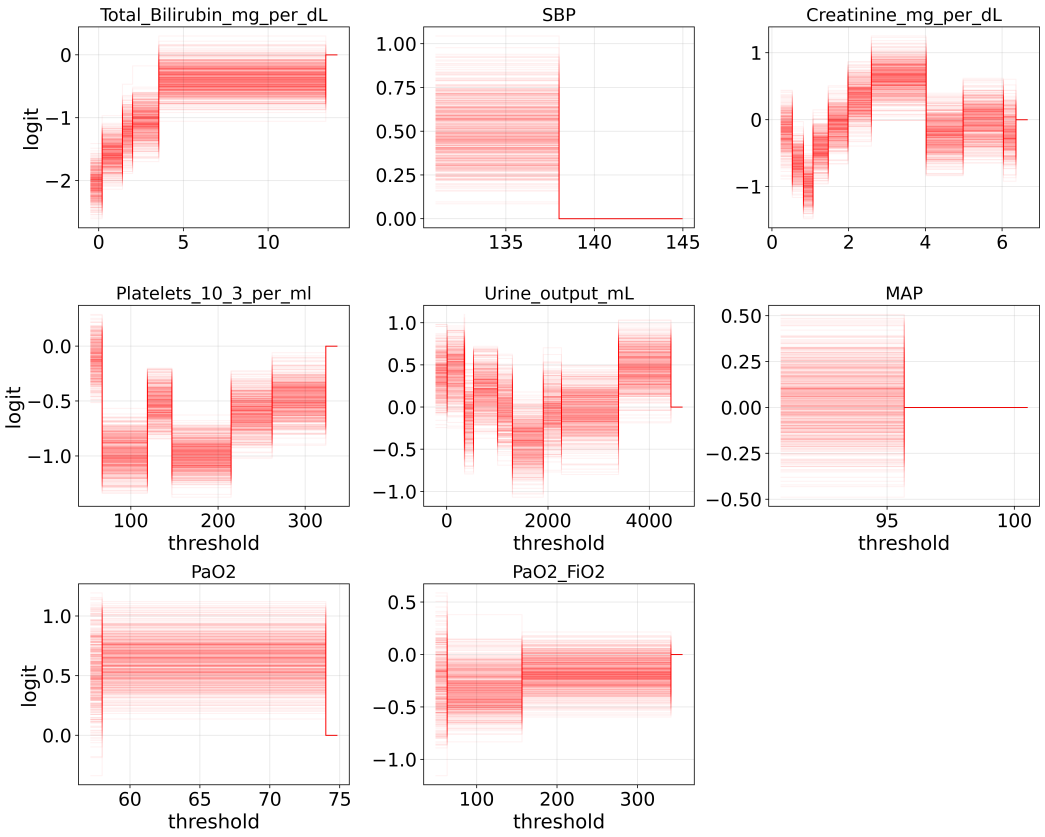


Figure 13: Illustration of GAMs output by the ellipsoid method in [36] for 8 features in the MGH dataset. For this visualization, we set the Rashomon upper bound to match that of GRAM in Figure 12. We chose the optimal support set found by GRAM for this ellipsoid - it is of size 50.

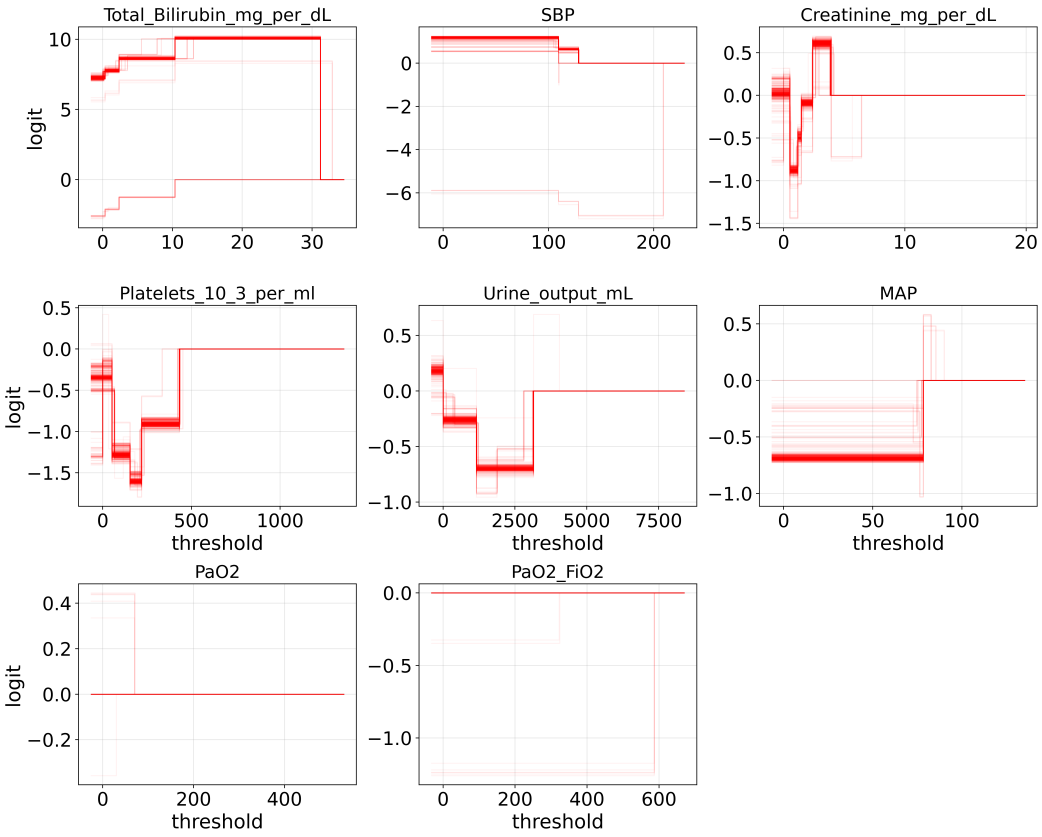


Figure 14: Illustration of GAMs output by the swapping method in [24] for 8 features in the MGH dataset. For this visualization, we set the Rashomon upper bound to match that of GRAM in Figure 12.

F.2 FICO Dataset

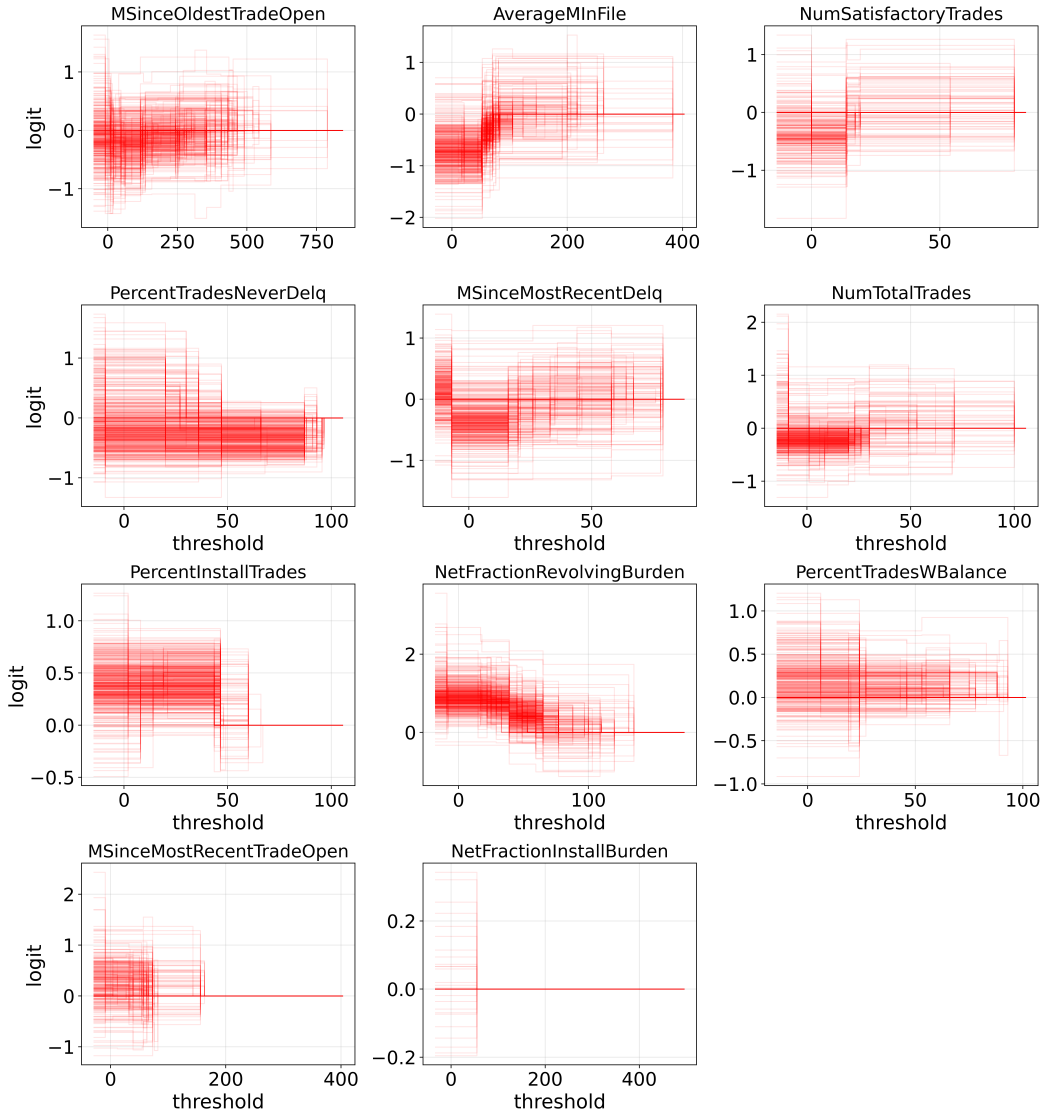


Figure 15: Illustration of GAMs output by GRAM on the FICO dataset (11 shape functions). The average support set size is ≈ 19. We set $\epsilon = 0.025, \sigma^2 = 0.05, p_{feat} = 1e-4$.

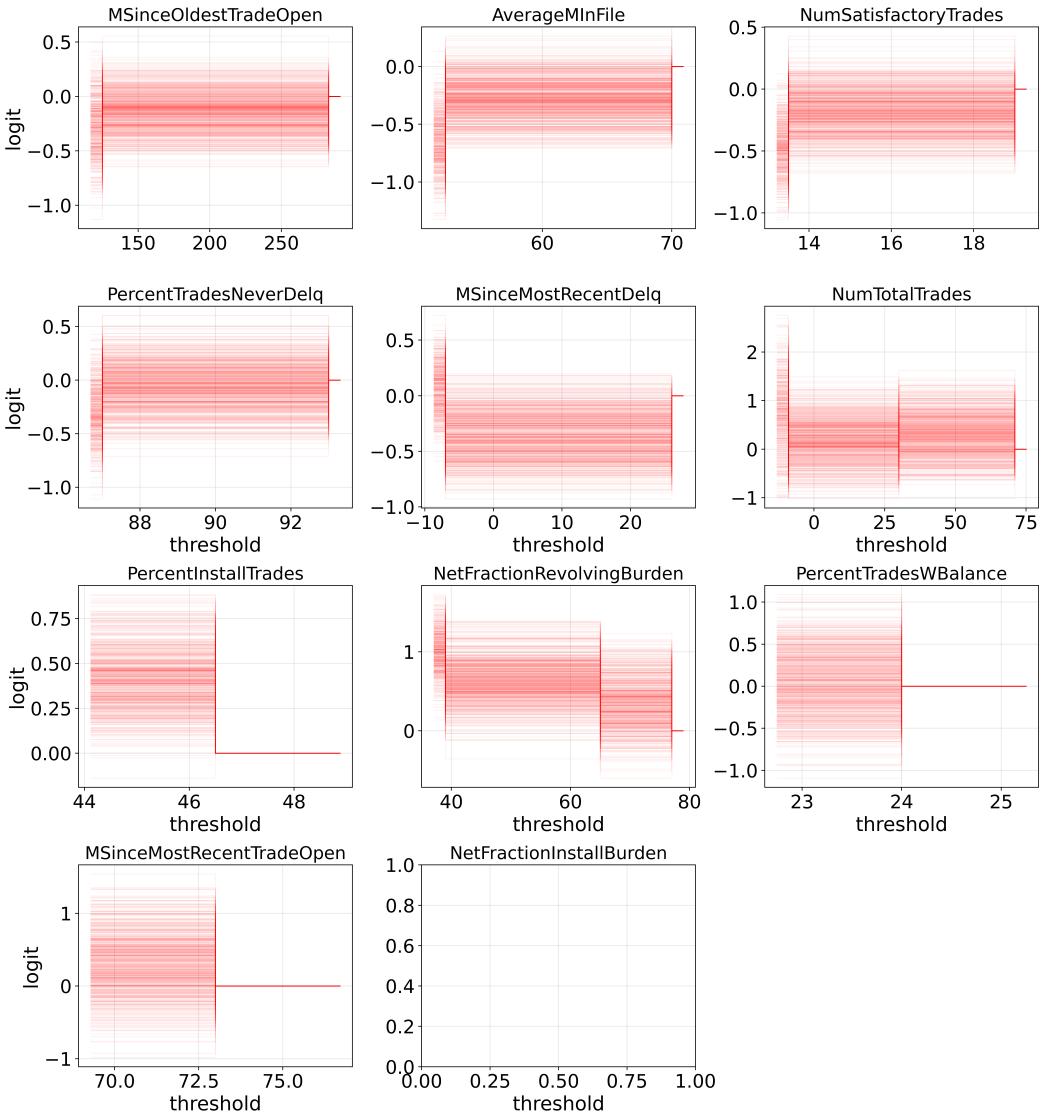


Figure 16: Illustration of GAMs output by the ellipsoid method in [36] for 11 features on the FICO dataset. For this visualization, we set the Rashomon upper bound to match that of GRAM in Figure 15. We chose the optimal support set found by GRAM for this ellipsoid - it is of size 28.

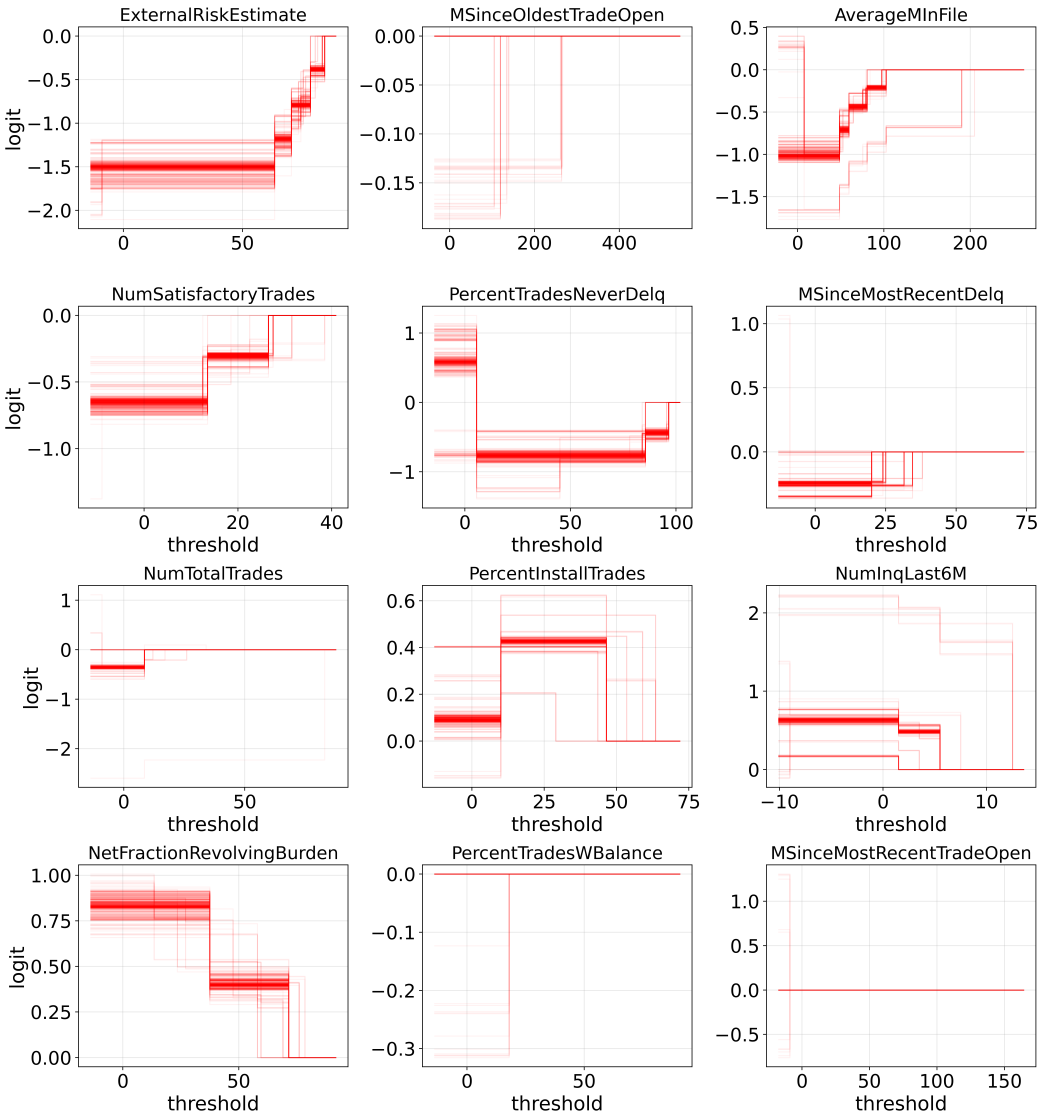


Figure 17: Illustration of GAMs output by the swapping method in [24] for 11 features on the FICO dataset. For this visualization, we prune all models outside the Rashomon upper bound set by GRAM in Figure 15.

F.3 COMPAS Dataset

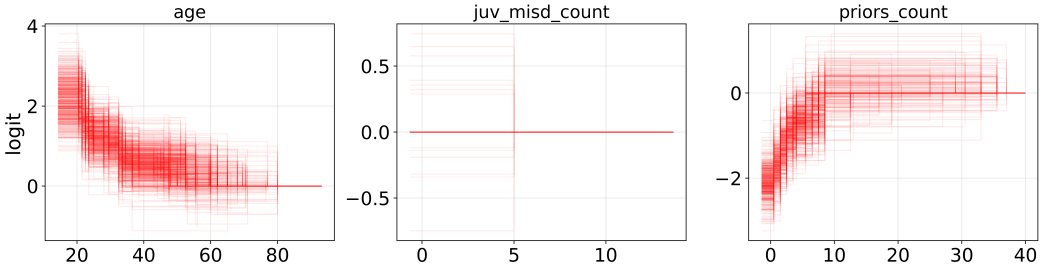


Figure 18: Illustration of GAMs output by GRAM on the COMPAS dataset (3 shape functions). The average support set size is ≈ 11. We set $\epsilon = 0.025, \sigma^2 = 0.05, p_{feat} = 1e-4$.

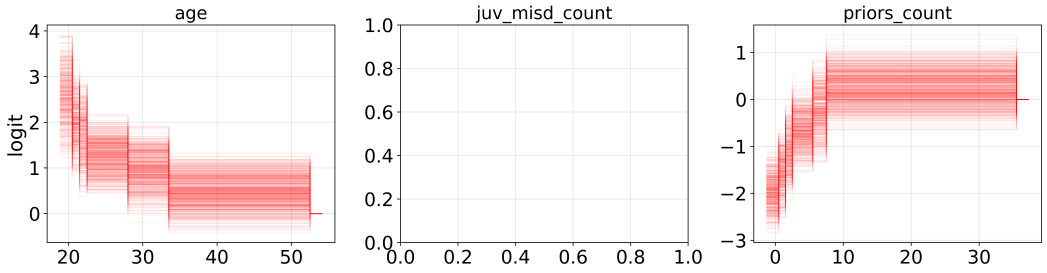


Figure 19: Illustration of GAMs output by the ellipsoid method in [36] for 3 features on the COMPAS dataset. For this visualization, we set the Rashomon upper bound to match that of GRAM in Figure 18. We chose the optimal support set found by GRAM for this ellipsoid - it is of size 12.

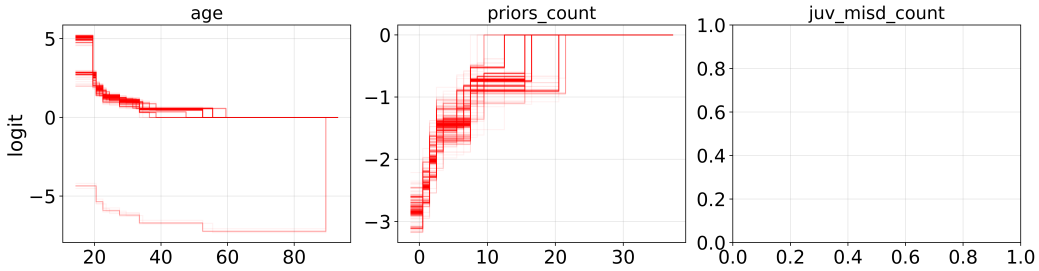


Figure 20: Illustration of GAMs output by the swapping method in [24] for 3 features on the COMPAS dataset. For this visualization, we set the Rashomon upper bound to match that of GRAM in Figure 18.

G Description of Datasets and Computing Resources

We perform experiments with 10 datasets in this paper. We describe them in Table 17. Note that these are the non-binarized versions of the datasets - the binary versions often have 1000s of features.

Dataset	Description	Source
NIJ Recidivism	Predict 1/2/3-year recidivism for 16264 individuals released from Georgia parole; demographic, supervision, and prior-offense features.	[29]
Adult	Predict whether U.S. adult income exceeds $50K from 1994 Census data (48842 examples, 14 demographic/employment features).	[4]
COMPAS	Broward County, FL pretrial defendants; task is to predict 2-year recidivism. Widely used in algorithmic-fairness studies. (6907 rows, 7 credit and demographic related features)	[3]
Covertype	Predict forest cover type for 30×30m cells in Roosevelt Nat'l Forest from cartographic variables (∼581K rows in the original dataset, but we subsample to 10k rows. 54 features)	[6]
FICO	Home Equity Line of Credit (HELOC) applications; predict serious delinquency. Released for the FICO Explainable ML Challenge. 10459 examples, 23 features.	[15]
Heart	Cleveland Clinic heart-disease dataset (303 patients, 13 clinical features); binary diagnosis of coronary artery disease.	[9]
MGH	Massachusetts General Hospital dataset of physiological parameters of patients who spent time in ICU. The aim is to predict whether they died in ICU. It has 8702 examples and 28 features. This dataset is not public.	
Spambase	4601 emails labeled spam/non-spam with 57 word- and character-frequency features.	[19]
Student	Portuguese secondary-school students; predict final grade in math/Portuguese from demographic and school-related features. 649 examples, 29 features.	[8]
Wine	Red and white *vinho verde* samples; predict sensory quality score from 11 physicochemical measurements. It has 6497 examples.	[1]

Table 17: Datasets used in our experiments.

We ran all experiments on a standard cluster, with an Intel Xeon 6530P CPU with 150 GB of allocated RAM. All experiments were run for up to 24 hours if needed, though most took only a few hours to run (including the time for 5 trials).

H Description of the Swapping Method Baseline

We compare against a swapping-based baseline adapted from the sparse diverse-pool procedure in FasterRisk [24]. Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ with p total bin-boundary features across all features (as defined in Section 2), we first fit a sparse GAM by minimizing the objective in Eq. 3 and obtain an anchor model $(\boldsymbol{\omega}^*, \omega_0^*)$. Let

$$S^* = \mathrm{supp}(\boldsymbol{\omega}^*) \ \subseteq \ [p]$$

denote the active support of the anchor, and let $D = |S^*|$ be the support size.

Gradient-guided candidate generation. Starting from the anchor support, the method performs a beam search over nearby supports obtained through one-for-one feature swaps. At a given search state with support S, the method considers removing one active bin boundary $j \in S$ and adding one inactive bin boundary $\ell \notin S$, forming

$$S' = (S \setminus \{j\}) \cup \{\ell\}.$$

Rather than exhaustively evaluating all $D \times |S^c|$ possible swaps, inactive features are ranked by the magnitude of their coordinate gradient after temporarily zeroing out j. At most $Z = \min(|S^c|, 5)$ replacement features are evaluated for each removed feature j, dramatically reducing the branching factor while targeting the most promising bin boundaries.

Re-optimization and loss-gating. For each candidate support S', the coefficient of the newly introduced feature ℓ is re-optimized via 10 steps of coordinate descent, keeping all other coefficients fixed, followed by optional fine-tuning on the current support. A candidate $(\boldsymbol{\omega}', \omega_0')$ is retained in the pool only if its relative increase in the regularized logistic objective satisfies the gap criterion

$$\frac{\mathcal{L}(\boldsymbol{\omega}', \omega_0', \mathcal{D}) - \mathcal{L}(\boldsymbol{\omega}^*, \omega_0^*, \mathcal{D})}{\mathcal{L}(\boldsymbol{\omega}^*, \omega_0^*, \mathcal{D})} \ \leq \ \epsilon_{\mathrm{swap}},$$

where $\mathcal{L}$ is the full regularized objective from Eq. 3 and $\epsilon_{\mathrm{swap}} = \gamma - 1$ for multiplicative swap tolerance $\gamma \geq 1$. This in-loop gating avoids expanding unpromising branches before they enter the beam.

Beam truncation and duplicate removal. After all candidate swaps at a given depth are evaluated, duplicate support patterns are removed. The surviving candidates are ranked by $\mathcal{L}(\boldsymbol{\omega}, \omega_0, \mathcal{D})$, and the best B are retained as the next beam $\mathcal{B}_t$. At the final depth $t = K$, beam truncation is skipped and all accepted candidates are kept. This process repeats for up to K swap rounds; the effective number of rounds is capped at $\min(K, Z, D)$ to respect the support size and the number of available inactive features.

Final Rashomon filter. After beam search, all discovered candidates $\mathcal{C} = \bigcup_{t=0}^K \mathcal{B}_t$ are converted from the cumulative (step-function) parameterization back to the binned GAM representation. We then apply the Rashomon criterion using $\mathcal{L}_c$, the candidate-evaluation loss used in the experiments, retaining candidates satisfying

$$\mathcal{L}_c(\boldsymbol{\omega}, \omega_0, \mathcal{D}) \ \leq \ \epsilon,$$

where B is the Rashomon loss threshold used throughout the experiments. The resulting set $\widehat{\mathcal{R}}_{\mathrm{swap}}$ is the swapping-based approximation of the Rashomon set.

Algorithm 3 BeamSearchSwap for Approximate Rashomon-Set Construction

Require: Dataset $\mathcal{D}$, anchor model $(\boldsymbol{\omega}^*, \omega_0^*)$, max swap rounds K, beam width B, at most 5 gradient-filtered replacements per active feature, gap tolerance $\epsilon_{\text{swap}} = \gamma - 1$ for $\gamma \geq 1$, Rashomon threshold ϵ

Ensure: Swapping-based Rashomon approximation $\widehat{\mathcal{R}}_{\text{swap}}$

$\mathcal{B}_0 \leftarrow \{(\boldsymbol{\omega}^*, \omega_0^*)\}$ ▷ Initialize beam with anchor
$\mathcal{C} \leftarrow \{(\boldsymbol{\omega}^*, \omega_0^*)\}$ ▷ All discovered candidates
for $t = 1$ **to** K **do**
  $\mathcal{N} \leftarrow \emptyset$ ▷ Candidate pool for this round
  for each $(\boldsymbol{\omega}, \omega_0) \in \mathcal{B}_{t-1}$ **do**
    $S \leftarrow \text{supp}(\boldsymbol{\omega})$
    for each $j \in S$ **do**
      Temporarily zero out $\boldsymbol{\omega}_j$
      Rank inactive coordinates $\ell \notin S$ by magnitude of coordinate gradient
      $\mathcal{L}_{\text{top}} \leftarrow$ top-$\min(|S^c|, 5)$ inactive features ▷ Gradient filter
      for each $\ell \in \mathcal{L}_{\text{top}}$ **do**
        $S' \leftarrow (S \setminus \{j\}) \cup \{\ell\}$
        $(\boldsymbol{\omega}', \omega_0') \leftarrow \text{CoordDescent}(\mathcal{D},\ S',\ \boldsymbol{\omega},\ \text{steps} = 10)$
        if $\dfrac{\mathcal{L}(\boldsymbol{\omega}', \omega_0', \mathcal{D}) - \mathcal{L}(\boldsymbol{\omega}^*, \omega_0^*, \mathcal{D})}{\mathcal{L}(\boldsymbol{\omega}^*, \omega_0^*, \mathcal{D})} \leq \epsilon_{\text{swap}}$ **then**
          $\mathcal{N} \leftarrow \mathcal{N} \cup \{(\boldsymbol{\omega}', \omega_0')\}$
        end if
      end for
    end for
  end for
  Remove duplicate support patterns from $\mathcal{N}$
  if $t < K$ **then**
    $\mathcal{B}_t \leftarrow$ top-B candidates in $\mathcal{N}$ by $\mathcal{L}(\boldsymbol{\omega}, \omega_0, \mathcal{D})$
  else
    $\mathcal{B}_t \leftarrow \mathcal{N}$ ▷ Keep all at final depth
  end if
  $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{B}_t$
end for
Convert all $(\boldsymbol{\omega}, \omega_0) \in \mathcal{C}$ to binned GAM representation
$\widehat{\mathcal{R}}_{\text{swap}} \leftarrow \{(\boldsymbol{\omega}, \omega_0) \in \mathcal{C} : \mathcal{L}_c(\boldsymbol{\omega}, \omega_0, \mathcal{D}) \leq \epsilon\}$
return $\widehat{\mathcal{R}}_{\text{swap}}$

I Detailed Version of Table 1

Note that for all tables below − means that the method did not return any GAMs in the Rashomon bound in that configuration.

Table 18: Baseline comparison on the Support Jaccard diversity metric. Rows are (dataset, method) pairs; columns are ε. Each entry is the mean $\pm$ standard error across trials (sample standard deviation divided by $\sqrt{n}$). The standard error is omitted when only one trial has a score. Higher is more diverse; the per-(dataset, ε) maximum is **bold**.

Dataset	Method	ε=0.005	ε=0.01	ε=0.015	ε=0.025
NIJ Recidivism	Bootstrap LogReg	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
	GRAM	**0.240 ± 0.013**	**0.368 ± 0.020**	**0.408 ± 0.023**	**0.492 ± 0.008**
	Ellipsoid (Reject.)	–	–	0.344 ± 0.054	0.453 ± 0.003
	Ellipsoid (Unif.)	–	–	0.348 ± 0.019	0.451 ± 0.004
	Single Ellipsoid	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000

Continued on next page

Table 18 — continued from previous page

Dataset	Method	ε=0.005	ε=0.01	ε=0.015	ε=0.025
	Swapping	–	–	–	–
Adult	Bootstrap LogReg	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
	GRAM	$\mathbf{0.084 \pm 0.037}$	$\mathbf{0.131 \pm 0.030}$	$\mathbf{0.256 \pm 0.075}$	$\mathbf{0.500 \pm 0.077}$
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
	Swapping	–	–	–	–
COMPAS	Bootstrap LogReg	0.002 ± 0.000	0.002 ± 0.000	0.002 ± 0.000	0.002 ± 0.000
	GRAM	$\mathbf{0.597 \pm 0.077}$	$\mathbf{0.757 \pm 0.027}$	$\mathbf{0.810 \pm 0.009}$	0.837 ± 0.002
	Ellipsoid (Reject.)	–	–	–	**0.879**
	Ellipsoid (Unif.)	–	–	–	0.722 ± 0.056
	Single Ellipsoid	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
	Swapping	–	–	–	–
Covertype	Bootstrap LogReg	–	–	–	0.000
	GRAM	$\mathbf{0.342 \pm 0.086}$	$\mathbf{0.541 \pm 0.096}$	$\mathbf{0.577 \pm 0.109}$	$\mathbf{0.818 \pm 0.014}$
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
	Swapping	–	–	0.161 ± 0.000	0.161 ± 0.000
FICO	Bootstrap LogReg	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
	GRAM	$\mathbf{0.626 \pm 0.053}$	$\mathbf{0.815 \pm 0.014}$	$\mathbf{0.853 \pm 0.008}$	$\mathbf{0.891 \pm 0.003}$
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
	Swapping	–	–	0.157	0.157 ± 0.000
Heart	Bootstrap LogReg	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
	GRAM	0.007 ± 0.007	0.057 ± 0.049	0.119 ± 0.066	0.224 ± 0.080
	Ellipsoid (Reject.)	**0.344**	**0.401**	**0.351**	$\mathbf{0.320 \pm 0.049}$
	Ellipsoid (Unif.)	0.340	0.136 ± 0.136	0.317	0.243 ± 0.122
	Single Ellipsoid	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
	Swapping	–	–	–	–
MGH	Bootstrap LogReg	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
	GRAM	$\mathbf{0.143 \pm 0.054}$	$\mathbf{0.291 \pm 0.112}$	$\mathbf{0.454 \pm 0.075}$	$\mathbf{0.730 \pm 0.029}$
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
	Swapping	–	–	–	–
Spambase	Bootstrap LogReg	–	–	–	–
	GRAM	$\mathbf{0.044 \pm 0.018}$	$\mathbf{0.087 \pm 0.018}$	$\mathbf{0.093 \pm 0.016}$	$\mathbf{0.193 \pm 0.035}$
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
	Swapping	–	–	–	–
Student	Bootstrap LogReg	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
	GRAM	$\mathbf{0.173 \pm 0.074}$	$\mathbf{0.248 \pm 0.095}$	$\mathbf{0.466 \pm 0.020}$	$\mathbf{0.545 \pm 0.010}$
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
	Swapping	–	–	–	–
Wine	Bootstrap LogReg	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
	GRAM	$\mathbf{0.098 \pm 0.035}$	$\mathbf{0.144 \pm 0.057}$	$\mathbf{0.238 \pm 0.050}$	$\mathbf{0.424 \pm 0.041}$

Continued on next page

Table 18 — continued from previous page

Dataset	Method	ε=0.005	ε=0.01	ε=0.015	ε=0.025
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
	Swapping	–	–	–	–

Table 19: Baseline comparison on the Weight L1 diversity metric. Rows are (dataset, method) pairs; columns are ε. Each entry is the mean ± standard error across trials (sample standard deviation divided by $\sqrt{n}$). The standard error is omitted when only one trial has a score. Higher is more diverse; the per-(dataset, ε) maximum is **bold**.

Dataset	Method	ε=0.005	ε=0.01	ε=0.015	ε=0.025
NIJ Recidivism	Bootstrap LogReg	0.061 ± 0.000	0.061 ± 0.000	0.061 ± 0.000	0.061 ± 0.000
	GRAM	0.127 ± 0.001	0.161 ± 0.001	0.182 ± 0.000	0.211 ± 0.000
	Ellipsoid (Reject.)	–	–	0.160 ± 0.018	0.192 ± 0.003
	Ellipsoid (Unif.)	–	0.100	0.171 ± 0.006	0.196 ± 0.002
	Single Ellipsoid	**0.135 ± 0.000**	**0.170 ± 0.000**	**0.190 ± 0.000**	**0.214 ± 0.000**
	Swapping	–	–	–	–
Adult	Bootstrap LogReg	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
	GRAM	0.001 ± 0.000	0.001 ± 0.000	0.002 ± 0.000	0.002 ± 0.000
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	**0.001 ± 0.000**	**0.002 ± 0.000**	**0.002 ± 0.000**	**0.002 ± 0.000**
	Swapping	–	–	–	–
COMPAS	Bootstrap LogReg	0.076 ± 0.001	0.078 ± 0.000	0.078 ± 0.000	0.078 ± 0.000
	GRAM	0.095 ± 0.001	**0.110 ± 0.001**	0.115 ± 0.001	0.117 ± 0.001
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	**0.099 ± 0.000**	0.110 ± 0.004	**0.117 ± 0.004**	**0.123 ± 0.004**
	Swapping	–	–	–	–
Covertype	Bootstrap LogReg	–	–	–	0.001 ± 0.000
	GRAM	**0.002 ± 0.000**	**0.002 ± 0.000**	**0.003 ± 0.000**	0.003 ± 0.000
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	0.002 ± 0.000	0.002 ± 0.000	0.003 ± 0.000	0.003 ± 0.000
	Swapping	–	–	–	**0.004 ± 0.000**
FICO	Bootstrap LogReg	**0.006 ± 0.000**	0.006 ± 0.000	0.006 ± 0.000	0.006 ± 0.000
	GRAM	0.006 ± 0.000	**0.007 ± 0.001**	**0.010 ± 0.002**	**0.013 ± 0.002**
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	0.005 ± 0.000	0.006 ± 0.000	0.007 ± 0.000	0.008 ± 0.000
	Swapping	–	–	–	–
Heart	Bootstrap LogReg	–	–	0.043 ± 0.001	0.053 ± 0.000
	GRAM	0.025 ± 0.000	0.035 ± 0.001	0.045 ± 0.000	0.059 ± 0.000
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	**0.064**
	Single Ellipsoid	**0.027 ± 0.000**	**0.038 ± 0.000**	**0.046 ± 0.000**	0.059 ± 0.000
	Swapping	–	–	–	–
MGH	Bootstrap LogReg	0.001 ± 0.000	0.001 ± 0.000	0.001 ± 0.000	0.001 ± 0.000
	GRAM	0.002 ± 0.000	0.003 ± 0.000	**0.003 ± 0.000**	**0.004 ± 0.000**

Continued on next page

Table 19 — continued from previous page

Dataset	Method	ε=0.005	ε=0.01	ε=0.015	ε=0.025
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	**0.002 ± 0.000**	**0.003 ± 0.000**	0.003 ± 0.000	0.004 ± 0.000
	Swapping	–	–	–	–
Spambase	Bootstrap LogReg	–	–	–	–
	GRAM	**0.001 ± 0.000**	**0.001 ± 0.000**	**0.001 ± 0.000**	**0.001 ± 0.000**
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	**0.001 ± 0.000**	**0.001 ± 0.000**	**0.001 ± 0.000**	**0.001 ± 0.000**
	Swapping	–	–	–	–
Student	Bootstrap LogReg	–	0.036 ± 0.000	0.042 ± 0.000	0.049 ± 0.000
	GRAM	0.035 ± 0.001	**0.052 ± 0.001**	**0.063 ± 0.000**	**0.077 ± 0.000**
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	**0.036 ± 0.000**	0.049 ± 0.000	0.060 ± 0.000	0.075 ± 0.000
	Swapping	–	–	–	–
Wine	Bootstrap LogReg	0.033 ± 0.000	0.033 ± 0.000	0.033 ± 0.000	0.033 ± 0.000
	GRAM	0.037 ± 0.000	0.052 ± 0.000	0.063 ± 0.000	**0.079 ± 0.000**
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	**0.039 ± 0.000**	**0.053 ± 0.000**	**0.064 ± 0.000**	0.078 ± 0.000
	Swapping	–	–	–	–

Table 20: Baseline comparison on the Prediction Hamming diversity metric. Rows are (dataset, method) pairs; columns are ε. Each entry is the mean ± standard error across trials (sample standard deviation divided by $\sqrt{n}$). The standard error is omitted when only one trial has a score. Higher is more diverse; the per-(dataset, ε) maximum is **bold**.

Dataset	Method	ε=0.005	ε=0.01	ε=0.015	ε=0.025
NIJ Recidivism	Bootstrap LogReg	0.014 ± 0.000	0.014 ± 0.000	0.014 ± 0.000	0.014 ± 0.000
	GRAM	**0.042 ± 0.001**	**0.068 ± 0.001**	0.082 ± 0.001	**0.107 ± 0.002**
	Ellipsoid (Reject.)	–	0.066	**0.096 ± 0.008**	0.102 ± 0.001
	Ellipsoid (Unif.)	–	–	0.078 ± 0.007	0.101 ± 0.001
	Single Ellipsoid	0.041 ± 0.001	0.062 ± 0.001	0.077 ± 0.001	0.101 ± 0.001
	Swapping	–	–	–	–
Adult	Bootstrap LogReg	**0.020 ± 0.000**	0.020 ± 0.000	0.020 ± 0.000	0.020 ± 0.000
	GRAM	0.017 ± 0.000	0.027 ± 0.002	0.032 ± 0.002	0.042 ± 0.002
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	0.019 ± 0.000	**0.027 ± 0.000**	**0.033 ± 0.000**	**0.043 ± 0.000**
	Swapping	–	–	–	–
COMPAS	Bootstrap LogReg	0.056 ± 0.000	0.056 ± 0.000	0.056 ± 0.000	0.056 ± 0.000
	GRAM	**0.069 ± 0.007**	**0.099 ± 0.007**	**0.124 ± 0.005**	**0.161 ± 0.005**
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	0.144 ± 0.037
	Single Ellipsoid	0.063 ± 0.003	0.083 ± 0.004	0.098 ± 0.004	0.122 ± 0.004
	Swapping	–	–	–	–
Covertype	Bootstrap LogReg	–	–	–	–
	GRAM	**0.024 ± 0.001**	**0.033 ± 0.001**	**0.041 ± 0.003**	**0.057 ± 0.003**

Continued on next page

Table 20 — continued from previous page

Dataset	Method	ε=0.005	ε=0.01	ε=0.015	ε=0.025
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	0.024 ± 0.001	0.033 ± 0.001	0.041 ± 0.001	0.053 ± 0.001
	Swapping	–	–	–	–
FICO	Bootstrap LogReg	$\mathbf{0.045 \pm 0.000}$	0.045 ± 0.000	0.045 ± 0.000	0.045 ± 0.000
	GRAM	0.040 ± 0.001	$\mathbf{0.054 \pm 0.001}$	$\mathbf{0.066 \pm 0.001}$	0.087 ± 0.002
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	0.035 ± 0.000	0.052 ± 0.000	0.066 ± 0.000	$\mathbf{0.088 \pm 0.001}$
	Swapping	–	–	–	–
Heart	Bootstrap LogReg	0.061 ± 0.000	0.061 ± 0.000	0.061 ± 0.000	0.061 ± 0.000
	GRAM	0.031 ± 0.005	0.043 ± 0.005	0.053 ± 0.005	0.064 ± 0.004
	Ellipsoid (Reject.)	$\mathbf{0.063 \pm 0.002}$	$\mathbf{0.071 \pm 0.004}$	0.070 ± 0.008	0.072 ± 0.005
	Ellipsoid (Unif.)	0.041 ± 0.024	0.067 ± 0.004	**0.079**	$\mathbf{0.076 \pm 0.010}$
	Single Ellipsoid	0.025 ± 0.003	0.035 ± 0.003	0.042 ± 0.003	0.053 ± 0.002
	Swapping	–	–	–	–
MGH	Bootstrap LogReg	$\mathbf{0.049 \pm 0.000}$	0.049 ± 0.000	0.049 ± 0.000	0.049 ± 0.000
	GRAM	0.040 ± 0.002	0.054 ± 0.003	0.067 ± 0.003	0.083 ± 0.007
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	0.041 ± 0.001	$\mathbf{0.058 \pm 0.001}$	$\mathbf{0.070 \pm 0.001}$	$\mathbf{0.090 \pm 0.001}$
	Swapping	–	–	–	–
Spambase	Bootstrap LogReg	–	–	–	–
	GRAM	0.003 ± 0.000	0.004 ± 0.000	0.005 ± 0.000	0.007 ± 0.000
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	$\mathbf{0.003 \pm 0.000}$	$\mathbf{0.005 \pm 0.000}$	$\mathbf{0.006 \pm 0.000}$	$\mathbf{0.007 \pm 0.000}$
	Swapping	–	–	–	–
Student	Bootstrap LogReg	$\mathbf{0.056 \pm 0.000}$	0.056 ± 0.000	0.056 ± 0.000	0.056 ± 0.000
	GRAM	0.052 ± 0.011	$\mathbf{0.061 \pm 0.007}$	$\mathbf{0.072 \pm 0.008}$	0.084 ± 0.010
	Ellipsoid (Reject.)	–	–	–	0.076 ± 0.020
	Ellipsoid (Unif.)	–	–	–	**0.100**
	Single Ellipsoid	0.031 ± 0.003	0.044 ± 0.004	0.054 ± 0.004	0.070 ± 0.004
	Swapping	–	–	–	–
Wine	Bootstrap LogReg	0.031 ± 0.000	0.032 ± 0.000	0.032 ± 0.000	0.032 ± 0.000
	GRAM	$\mathbf{0.035 \pm 0.001}$	0.048 ± 0.001	0.059 ± 0.001	0.075 ± 0.001
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	0.034 ± 0.001	$\mathbf{0.048 \pm 0.001}$	$\mathbf{0.059 \pm 0.001}$	$\mathbf{0.076 \pm 0.001}$
	Swapping	–	–	–	–

Table 21: Baseline comparison on the Monotonicity entropy diversity metric. Rows are (dataset, method) pairs; columns are ε. Each entry is the mean $\pm$ standard error across trials (sample standard deviation divided by $\sqrt{n}$). The standard error is omitted when only one trial has a score. Higher is more diverse; the per-(dataset, ε) maximum is **bold**.

Dataset	Method	ε=0.005	ε=0.01	ε=0.015	ε=0.025
NIJ Recidivism	Bootstrap LogReg	$\mathbf{0.302 \pm 0.001}$	$\mathbf{0.302 \pm 0.001}$	0.302 ± 0.001	0.302 ± 0.001
	GRAM	0.183 ± 0.005	0.252 ± 0.005	$\mathbf{0.315 \pm 0.004}$	0.397 ± 0.002

Continued on next page

Table 21 — continued from previous page

Dataset	Method	ε=0.005	ε=0.01	ε=0.015	ε=0.025
	Ellipsoid (Reject.)	–	0.000	0.296 ± 0.111	0.397 ± 0.016
	Ellipsoid (Unif.)	–	0.000 ± 0.000	0.246 ± 0.028	**0.399 ± 0.018**
	Single Ellipsoid	0.050 ± 0.050	0.093 ± 0.057	0.160 ± 0.048	0.262 ± 0.027
	Swapping	–	–	–	–
Adult	Bootstrap LogReg	0.037 ± 0.002	0.047 ± 0.010	0.056 ± 0.012	0.066 ± 0.011
	GRAM	**0.217 ± 0.059**	**0.214 ± 0.009**	**0.269 ± 0.025**	**0.315 ± 0.036**
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	0.067 ± 0.028	0.098 ± 0.028	0.094 ± 0.035	0.110 ± 0.027
	Swapping	–	0.029	0.029 ± 0.000	0.029 ± 0.000
COMPAS	Bootstrap LogReg	0.042 ± 0.001	0.042 ± 0.001	0.042 ± 0.001	0.042 ± 0.001
	GRAM	**0.617 ± 0.025**	**0.655 ± 0.024**	**0.657 ± 0.032**	**0.605 ± 0.043**
	Ellipsoid (Reject.)	–	–	–	0.000
	Ellipsoid (Unif.)	–	–	–	0.000 ± 0.000
	Single Ellipsoid	0.263 ± 0.045	0.231 ± 0.049	0.199 ± 0.046	0.160 ± 0.045
	Swapping	–	–	–	–
Covertype	Bootstrap LogReg	–	–	–	–
	GRAM	**0.305 ± 0.026**	**0.333 ± 0.021**	**0.322 ± 0.029**	**0.314 ± 0.019**
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	0.239 ± 0.025	0.262 ± 0.024	0.203 ± 0.042	0.203 ± 0.041
	Swapping	–	0.131	0.131 ± 0.000	0.131 ± 0.000
FICO	Bootstrap LogReg	0.041 ± 0.002	0.041 ± 0.002	0.041 ± 0.001	0.027 ± 0.001
	GRAM	**0.408 ± 0.054**	**0.490 ± 0.023**	**0.526 ± 0.016**	**0.544 ± 0.021**
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	0.259 ± 0.052	0.308 ± 0.048	0.335 ± 0.045	0.303 ± 0.029
	Swapping	–	–	–	0.032 ± 0.000
Heart	Bootstrap LogReg	**0.084 ± 0.021**	**0.088 ± 0.019**	**0.092 ± 0.015**	**0.098 ± 0.009**
	GRAM	0.002 ± 0.002	0.008 ± 0.007	0.012 ± 0.007	0.027 ± 0.012
	Ellipsoid (Reject.)	–	0.000 ± 0.000	0.000	0.000
	Ellipsoid (Unif.)	–	–	0.000	0.000
	Single Ellipsoid	0.000 ± 0.000	0.000 ± 0.000	0.007 ± 0.004	0.039 ± 0.014
	Swapping	–	–	–	–
MGH	Bootstrap LogReg	0.053 ± 0.001	0.053 ± 0.001	0.053 ± 0.001	0.053 ± 0.001
	GRAM	**0.253 ± 0.023**	**0.295 ± 0.024**	**0.333 ± 0.014**	**0.372 ± 0.017**
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	0.122 ± 0.028	0.162 ± 0.029	0.183 ± 0.025	0.220 ± 0.026
	Swapping	–	–	–	–
Spambase	Bootstrap LogReg	–	–	–	–
	GRAM	**0.247 ± 0.016**	**0.309 ± 0.011**	**0.355 ± 0.018**	**0.422 ± 0.023**
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	0.239 ± 0.012	0.299 ± 0.015	0.333 ± 0.016	0.369 ± 0.015
	Swapping	–	–	–	–
Student	Bootstrap LogReg	**0.147 ± 0.024**	**0.119 ± 0.025**	**0.137 ± 0.018**	0.162 ± 0.005
	GRAM	0.010 ± 0.010	0.034 ± 0.019	0.043 ± 0.009	0.067 ± 0.016
	Ellipsoid (Reject.)	–	–	–	0.000
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	0.000 ± 0.000	0.046 ± 0.025	0.089 ± 0.043	**0.169 ± 0.061**

Continued on next page

Table 21 — continued from previous page

Dataset	Method	ε=0.005	ε=0.01	ε=0.015	ε=0.025
	Swapping	–	–	–	–
Wine	Bootstrap LogReg	**0.171 ± 0.002**	**0.171 ± 0.002**	**0.171 ± 0.002**	0.171 ± 0.002
	GRAM	0.042 ± 0.010	0.081 ± 0.007	0.137 ± 0.010	**0.237 ± 0.010**
	Ellipsoid (Reject.)	–	–	–	–
	Ellipsoid (Unif.)	–	–	–	–
	Single Ellipsoid	0.036 ± 0.016	0.089 ± 0.021	0.143 ± 0.020	0.231 ± 0.018
	Swapping	–	–	–	–

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We claim that our method produces more diverse solutions across different tested metrics - we provide exhaustive experimentation to back these claims.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The conclusion discusses areas for future work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.

- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper provides a proof for the main Theorem (3.1) in the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: For all experiments, we detail the different hyperparameters set and the number of iterations run for the algorithm. We also provide code.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is provided. All datasets except the MGH dataset are public.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: For all experiments, we detail the different hyperparameters set and the number of iterations run for the algorithm.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Standard error over 5 trials is reported for all experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section G of the paper describes basic compute resources used.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work fully complies with the guidelines.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We provide a brief broader impacts section discussing this point.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: This work poses no substantial risk for misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: This work relies on public and open access assets through code packages and published papers, which are properly cited and used in compliance with their licenses.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The primary asset introduced in this work is code, which is documented with a README file.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: There is no human subject experiment done in this work.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

Justification: There is no human subject experiment done in this work.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs were only used for minor editing/rephrasing and basic code assistance.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.