More of Less: A Rashomon Algorithm for Sparse Model Sets

Anonymous authors
Paper under double-blind review

Abstract

The current paradigm of machine learning consists in finding a single best model to deliver predictions and, if possible, interpretations for a specific problem. This paradigm has however been strongly challenged in recent years through the study of the *Rashomon Effect* which was coined initially by Leo Breiman. This phenomenon occurs when there exist many good predictive models for a given dataset/problem, with considerable practical implications in terms of interpretation, usability, variable importance, replicability and many others. The set of models (within a specific class of functions) which respect this definition is referred to as the *Rashomon set* and an important amount of recent work has been focused on ways of finding these sets as well as studying their properties. Developed in parallel to current research on the Rashomon Effect and motivated by sparse latent representations for high-dimensional problems, we present a heuristic procedure that aims to find sets of sparse models with good predictive power through a greedy forward-search that explores the low-dimensional variable space. Throughout this algorithm, good low-dimensional models identified from the previous steps are used to build models with more variables in the following steps. While preserving almost-equal performance with respect to a single reference model in a given class (i.e. a Rashomon set), the sparse model sets from this algorithm include diverse models which can be combined into networks that deliver additional layers of interpretation and new insights into how variable combinations can explain the Rashomon Effect.

1 Introduction

The purpose of any machine learning algorithm is to deliver precise predictions with respect to a response (or responses) of interest given a set of variables (predictors), whether dealing with classification or more general regression problems. The current paradigm to achieve such predictions is commonly based on a *single* best model that has been chosen, parametrized and fine-tuned to address a specific problem or dataset. There are many reasons why such a paradigm is the dominating one, among which (i) its reliance on the existence of a unique optimal representation of the data generation process as well as (ii) the practical implications of relying on a single set of parameters and variables that can be used, and eventually interpreted, for predictions on similar problems. There are however various limitations with this paradigm, starting from areas of research and practice where there is a need for flexibility in the variables used. Indeed, from genomics (see e.g. Xiong et al., 2001) to online prediction (see e.g. Carmona-Cejudo et al., 2011), there are many tasks where a multitude of subsets of variables are useful, such as (i) in medical studies where machines collect different measurements (variables) for a specific problem (see e.g. Draghici et al., 2006); (ii) for online search algorithms where every subject provides different variables (according to their preferences or willingness to disclose information) to determine suggestions or matches (see e.g. Vaughan & Chen, 2015) ; (iii) in signal processing and pattern recognition where signals and images are collected at different resolutions and therefore a single representation may not be flexible enough to adapt to different signal and image features (see e.g. Elad & Yavneh, 2009; Wang et al., 2018). Intuitively, there can indeed be circumstances (e.g. medical diagnostics) where some variables are more costly to measure and practitioners would therefore prefer to have alternative good models with more "accessible" variables. In addition, interpretability of phenomena can be greatly enhanced and stabilized when considering a set of models as highlighted by the multimodel inference[1] and model selection uncertainty literature in areas such as sociology and, especially, ecology (see

[1]Not to be confused with multimodal inference.

e.g. Burnham & Anderson, 2004; Anderson & Burnham, 2004; Harrison et al., 2018; Caruana et al., 2004, for an overview). This is of particular relevance, for example, within the Predictability-Computability-Stability (PCS) framework of Yu & Kumbier (2020).

The existence of multiple models that fit a specific problem/dataset similarly well was already posited in Breiman (2001b), albeit for more fundamental reasons coming from the uncertainty and noise affecting data. Indeed, as a result of this uncertainty, many approximately-equally-good models can exist for a specific problem/dataset giving rise to the so-called *Rashomon Effect* (Breiman, 2001b). While being commonly observed in practice, this phenomenon returned to be an area of fundamental research only in recent years starting from the work of Fisher et al. (2019) where they develop a new metric of variable importance using multiple good models, and followed by a growing focus on this topic in Semenova et al. (2022); Xin et al. (2022); Zhong et al. (2024); Liu et al. (2022); Qinyu Zhu et al. (2023); Semenova et al. (2024); Kissel & Mentch (2024) where they develop methods and measures to select and evaluate sets of good models belonging to different model classes. The need for a paradigm shift in machine learning as a result of this effect was underlined more recently in Rudin et al. (2024) where the authors also highlighted the numerous advantages that can be obtained when finding sets of models, as opposed to a single one, especially for high-stakes decision making. In particular these advantages include, but are not limited to: (i) the possibility for users to determine their preferred representation of a problem without losing predictive power; (ii) stable variable importance information; and (iii) novel insights for model interpretability and prediction uncertainty. As a consequence, there is a need to develop procedures to find these sets of models which, intuitively, are called *Rashomon sets* (Fisher et al., 2019; Semenova et al., 2022). The push to develop such procedures is very recent and, currently, exist for decision trees (Xin et al., 2022), generalized additive models (Zhong et al., 2024) and risk scores (Liu et al., 2022; Qinyu Zhu et al., 2023). More specifically, having defined a class of models, these procedures are able to rapidly find the corresponding (approximate) Rashomon sets which achieve a predictive performance within an θ-range of a reference model.

In this work, we continue the current effort studying the Rashomon Effect by delivering a specific procedure that aims at selecting a set of good predictive models under the *constraint of sparsity for high-dimensional problems* and for any class of functions defined by the user. The motivation behind this new procedure lies in another possible explanation for the existence of the Rashomon Effect for high-dimensional settings, namely the presence of *latent (unobserved) variables* that generate, or are associated to, the manifest variables used in practice to predict the response (which is actually linked to the latent variables). Indeed, the existence of the Rashomon Effect has mainly been explained this far by the presence of nondeterministic processes in the data which are affected by noise and uncertainty (Semenova et al., 2024). This indeed can explain how different model parametrizations from the same class, or even how models from different classes, can be considered almost-equally good. However, the presence of latent variables can also explain how models (in a given class) can contain different combinations of manifest variables that all have similar predictive performance based on how well these combinations represent the underlying latent (true) model. This is in line with Fisher et al. (2019) where they state that, even if the models in the set do not contain the true data generating process (e.g. the latent structure), one can hope that some models in the set can work in similar ways to it. While the study of latent variable modelling is vast (see e.g. Borsboom, 2008; Muthén & Muthén, 2009), including sparse latent representations (see e.g. Wu et al., 2022; Ahuja et al., 2022; Fumero et al., 2023), the idea of studying sparse latent structures through Rashomon sets is novel and can provide new ways of representing these problems. In particular, since "*the Rashomon Effect gives rise to simpler-yet-accurate models*" (Rudin et al., 2024), we focus on finding sparse representations of high-dimensional problems by finding sets of sparse models containing combinations of fewer variables which also have the advantage of delivering more interpretable models (Rudin, 2019; Rudin et al., 2022). In addition, the proposed procedure is class-independent since it consists in a wrapper heuristic in which the user can select their class of preference (similarly to Guerrier et al., 2016; Kissel & Mentch, 2024, for example). More specifically, the proposed procedure performs variable screening procedures and greedily combines the screened variables to deliver larger sets of sparse models. We call this procedure the "Sparse Wrapper AlGorithm" (SWAG) since any model class can be run within it (hence a wrapper) and it outputs a set of sparse representations for these models with good predictive performance. In particular, as highlighted earlier, this method delivers one class of possible perturbations recommended within the PCS framework put forward in Yu & Kumbier (2020) and therefore contributes to model-stability (see also Kissel & Mentch, 2024). The following section describes

the proposed algorithm, followed by various applied examples highlighting its advantages for different model classes and datasets.

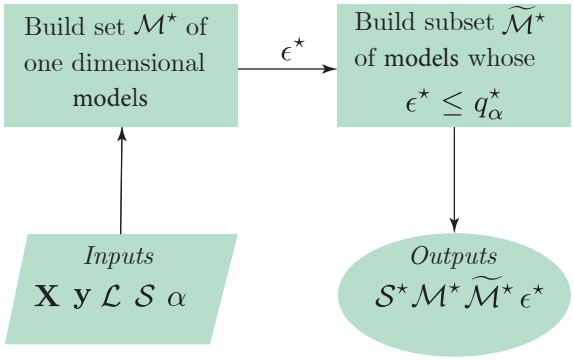
2 A Sparse Rashomon Algorithm

Let us start with a general overview of the SWAG. This algorithm proceeds in a forward-stepwise manner: it builds and tests models starting from one variable until it includes a maximal number of variables per model, increasing the number of variables at each step. Hence, for each fixed number of variables, the algorithm tests various (randomly selected) models and picks those with the best performance in terms of a chosen performance metric (e.g. test error estimates). Throughout, the algorithm uses the information coming from the best models at the previous step to build and test models in the following step. In the end, it outputs a set of sparse models with similar performance to the best one in the set. Given its nature, a more detailed comparison of this algorithm with the current stepwise selection procedures can be found in App. A.

To now provide a more formal and detailed description, let us define some basic notation. Let $\mathbf{y} \in \mathbb{R}^n$ denote the response and $\mathbf{X} \in \mathbb{R}^{n \times p}$ denote a variable (design) matrix with n samples and p variables, the latter being indexed by a set $\mathcal{S} := \{1, \dots, p\}$. In addition, we denote a class of functions (i.e. models) as $\mathcal{L} := \mathcal{L}(\mathbf{y}, \mathbf{X})$ with $l \in \mathcal{L}$ denoting a general model which is built by using a subset $s_l \in \mathcal{P}(\mathcal{S})$ of variables in $\mathbf{X}$, where we let $\mathcal{P}(\mathcal{A})$ and $|\mathcal{A}|$ denote respectively the power set and cardinality of a set $\mathcal{A}$. In the following paragraphs we will proceed to describing the algorithm and introduce meta-parameters whose interpretation and selection will be discussed later in Sec. 2.1. Also, based on this description, Sec. 2.2 will compare the SWAG models to Rashomon sets. It must also be noted that, given the heuristic nature of the algorithm, many of its steps described below can be adapted/modified according to the user's needs. This being said, the first choice to make for the SWAG is to determine the maximum dimension of variables that the user wants to be considered in a model and we denote this parameter as $p_{\max} < p$. Based on this parameter, the SWAG aims at exploring the space of variables to find sets of models using $\hat{p}$ variables ($1 \leq \hat{p} \leq p_{\max}$) with low error (or good prediction). In this respect, without loss of generality, we assume the choice of a performance metric that should be minimized, i.e. an error ϵ, such as estimates of test errors (e.g. cross-validation, Akaike Information Criterion etc.). With this in mind, the SWAG is described in the following paragraphs, where the first two algorithms are defined for a general model dimension $\hat{p}$ and are then used within the third algorithm which combines them in a greedy manner.

First Step The first screening step starts by using one *distinct* variable at a time to create p models. Once these models are built, a set of models $\mathcal{M}^\star$ is now available which is indexed by the ordered index set $\mathcal{I} := \{1, \dots, p\}$ (i.e. each model $l \in \mathcal{M}^\star$ is indexed by a unique element $i \in \mathcal{I}$). Having chosen a performance error ϵ, we denote the vector containing the errors of the p one-dimensional variable models as $\boldsymbol{\epsilon}^\star \in \mathbb{R}^p$ which is also indexed by the set $\mathcal{I}$ (i.e. each model $l \in \mathcal{M}^\star$ is associated with an element in the error vector $\boldsymbol{\epsilon}^\star$). Given this, it is now possible to select a performance quantile $q^\star_\alpha$, where $\alpha \in (0, 1)$ and the quantile estimator is chosen by the user (in this work we use the default choice for all statistical software, see e.g. Hyndman & Fan, 1996). The smaller the value of α, the smaller the errors selected. The procedure then selects all the models whose error is smaller or equal to $q^\star_\alpha$ and includes these in a new model set $\widetilde{\mathcal{M}}^\star$. The set $\widetilde{\mathcal{M}}^\star$ therefore collects one-dimensional models (i.e. models with one variable) with small errors and are therefore based on a subset of variables $\mathcal{S}^\star \subset \mathcal{S}$ that can be assumed to be predictive with respect to the response of interest $\mathbf{y}$. Similarly to other screening procedures, we assume that this procedure is able to select predictive variables with high probability. This assumption however does not necessarily hold when there are predictive variables in the data that only demonstrate importance (i.e. predictive power) when combined in models with other variables: if they exist, these variables will not be evaluated in the next steps of the SWAG even though they would be important to better predict the response. To avoid this limitation, modifications can obviously be made, including a change of the screening criterion or a less strict exclusion of non-screened variables in the following steps of the SWAG. Nevertheless, based on all examples presented in Sec. 3 (and in other applications), this limitation does not appear to affect the algorithm's performance compared to, for example, models that make use of all variables. This screening procedure is described in Algo. 1.

Remark. *This first step is conceived for high-dimensional problems (i.e. large p). However, this step can be avoided entirely if the dimension of the data is reasonable. Indeed, we would ideally like to skip this first step*



Algorithm 1 First Screening Algorithm

INPUTS: A response $\mathbf{y} \in \mathbb{R}^n$ and variables $\mathbf{X} \in \mathbb{R}^{n \times p}$; A variable index set $\mathcal{S} := \{1, \ldots, p\}$; A class of models $\mathcal{L}$; A performance percentile $\alpha \in (0, 1)$; Eventual parameters to compute the error ϵ.

1: Using the class $\mathcal{L}$, build p models by using all variables in the set $\mathcal{S}$
2: Create a model set $\mathcal{M}^\star$ (with $|\mathcal{M}^\star| = p$)
3: Build an error vector $\boldsymbol{\epsilon}^\star \in \mathbb{R}^p$ and identify the α-quantile $q_\alpha^\star$ of this vector
4: Create new model set $\widetilde{\mathcal{M}}^\star$ with models whose error is smaller or equal to $q_\alpha^\star$
5: Create variable index set $\mathcal{S}^\star$ with variables included in the models in the set $\mathcal{M}^\star$

OUTPUTS: $\mathcal{S}^\star$; $\widetilde{\mathcal{M}}^\star$; $\mathcal{M}^\star$; $\boldsymbol{\epsilon}^\star$

and consider all variables (or at least force some additional variables) in the following steps of the SWAG. This would imply that $\widetilde{\mathcal{M}}^\star = \{l \in \mathcal{L},\ \forall\, s_l \in \mathcal{S}\}$ and $\mathcal{S}^* = \mathcal{S}$ within the steps described below.

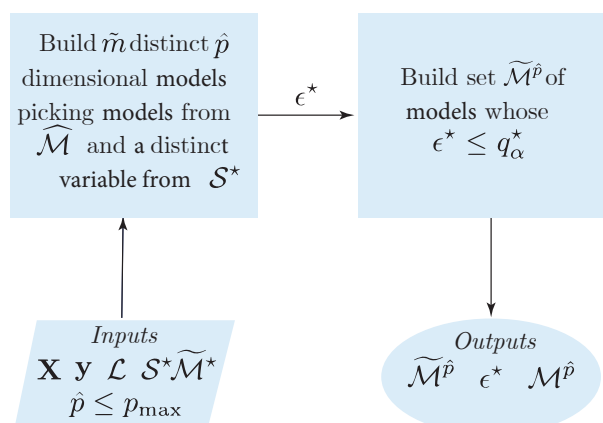
General Step Fixing a given variable dimension $\hat{p}$ such that $2 \leq \hat{p} \leq p_{\max}$, the general screening step builds a maximum number m of *distinct* models which will all be included in a model set $\mathcal{M}^{\hat{p}}$ where each model is built on combinations of $\hat{p}$ *distinct* variables. In order to build these m models, the general step takes the variable index set $\mathcal{S}^\star$ from Algo. 1 as well as a set of models $\widehat{\mathcal{M}}$ where each model is of dimension $\hat{p} - 1$ (i.e. each model in $\widehat{\mathcal{M}}$ takes $\hat{p} - 1$ variables as an input). We let $s_l \in \tilde{S} := \{s \in \mathcal{P}(\mathcal{S}^\star) \mid |s| = \hat{p} - 1\}$ denote the variable indices for a specific model $l \in \widehat{\mathcal{M}}$. By randomly selecting a model $l \in \widehat{\mathcal{M}}$, the general step will build a new model by using the variables indexed by s_l and a randomly selected distinct variable from the index set $\mathcal{S}^\star \setminus s_l$. This is repeated until m distinct models are built and included in the model set $\mathcal{M}^{\hat{p}}$. It must be noted that the total number of distinct models that can be built this way, say $\tilde{m}$, can actually be smaller than m. In this case, the set $\mathcal{M}^{\hat{p}}$ will contain all possible models that can be built through this procedure. Once the candidate model set $\mathcal{M}^{\hat{p}}$ is built, the general step of the algorithm closely follows Algo. 1. More specifically, an error vector $\boldsymbol{\epsilon}^{\hat{p}}$ is built and its performance quantile $q_\alpha^{\hat{p}}$ is computed. The main output of this step is a new model set $\widetilde{\mathcal{M}}^{\hat{p}}$ which includes all models whose error in $\boldsymbol{\epsilon}^{\hat{p}}$ is smaller or equal to $q_\alpha^{\hat{p}}$ (with both $\boldsymbol{\epsilon}^{\hat{p}}$ and $\mathcal{M}^{\hat{p}}$ being ordered by the same index set $\mathcal{I}^{\hat{p}} := \{1, \ldots, m\}$). This general step is described in Algo. 2.

Algorithm 2 General Screening Algorithm

INPUTS: A response $\mathbf{y} \in \mathbb{R}^n$ and variables $\mathbf{X} \in \mathbb{R}^{n \times p}$; A variable index set $\mathcal{S}^\star \subset \{1, \ldots, p\}$ from Algo. 1; A number of variables $\hat{p} \leq p_{\max}$; A model set $\widehat{\mathcal{M}}$; A class of models $\mathcal{L}$; A maximum number of models m; A performance percentile $\alpha \in (0, 1)$; Eventual parameters to compute the error ϵ.

1: Define total possible $\hat{p}$-dimenisonal models, i.e. $\tilde{m}$
2: **if** $\tilde{m} \leq m$ **then**
3: Using the class $\mathcal{L}$, build all possible $\tilde{m}$ models with $\hat{p}$ variable inputs to create model set $\mathcal{M}^{\hat{p}}$
4: **else**
5: Using the class $\mathcal{L}$, build m models with $\hat{p}$ variable inputs by extracting s_l from randomly sampled models $l \in \widehat{\mathcal{M}}$ and adding a variable from $\mathcal{S}^\star \setminus s_l$ to create model set $\mathcal{M}^{\hat{p}}$
6: **end if**
7: Build an error vector $\boldsymbol{\epsilon}^{\hat{p}}$ and identify the α-quantile $q_\alpha^{\hat{p}}$ of this vector
8: Create new model set $\widetilde{\mathcal{M}}^{\hat{p}}$ with models whose error is smaller or equal to $q_\alpha^{\hat{p}}$

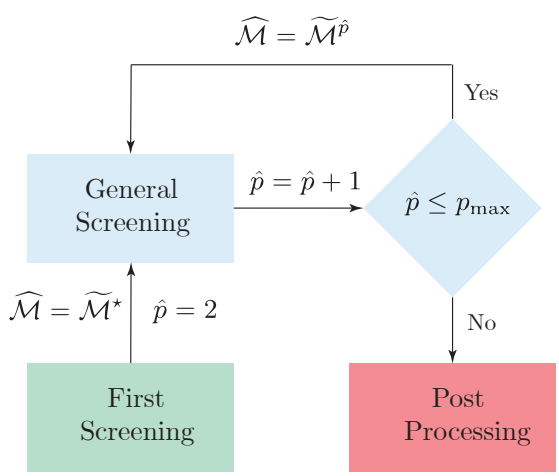
OUTPUTS: $\widetilde{\mathcal{M}}^{\hat{p}}$; $\mathcal{M}^{\hat{p}}$; $\boldsymbol{\epsilon}^{\hat{p}}$.



The above described algorithms provide a straightforward manner of selecting a set of good models for a given variable dimension $\hat{p}$. However, as the variable dimension increases, the number of possible distinct variable combinations increases exponentially fast leading to an increased risk of inefficiently exploring the variable space if one simply randomly picks m variable combinations. For this reason, the SWAG performs a *greedy* procedure which uses the information from Algo. 1 to obtain the set of best variables $\mathcal{S}^\star$ which is the easiest to explore completely. The next step of the algorithm, with $\hat{p} = 2$, then takes the set $\mathcal{S}^\star$ and the set of best models $\widetilde{\mathcal{M}}^\star$ as the input $\widehat{\mathcal{M}}$ for Algo. 2. At each of the following steps, with $\hat{p} > 2$, the algorithm defines $\widehat{\mathcal{M}} := \widetilde{\mathcal{M}}^{\hat{p}-1}$. Therefore, when increasing the variable dimension, the algorithm only considers variable combinations based on good models and variables from the previous dimension. This procedure is repeated for all variable dimensions until the maximal dimension $p_{\max}$ is reached. Throughout the procedure, the algorithm saves the good model sets and error vectors for each dimension $\hat{p}$. More formally, letting l_i and $\boldsymbol{\epsilon}_i$ represent the ith model and its associated error metric respectively, the set of SWAG models can be therefore defined as:

$$\widetilde{\mathcal{M}} := \bigcup_{\hat{p}=1}^{p_{\max}} \left\{ l_i \in \mathcal{M}^{\hat{p}} \,\middle|\, \boldsymbol{\epsilon}_i^{\hat{p}} \leq q_\alpha^{\hat{p}} \,, \forall\, i \in \mathcal{I}^{\hat{p}} \right\}.$$

This procedure defines the SWAG which is described in Algo. 3. Let us now consider some aspects of interest for this algorithm, discussed in the following paragraphs.



Algorithm 3 SWAG

INPUTS: A response $\mathbf{y} \in \mathbb{R}^n$ and variables $\mathbf{X} \in \mathbb{R}^{n \times p}$; A variable index set $\mathcal{S} := \{1, \ldots, p\}$; A class of models $\mathcal{L}$; A maximum number of variables $p_{\max}$ $(< p)$; A maximum number of models m for each step; A performance percentile $\alpha \in (0, 1)$; Eventual parameters to compute the error ϵ.

1: Run Algo. 1 using inputs $\mathbf{y}$, $\mathbf{X}$, $\mathcal{S}$, $\mathcal{L}$, m, α and obtain $\mathcal{S}^\star$ and $\widetilde{\mathcal{M}}^\star$
2: $\widehat{\mathcal{M}} \leftarrow \widetilde{\mathcal{M}}^\star$
3: $\hat{p} \leftarrow 2$
4: **while** $\hat{p} \leq p_{\max}$ **do**
5: Run Algo. 2 using inputs $\mathbf{y}$, $\mathbf{X}$, $\mathcal{S}^\star$, $\widehat{\mathcal{M}}$, $\mathcal{L}$, m, α and obtain $\widetilde{\mathcal{M}}^{\hat{p}}$
6: $\widehat{\mathcal{M}} \leftarrow \mathcal{M}^{\hat{p}}$
7: $\hat{p} \leftarrow \hat{p} + 1$
8: **end while**
9: Create
- a set of good model sets $\widetilde{\mathcal{M}} := \{\widetilde{\mathcal{M}}^\star, \widetilde{\mathcal{M}}^2, \ldots, \widetilde{\mathcal{M}}^{p_{\max}}\}$
- a set of error vectors $\tilde{\epsilon} := \{\epsilon^\star, \epsilon^2, \ldots, \epsilon^{p_{\max}}\}$

OUTPUTS: $\widetilde{\mathcal{M}}$; $\tilde{\epsilon}$

Post-Processing The user could choose to directly make use of the set of good models $\widetilde{\mathcal{M}}$ to make predictions based on different sets of variables or arrange variables into networks for interpretation and exploration (see for example Figs. 2 and 3 further on). However, these sets of models could undergo an additional screening procedure according to the needs of the user. For example, a possible approach that is used for the applications in Sec. 3 is to compute the median error for each vector in the set $\tilde{\epsilon}$ (as defined in Algo. 3) and select the quantile $\tilde{q}_\delta$ corresponding to the dimension whose median is the lowest, where $\delta \in (0, 1)$ can differ from α (a possible choice is 0.01). Having identified this quantile, we then select the models (with the desired dimensions) whose error is smaller or equal to this quantile. In particular, this procedure would deliver a set closer in spirit (but not equivalent) to the original definition of the Rashomon set which contains all models whose performance is within an θ-range of the reference model (see Sec. 2.2 for the definition). For this specific choice of post-processing, the quantile $\tilde{q}_\delta$ would represent the upper bound defining the performance of models in the Rashomon set, i.e.:

$$\widetilde{\mathcal{M}}_{\mathcal{R}} := \bigcup_{\hat{p}=1}^{p_{\max}} \left\{ l_i \in \mathcal{M}^{\hat{p}} \,\middle|\, \boldsymbol{\epsilon}_i^{\hat{p}} \leq \tilde{q}_\delta \,, \forall\, i \in \mathcal{I}^{\hat{p}} \right\}.$$

Computational Complexity The SWAG is a computationally intense procedure and builds at most $p + m\,(p_{\max} - 1)$ models in total which implies that its order of computation is proportional to this factor. In addition, the choice of the error and its computation can add complexity to the procedure (e.g. r-repeated k-fold cross-validation). Indeed, letting $C(n, \hat{p})$ represent the complexity of fitting a model $l \in \mathcal{L}$ to a dataset with n samples and $\hat{p}$ variables, the worst-case complexity is given by $O\big(r\,k\,\big(p\,C(n,1) + p_{\max}\,m\,C(n, p_{\max})\big)\big)$. If a more efficient evaluation of the test error ϵ is available, this complexity can be considerably reduced. As an example, if using a generalized linear model with complexity $O(n\hat{p}^2 + \hat{p}^3)$ and using AIC or BIC as the error metric ϵ, then the worst-case complexity becomes $O\left(np + mnp_{\max}^3 + mp_{\max}^4\right)$. One could even further reduce computational complexity by directly using training error metrics ϵ_{train} since the parameter $p_{\max}$ could already be considered as a regularizer to limit over-fitting (see Efron et al., 2004, and discussions in Sec. 2.1). A more detailed discussion of SWAG's computational complexity is found in App. B.

Ensemble Learning and Stability The goal of the SWAG does not consist in improving prediction accuracy *per se*, but more in providing user-flexibility and interpretability while preserving accuracy (to the best extent possible) by relying on individual models chosen from the diverse representations in the set $\widetilde{\mathcal{M}}$. Nevertheless, nothing stops us from using this set of good sparse models within an ensemble approach. For example the SWAG models could be included in Bagging (Breiman, 1996), Boosting (Schapire, 1990), Stacking (Wolpert, 1992) or other model-averaging approaches (see e.g., Raftery et al., 1997). In this sense, similarly to Kissel & Mentch (2024), the SWAG provides a procedure to define model perturbations as recommended, among others, within the PCS framework of Yu & Kumbier (2020).

2.1 SWAG Parameters

For the application of the SWAG, we assume the user has chosen a class of models $\mathcal{L}$ for which they would like to obtain sparse representations without losing significant predictive power. Based on this, the user has to define the meta-parameters of the algorithm which are (i) the maximum variable dimension $p_{\max}$; (ii) the maximum number of models m to be built within each step and (iii) the performance percentile α. Ideally, with unlimited computing power, the first two parameters would be as large as possible, i.e. $p_{\max} = p$ and $m = \binom{p}{\lceil \frac{p}{2} \rceil}$, leading to best subset selection. However, this defeats the purpose of the algorithm and therefore the decision of these parameters must be based mainly on interpretability/flexibility requirements as well as available computing power and time. Below are some rules-of-thumb for the choice of these parameters:

- $p_{\max}$: Fixing the available computing power and the efficiency in the computation of the model class $\mathcal{L}$, this parameter will depend on the total dimension of the problem p. Indeed, the goal of the SWAG is to find very sparse models and the parameter $p_{\max}$ can be viewed as a regularization parameter (Efron et al., 2004). Therefore, even with very large p, one can always fix this parameter within a range of 5-20 (or smaller) for interpretability and/or replicability purposes. Another criterion, when working with binary classification problems, is to use the *Event Per Variable* (EPV) rule presented in Vittinghoff & McCulloch (2007) and discussed in van der Ploeg et al. (2014) (see Sec. 3 for example). This rule aims at limiting the number of variables included in models to ensure that enough data points are available in each dimension spanned by these variables, thereby preserving sufficient information to adequately estimate such models. In future work, this parameter can be implicitly determined by the algorithm based on the error quantile (or other metric) as a stopping-criterion thereby defining $p_{\max}$ as the variable dimension where the error curve stops decreasing significantly similarly to the scree plot in factor or principal component analysis (see e.g. Cattell, 1966).

- m: Fixing the available computing power and the efficiency in the computation of the model class $\mathcal{L}$, this parameter will determine the proportion of variable space that will be explored by the algorithm. We know that it depends on the size of the problem p since we necessarily have $m \geq p$ for the screening step of Algo. 1. In addition, this parameter needs to be chosen considering the performance percentile α: if m is small and α is small, then the number of good models being selected could be extremely low (possibly zero) since not enough variables may be selected from previous steps to build m combinations. In general, we would want a large m (so that α can eventually be chosen to be very small) and, remembering that $p^\star$ is the number of variables released from Algo. 1, a

rule-of-thumb is to set $m = \binom{p^\star}{2}$ (or close to it) in order to explore the entire (or most of the) subspace of two-dimensional models generated by $p^\star$.

- α: as discussed in the previous point, this parameter is related to the maximum number of models m. The larger α, the more the variable space is explored. Ideally, we want to choose a small α since we would want to select good models (with extremely low error) and this is possible if m is large enough. Generally good values for α are 0.01 or 0.05, implying that (roughly) 1% or 5% of the m models are selected at each step.

The SWAG could be modified to perform tests to determine the best models across the different dimensions (e.g. equivalence tests), therefore removing the need to specify α. This modification however is left for future research. A study on the sensitivity of the SWAG to these meta-parameters is given in App. D where results suggest that algorithm outputs are stable with respect to these choices.

2.2 SWAG Libraries and Rashomon Sets

It is reasonable to ask ourselves how the set of SWAG models $\widetilde{\mathcal{M}}$ (or $\widetilde{\mathcal{M}}_{\mathcal{R}}$ after post-processing) compare to the formal definition of a Rashomon set. For this reason, let us recall this definition which generally starts from a reference model $l^* \in \mathcal{L}$, based on which the Rashomon set $\mathcal{R}_\theta(l^*)$ is defined as:

$$\mathcal{R}_\theta(l^*) := \{l \in \mathcal{L} \,|\, D(l) \leq D(l^*) + \theta\},$$

where D is a loss function (in our notation ϵ) and $\theta > 0$ is a user-defined parameter which defines the margin within which model performance can be considered "almost-equivalent" (see e.g. Fisher et al., 2019). In this respect, the SWAG does not predefine a reference model l^* or an "almost-equivalence" parameter θ but finds "almost-equivalent" good sparse models through the percentile α which is therefore not absolute (like θ) but relative to the performance of all models evaluated in a given dimension $\hat{p}$. Following the definition of a Rashomon set, technically speaking, if we were to define a reference model l^* then there would exist a θ for each dataset and model class such that the SWAG library constitutes a sparse subset of the respective Rashomon sets. However this is a post-hoc adaptation of the definition since the exploration of "almost-equally" performing models occurs differently between the SWAG and the formal definition of a Rashomon set. This being said, the SWAG can easily be modified to more directly search for models in the Rashomon set by selecting some sparse models whose loss (e.g. generalization error) is within a θ-range of a pre-specified model l^*. Therefore, while these sets are indeed built on different criteria (and are thus not directly comparable in terms of outputs), one could argue that the SWAG targets a sparse subset of the Rashomon set for a given model class $\mathcal{L}$ (see results in Sec. 3), thereby enhancing its interpretability for practitioners—particularly in light of the growing emphasis on "Explainable AI" (see e.g. Confalonieri et al., 2021, for a historical perspective). Moreover, under certain aspects, it can be considered more intuitive in the way models from a certain class are selected. For example, the SWAG does not require the pre-specification of a reference model l^* and, in addition, the parameter defining the models in the SWAG set is a performance percentile rather than a pre-specified parameter θ related to the loss value D (for which a reasonable range may sometimes be hard to determine *a priori*). However, as opposed to some currently available Rashomon set procedures (e.g. Xin et al., 2022; Qinyu Zhu et al., 2023; Zhong et al., 2024), the SWAG does not guarantee to find *all* models (but only some) with almost-equivalent performance for dimensions $1 \leq \hat{p} \leq p_{\max}$.

3 Empirical Results

We study the empirical performance of the SWAG with different model classes and on different datasets taken from the UCI Machine Learning Repository (see Dua & Graff, 2017), ArrayExpress (see Kolesnikov et al., 2015) and from the GitHub repository `https://github.com/ramhiser/datamicroarray`. More specifically, the chosen model classes are: (i) Lasso (logistic) (Friedman et al., 2010); (ii) Linear-Kernel Support Vector Machine (L-SVM) (see e.g., (Vapnik, 2013)); (iii) Radial-Kernel SVM (R-SVM) (Cortes & Vapnik, 1995); (iv) Random Forest (RF) (Breiman, 2001a). Compared to these models classes, the use of more complex models, such as neural networks, requires more thought on how to adapt their architecture as the number and combination of variables change throughout the SWAG (this is left for future work). To ensure a fair

comparison, we run all the analyses with the `caret` R package (see e.g. Kuhn, 2008, for a review). All the hyper-parameters specific to each model class are set based on common/default choices and are discussed in App. C. With regards to the datasets, they are the following: (i) MeterA (Gyamfi et al., 2018); (ii) LSVT (Tsanas et al., 2013); (iii) Ahus (Haakensen et al., 2016); (iv) Colon (Alon et al., 1999); (v) Leukemia (Golub et al., 1999). The details regarding these data and the choices for the SWAG meta-parameters are listed in Tab. 1: the choice of the SWAG meta-parameters is made in line with the rules of thumb described in Sec. 2.1 and with the intent of reducing overall computational time while ensuring a reasonable exploration of the variable space. More specifically, we choose $0.01 \leq \alpha \leq 0.05$ as a good compromise between fixing an α as small as possible (to select good models) and the possibility of exploring a considerable portion of the variable space. With m depending on the computing time and power available, we choose m in order to at least reasonably explore the subspace of two-dimensional models for all the model classes and datasets. The choice of $p_{\max}$ is made guided by the EPV rule discussed in van der Ploeg et al. (2014). As mentioned earlier, an extended study and discussion on the sensitivity of the SWAG results with respect to the choice of these meta-parameters can be found in App. D where, for example, it is also underlined how similar choices are needed for Rashomon sets regarding the choice of a reference model and "equivalence" parameter θ. Finally, we apply 10-repeated 10-fold cross-validation (i.e. $r = 10$ and $k = 10$) as the error metric ϵ and choose $\delta = 0.01$ for the post-processing based on the previously described median rule.

Table 1: For each dataset: # of instances per training and test set, # of variables and relative SWAG meta-parameter choices.

Data	n_{train}	n_{test}	p	$p_{\max}$	m	α
MeterA	69	18	666	6	3984	0.05
LSVT	100	26	310	6	1014	0.05
Ahus	125	31	15,739	8	99080	0.01
Colon	50	12	2,000	4	7996	0.03
Leukemia	58	14	7,129	4	10160	0.01

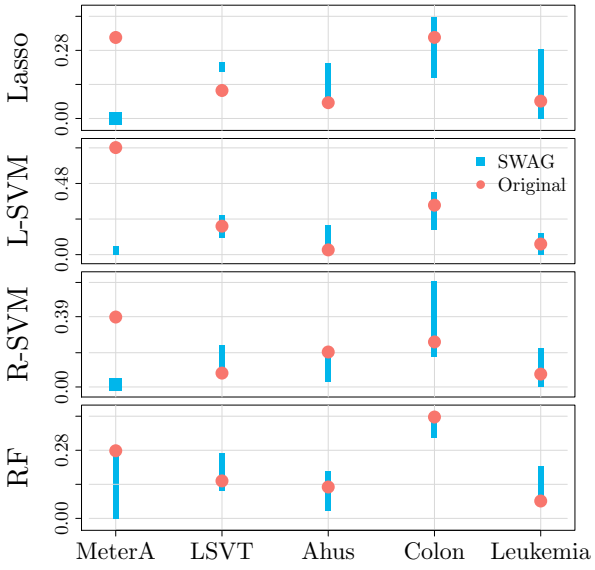


Figure 1: SWAG test error range ϵ_{test} (blue rectangles) and original model test error (red dots). The blue *squares* imply that the SWAG models all have the same test error.

With the above in mind, we present some summary measures of the set of SWAG models $\widetilde{\mathcal{M}}$ such as their dimensions and their Jaccard diversity indices, allowing to understand how interpretable (sparse) and how flexible (diverse) the resulting SWAG models are (see App. E.3 and E.4 for details on number of SWAG models $|\widetilde{\mathcal{M}}|$ and Jaccard index). Tab. 2 shows the range of dimensions of the SWAG models ($|s_l|$), the median value for the pairwise Jaccard indices for all models in $\widetilde{\mathcal{M}}$ (med_J) as well as their range (range_J) for each dataset. It can be seen how the SWAG models are all extremely sparse (also as a result of $p_{\max}$) and that at least half of the Jaccard indices for all model classes and datasets are below or equal to 0.5, indicating that there is a reasonable or high level of diversity in the SWAG models which is essential for user-flexibility as well as interpretability of latent structures. Indeed, with respect to user-flexibility, variable diversity can be important for practitioners who need similarly good models with variables that are less costly/invasive to collect and measure, or that can be used when different institutions do not collect the same variables (see e.g. Kissel & Mentch, 2024). However, an important aspect to underline is that, while the goal of the SWAG is to deliver the latter two advantages, these seem to come with a limited cost in terms of prediction error with respect to the original reference models, thereby suggesting it is capable of finding *sparse subsets of the Rashomon set* (if these exist). Indeed, Fig. 1 represents the test error, denoted as ϵ_{test}, of the original models (i.e. models trained using all variables) represented by the

Table 2: Model class (main columns) and datasets (rows): (i) range of SWAG model dimensions, (ii) median Jaccard index and (iii) range of the Jaccard indices for the SWAG models.

	Lasso			L-SVM			R-SVM			RF		
	$\lvert s_l \rvert$	med_J	range_J	$\lvert s_l \rvert$	med_J	range_J	$\lvert s_l \rvert$	med_J	range_J	$\lvert s_l \rvert$	med_J	range_J
MeterA	$[4, 6]$	0.38	$[0.09, 0.83]$	$[2, 6]$	0.16	$[0.00, 0.83]$	$[4, 6]$	0.20	$[0.09, 0.83]$	$[4, 6]$	0.33	$[0.09, 0.83]$
LSVT	$[4, 6]$	0.15	$[0.20, 0.83]$	$[5, 6]$	0.09	$[0.00, 0.83]$	$[4, 6]$	0.38	$[0.33, 0.83]$	$[3, 6]$	0.25	$[0.00, 0.83]$
Ahus	$[6, 8]$	0.15	$[0.00, 0.88]$	$[7, 8]$	0.23	$[0.00, 0.88]$	$[5, 8]$	0.23	$[0.00, 0.75]$	$[5, 8]$	0.25	$[0.00, 0.75]$
Colon	$[3, 4]$	0.14	$[0.00, 0.75]$	$[3, 4]$	0.33	$[0.00, 0.75]$	$[3, 4]$	0.14	$[0.00, 0.75]$	$[3, 4]$	0.40	$[0.33, 0.75]$
Leukemia	$[2, 3]$	0.00	$[0.00, 0.67]$	$[2, 3]$	0.20	$[0.00, 0.67]$	$[2, 3]$	0.00	$[0.00, 0.67]$	3	0.50	$[0.20, 0.50]$

red dots, and of the respective SWAG models, represented by a blue rectangle proportional to the range of test errors (a similar figure is available for the training error ϵ_{train} in App. E.2). It can be seen how in the majority of cases the SWAG models have close or comparable (if not sometimes better) prediction accuracy with respect to their original versions, with homogeneous and exact accuracy when using Lasso and R-SVM on the Meter A data for example. This range of performance of SWAG models is expected and in line, for example, with the definition of Rashomon sets which are built on "almost-equivalent" models within a θ-range of the reference one. Indeed, in our case, the prediction performance is overall preserved while selecting extremely sparse models (maximum dimension 8) while the Lasso (the only sparse alternative considered) selects between 10 and 26 variables. Moreover, the user can arbitrarily choose to trim these distributions further by restricting their final selection to the best models in the SWAG library.

Further extensive experiments are provided in App. E supporting the stability of the SWAG across (i) meta-parameters, (ii) datasets and (iii) model classes. Indeed, conclusions similar to those presented above hold even when using existing (step-wise) variable selection approaches as possible good reference models as well as across different meta-parameter and model-class choices (see e.g. App. A and D). Given the fact that prediction accuracy and stability is generally preserved with the SWAG models, let us investigate the advantages of the SWAG in terms of interpretability of latent structures (along with user-flexibility which was observed through the Jaccard index above). To do so we consider building SWAG networks, based on the final selected set of models, in which the size of the nodes are proportional to the presence of a variable in SWAG models while the thickness of edges is proportional to times the variables are in the same SWAG model. We briefly study and interpret the latent structures highlighted by these networks for two of the considered datasets (more information in App. E.5).

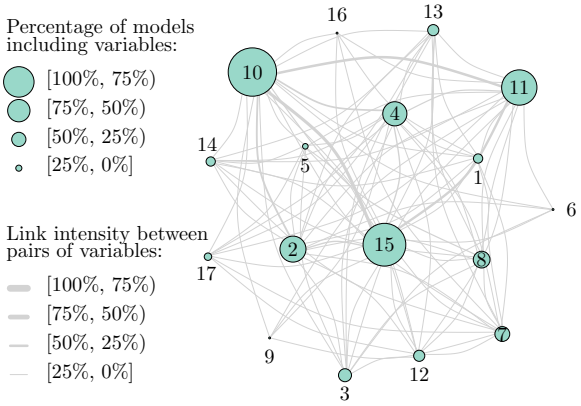


Figure 2: Network of variable importance and pairwise link-intensity in $\widetilde{\mathcal{M}}$ for Meter A dataset using the Lasso-based SWAG.

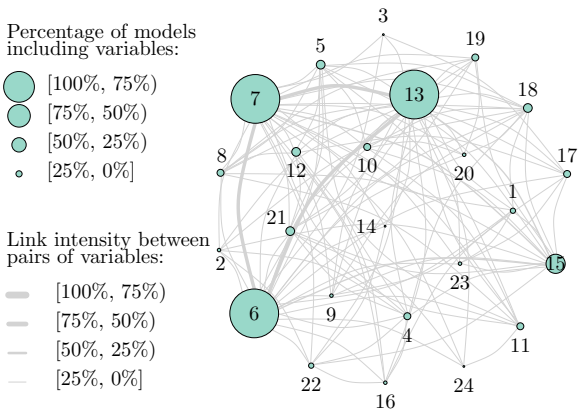


Figure 3: Network of variable importance and pairwise link-intensity in $\widetilde{\mathcal{M}}$ for LSVT dataset using the Radial-Kernel SVM SWAG.

Ultrasonic Flowmeters The Meter A data is analyzed in Gyamfi et al. (2018) and collects measurements on ultrasonic flowmeter diagnostics. Achieving good diagnostics regarding the health of a flowmeter is of extreme importance for condition-based maintenance in many industrial sectors such as the oil and gas industry since incorrect measurement can entail considerable economic and material losses (see e.g. TUV-NEL, 2012). In this data the goal is to classify the health of a meter into two classes: "Healthy" (Class 1) or "Installation effects" (Class 2). Given that the variables are measurements of physical nature, we decide to consider all first-order interactions which finally delivers a total variable size of $p = 666$ (36 original variables plus $\binom{36}{2} = 630$ interactions). Indeed, we do this since these interaction effects could not be adequately captured by models that only contain the variables of interest on their own (see e.g. Jaccard & Turrisi, 2003). Using the results from Lasso-based SWAG, the network in Fig. 2 would suggest that in order to understand the mechanics and perform diagnostics for this ultrasonic flowmeter, among the 666 variables, an engineer could focus their attention on variables labeled 10, 11 and 15 which correspond to the interaction (i) between flatness ratio and gain as well as (ii) between the speed of sound and gain at the first end (of the fifth path). For instance, the interaction between flatness ratio and gain is significant since, if the flatness ratio is low, it indicates possible disturbances in the flow that could affect the gain measurements, leading to potential inaccuracies in flow estimation. Moreover, combining the internal speed of sound with gains can help to detect anomalies that individual measurements might miss. This relationship suggests that efficiency in signal transmission could degrade over time and impact various paths differently. These variables are all linked to characteristics such as structural integrity, alignment quality and environmental conditions, suggesting that these could be the latent variables affecting the health of the flowmeter (Lynnworth & Liu, 2006).

LSVT The voice rehabilitation dataset was analyzed in Tsanas et al. (2013) in order to assess the effectiveness of a computer program called "Lee Silverman voice treatment (LSVT) Companion" which allows patients with Parkinson's disease (PD) to independently progress through a rehabilitative treatment session. Taking data on 126 samples from 14 patients who followed the latter treatment, 310 dysphonia measures were taken on each of them (plus information on sex and age of the patients) and used to understand if they could correctly predict whether the patients' voices were "acceptable" or "unacceptable" after this treatment. There is also scientific interest in determining the variables (and combinations thereof) that most contribute to the definition, in this case, of a Parkinson's speech treatment as being acceptable or not. Also in this case, the SWAG models (based on the R-SVM) can be arranged into a network to allow for interpretation as seen in Fig. 3. Based on the SWAG network, researchers interested in improving speech treatment should focus on variables 6, 7 and 13 which correspond to the 2nd and 3rd Mel-Frequency Cepstral Coefficients (MFCC) and to the entropy with base-4 logarithmic coefficients (as well as the interactions between these three variables as highlighted by their frequent presence in the same SWAG models). These variables indirectly measure latent characteristics such as voice quality and cognitive load: (i) fluctuations in MFCC can reflect changes in the vocal tract configuration, corresponding to mechanical or motor control difficulties in speech production in PD patients while (ii) entropy values can indirectly suggest the cognitive effort involved in speech production which, in patients with PD, can lead to reduced speech variability and difficulties in modulation, impacting treatment effectiveness (Dao et al., 2022; Zhang et al., 2023; Shen et al., 2024).

4 Conclusions

The proposed algorithm does not directly fit in the standard procedures to build Rashomon sets since these are commonly found starting from a reference model and explore all model dimensions. More specifically, without a reference model and by fixing a model class, the SWAG assumes the existence of sparse models in the Rashomon set of this class and aims to find (some of) them. By doing so, we can see that the SWAG produces many of the main results of the Rashomon Effect (Rudin et al., 2024), in particular: (i) the selection of many sparse-yet-accurate models (on par with their original counterparts applied to all variables) as shown by Fig. 1; (ii) flexibility to provide users with diverse and sparse models that are interchangeable in terms of predictive performance as highlighted by Tab. 2; (iii) evaluation of prediction uncertainty through SWAG error (or accuracy) ranges, in particular model-uncertainty (see again Fig. 1); (iv) stable variable importance metrics that can be interpreted also in terms of connections with other variables (see Fig. 2 and 3). Moreover, as shown in Sec. 3, all this is achieved through any model class of choice. Indeed, the possibility of studying the role of variables in a network (i.e. the change of effects when combined with other variables) can also

be a tool to address the *replication crisis* occurring in many domains. In this sense, finding the best single model in each study does not allow to assess how generalizable the successive interpretations can be, whereas sets of models can stabilize interpretations by verifying which variables have stable effects in the network. In particular, these properties directly address some of the requirements of the PCS framework of Yu & Kumbier (2020), specifically the model-perturbation that ensures that the final model(s) are more generalizable.

Given its heuristic nature, the proposed algorithm can be easily adapted for different user preferences, including the selection of sparse model sets based on domain-specific utility (cost) functions. Future directions of research for this procedure include the theoretical study of its properties when in the presence of sparse latent representations, including its ability to select models built with weakly correlated variables which are linked to distinct underlying latent functions. Moreover, the structure of the SWAG can possibly allow to construct inferential tools to determine the significance of the model set $\widetilde{\mathcal{M}}$, specifically if there are indeed important variables/models or if the set contains randomly selected models due to spurious correlations and/or to the absence of significant predictors. Using the network representations of the SWAG models, one could employ network statistics to overcome the problem of testing each model in the set individually, thereby delivering the first statistical tools for testing Rashomon sets.

References

Kartik Ahuja, Jason S Hartford, and Yoshua Bengio. Weakly supervised representation learning with sparse perturbations. *Advances in Neural Information Processing Systems*, 35:15516–15528, 2022.

Hirotugu Akaike. Information theory and an extension of the maximum likelihood principle. pp. 267–281, 1973.

Uri Alon, Naama Barkai, Daniel A Notterman, Kurt Gish, Suzanne Ybarra, Daniel Mack, and Arnold J Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96(12):6745–6750, 1999.

D Anderson and K Burnham. Model selection and multi-model inference. *Second. NY: Springer-Verlag*, 63 (2020):10, 2004.

Davide Bombarda, Giorgio Matteo Vitetta, and Giovanni Ferrante. Rail diagnostics based on ultrasonic guided waves: An overview. *Applied Sciences*, 11(3):1071, 2021.

Denny Borsboom. Latent variable theory. 2008.

L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001a.

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

Kenneth P Burnham and David R Anderson. Multimodel inference: understanding aic and bic in model selection. *Sociological methods & research*, 33(2):261–304, 2004.

J. M. Carmona-Cejudo, M. Baena-García, J. Campo-Avila, R. Morales-Bueno, J. Gama, and A. Bifet. Using gnusmail to compare data stream mining methods for on-line email classification. In *Proceedings of the Second Workshop on Applications of Pattern Analysis*, pp. 12–18, 2011.

R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes. Ensemble selection from libraries of models. In *Proceedings of the 21st International Conference on Machine Learning*, pp. 18, 2004.

R. B. Cattell. The scree test for the number of factors. *Multivariate Behavioral Research*, 1(2):245–276, 1966.

Roberto Confalonieri, Ludovik Coba, Benedikt Wagner, and Tarek R Besold. A historical perspective of explainable artificial intelligence. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 11(1):e1391, 2021.

C. Cortes and V. Vapnik. Support vector machine. *Machine Learning*, 20(3):273–297, 1995.

S. F. Crone. Training artificial neural networks for time series prediction using asymmetric cost functions. In *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP'02.*, volume 5, pp. 2374–2380. IEEE, 2002.

Son VT Dao, Zhiqiu Yu, Ly V Tran, Phuc NK Phan, Tri TM Huynh, and Tuan M Le. An analysis of vocal features for parkinson's disease classification using evolutionary algorithms. *Diagnostics*, 12(8):1980, 2022.

S. Draghici, P. Khatri, A. C. Eklund, and Z. Szallasi. Reliability and reproducibility issues in dna microarray measurements. *TRENDS in Genetics*, 22(2):101–109, 2006.

D. Dua and C. Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2): 407–499, 2004.

Michael A Efroymson. Multiple regression analysis. In Anthony Ralston and Herbert Wilf (eds.), *Mathematical Methods for Digital Computers*, pp. 191–203. Wiley, 1960.

Michael Elad and Irad Yavneh. A plurality of sparse representations is better than the sparsest one alone. *IEEE Transactions on Information Theory*, 55(10):4701–4714, 2009.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

Jianqing Fan and Jinchi Lv. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):849–911, 2008.

Jianqing Fan, Richard Samworth, and Yichao Wu. Ultrahigh dimensional feature selection: Beyond the linear model. *Journal of Machine Learning Research*, 10(Oct):2013–2038, 2009.

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.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1, 2010.

Marco Fumero, Florian Wenzel, Luca Zancato, Alessandro Achille, Emanuele Rodolà, Stefano Soatto, Bernhard Schölkopf, and Francesco Locatello. Leveraging sparse and shared feature activations for disentangled representation learning. *Advances in Neural Information Processing Systems*, 36:27682–27698, 2023.

R. Genuer, J.-M. Poggi, and C. Tuleau-Malot. Variable selection using random forests. *Pattern Recognition Letters*, 31(14):2225–2236, 2010.

Todd R Golub, Donna K Slonim, Pablo Tamayo, Christine Huard, Michelle Gaasenbeek, Jill P Mesirov, Hilary Coller, Mignon L Loh, James R Downing, Mark A Caligiuri, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, 286(5439):531–537, 1999.

S. Guerrier, N. Mili, R. Molinari, S. Orso, M. Avella-Medina, and Y. Ma. A predictive based regression algorithm for gene network selection. *Frontiers in Genetics*, 7:97, 2016.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

K. S. Gyamfi, J. Brusey, A. Hunt, and E. Gaura. Linear dimensionality reduction for classification via a sequential bayes error minimisation with an application to flow meter diagnostics. *Expert Systems with Applications*, 91:252–262, 2018.

V. D. Haakensen, V. Nygaard, L. Greger, M. R. Aure, B. Fromm, I. R.K. Bukholm, T. Lüders, S.-F. Chin, A. Git, C. Caldas, et al. Subtype-specific micro-rna expression signatures in breast cancer progression. *International Journal of Cancer*, 139(5):1117–1128, 2016.

Xavier A Harrison, Lynda Donaldson, Maria Eugenia Correa-Cano, Julian Evans, David N Fisher, Cecily ED Goodwin, Beth S Robinson, David J Hodgson, and Richard Inger. A brief introduction to mixed effects modelling and multi-model inference in ecology. *PeerJ*, 6:e4794, 2018.

Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844, 1998.

Rob J Hyndman and Yanan Fan. Sample quantiles in statistical packages. *The American Statistician*, 50(4): 361–365, 1996.

James Jaccard and Robert Turrisi. *Interaction effects in multiple regression*. Number 72. sage, 2003.

Nicholas Kissel and Lucas Mentch. Forward stability and model path selection. *Statistics and Computing*, 34 (2):82, 2024.

N. Kolesnikov, E. Hastings, M. Keays, O. Melnichuk, Y. A. Tang, E. Williams, M. Dylag, N. Kurbatova, M. Brandizi, and T. Burdett. Arrayexpress update—simplifying data submissions. *Nucleic Acids Research*, 43(D1):D1113–D1116, 2015.

M. Kuhn. Building predictive models in r using the caret package. *Journal of Statistical Software*, 28(5):1–26, 2008.

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.

LC Lynnworth and Yi Liu. Ultrasonic flowmeters: Half-century progress report, 1955–2005. *Ultrasonics*, 44: e1371–e1378, 2006.

H. Masnadi-Shirazi and N. Vasconcelos. Asymmetric boosting. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 609–619, 2007.

Bengt Muthén and Bengt O Muthén. *Statistical analysis with latent variables*, volume 123. Wiley New York, 2009.

Chloe Qinyu Zhu, Muhang Tian, Lesia Semenova, Jiachang Liu, Jack Xu, Joseph Scarpa, and Cynthia Rudin. Fast and interpretable mortality risk scores for critical care patients. *arXiv e-prints*, pp. arXiv–2311, 2023.

A. E. Raftery, D. Madigan, and J. A. Hoeting. Bayesian model averaging for linear regression models. *Journal of the American Statistical Association*, 92(437):179–191, 1997.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. nat mach intell 1 (5): 206–215, 2019.

Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16:1–85, 2022. doi: 10.1214/21-SS133.

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.

R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

Gideon Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

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*, pp. 1827–1858, 2022.

Lesia Semenova, Harry Chen, Ronald Parr, and Cynthia Rudin. A path to simpler models starts with noise. *Advances in neural information processing systems*, 36, 2024.

Matthew Shen, Pouria Mortezaagha, and Arya Rahgozar. Explainable artificial intelligence to diagnose early parkinson s disease via voice analysis. *medRxiv*, pp. 2024–09, 2024.

Xiaoyu Tang, Xiang Xie, Hongjian Zhang, and Hongliang Zhou. Data integration for multi-path ultrasonic flowmeter based on levenberg–marquardt algorithm. *IET Science, Measurement & Technology*, 9(8): 909–920, 2015.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 58(1):267–288, 1996.

Athanasios Tsanas, Max A Little, Cynthia Fox, and Lorraine O Ramig. Objective automatic assessment of rehabilitative speech treatment in parkinson's disease. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(1):181–190, 2013.

TUV-NEL. Testing the diagnostic capabilities of liquid ultrasonic flow meters. *National Measurement System*, 2012.

Tjeerd van der Ploeg, Peter C Austin, and Ewout W Steyerberg. Modern modelling techniques are data hungry: a simulation study for predicting dichotomous endpoints. *BMC medical research methodology*, 14 (1):1–13, 2014.

V. Vapnik. *The Nature of Statistical Learning Theory*. Springer science & business media, 2013.

L. Vaughan and Y. Chen. Data mining from web search queries: A comparison of google trends and baidu index. *Journal of the Association for Information Science and Technology*, 66(1):13–22, 2015.

E. Vittinghoff and C. E. McCulloch. Relaxing the rule of ten events per variable in logistic and cox regression. *American Journal of Epidemiology*, 165(6):710–718, 2007.

G. Wang, W. Li, M. A Zuluaga, R. Pratt, P. A. Patel, M. Aertsen, T. Doel, A. L. David, J. Deprest, S. Ourselin, et al. Interactive medical image segmentation using deep learning with image-specific fine tuning. *IEEE Transactions on Medical Imaging*, 37(7):1562–1573, 2018.

David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

Di Wu, Xin Luo, Yi He, and Mengchu Zhou. A prediction-sampling-based multilayer-structured latent factor model for accurate representation to high-dimensional and sparse data. *IEEE Transactions on Neural Networks and Learning Systems*, 35(3):3845–3858, 2022.

Rui Xin, Chudi Zhong, Zhi Chen, Takuya Takagi, Margo Seltzer, and Cynthia Rudin. Exploring the whole rashomon set of sparse decision trees. *Advances in neural information processing systems*, 35:14071–14084, 2022.

M. Xiong, X. Fang, and J. Zhao. Biomarker identification by feature wrappers. *Genome Research*, 11(11): 1878–1887, 2001.

Bin Yu and Karl Kumbier. Veridical data science. *Proceedings of the National Academy of Sciences*, 117(8): 3920–3929, 2020. doi: 10.1073/pnas.1901326117. URL `https://www.pnas.org/doi/abs/10.1073/pnas.1901326117`.

Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *Annals of Statistics*, 38 (2):894–942, 2010.

Tao Zhang, Liqin Lin, Jing Tian, Zaifa Xue, and Xiaonan Guo. Voice feature description of parkinson's disease based on co-occurrence direction attribute topology. *Engineering Applications of Artificial Intelligence*, 122:106097, 2023.

Chudi Zhong, Zhi Chen, Jiachang Liu, Margo Seltzer, and Cynthia Rudin. Exploring and interacting with the set of good sparse generalized additive models. *Advances in neural information processing systems*, 36, 2024.

A Comparison to Single-Model Variable Selection Framework

We firstly give a brief overview of variable selection methods that are relevant to this work, and later provide some comparisons between them and the SWAG framework. Therefore, let us start by recalling the main strategies in forward or backward selection as commonly applied to linear and logistic regression model classes. Probably the most well-known procedure, and still one of the most employed by practitioners, is plain *forward selection*. It starts from an empty model and iteratively adds the variable that results in the greatest improvement of a criterion e.g., the *Akaike Information Criterion* (AIC; Akaike, 1973) or the *Bayesian Information Criterion* (BIC; Schwarz, 1978). Historically, the roots of stepwise selection can be traced to Efroymson (1960), and various stepwise approaches remain standard for their simplicity, although they usually are not particularly suited for high-dimensional problems. To tackle this setting, another group of methods leverages *screening* which can then be combined with forward selection or any other feature selection procedure (similarly to SWAG). For example, in ultrahigh-dimensional settings, *Sure Independent Screening* (SIS) proposed by Fan & Lv (2008) and Fan et al. (2009) ranks variables by a measure of marginal association, drastically reducing the dimensionality. Its iterative counterpart, ISIS, refines this process by allowing repeated screening steps combined with penalized methods such as Lasso (Tibshirani, 1996), SCAD (Fan & Li, 2001), or MCP (Zhang, 2010). A further popular method is *Recursive Feature Elimination* (RFE), introduced by Guyon et al. (2002). RFE begins with the full set of variables and then: (i) trains a model within the class of choice e.g., Random Forests, (ii) computes variable-importance metrics, and (iii) eliminates the least important variables, iterating until a specified subset size is reached.

All the above methods (and, to the best of our knowledge, the vast majority of existing forward/backward and penalized variable selection algorithms) aim to find a single, final subset of features i.e., a single model. With SWAG, we build on the underlying idea of forward selection (i.e. growing from smaller to larger subsets of variables) but adapt it to identify multiple sparse and accurate models rather than a single best model. In essence, similarly to the MPS of Kissel & Mentch (2024), SWAG is a model-agnostic wrapper that uses any estimator of the test error metric, e.g. cross-validation (CV), to guide each enlargement of the variable set. It screens variables one at a time initially, then expands only those subsets that have already demonstrated strong predictive performance, and it does so for any chosen model class, whether linear/logistic regression, kernel SVMs, Random Forests or possibly even more complex learners. Moreover, one of the main reasons for running SWAG is to study the so-called Rashomon Effect (Breiman, 2001b; Fisher et al., 2019) where various models can deliver similarly good performance on a given dataset. Instead of discarding the "almost-as-good" alternatives, the SWAG keeps them therefore letting users explore the diversity of good solutions, check for potential sparse latent structures and build variable-importance networks to interpret effects. In comparing SWAG with existing methods, one can note that it preserves the forward-growing nature of classic stepwise selection but goes beyond it by exploring multiple subsets in parallel at each dimension rather than following a single path. This allows SWAG to present a library of equally good models rather than a single final model. Although nothing impedes from modifying SWAG in various ways, currently it does not rely only on marginal metrics for screening (differently from screening-based methods such as SIS and ISIS) but directly uses the test error ϵ which, after the first step of the procedure, accounts for conditional effects given other variables. Compared to RFE, SWAG pursues a forward approach rather than a backward one, which tends to be more computationally tractable when the number of variables is large. Also, RFE methods typically require a sensitivity metric specific to the chosen class of models and still aim for a single best subset of variables, whereas SWAG by construction accommodates different classes of models and retains many good subsets of variables. Lastly, it is interesting to compare SWAG with a variable selection method that also delivers a set of models i.e., the *random subspace method* (RSM) of Ho (1998). RSM is an *ensemble* approach in which variable subsets are chosen at random and used to train weak models. On the other hand, SWAG focuses on identifying good (strong) sparse models through a deliberate search guided by the error metric ϵ. Because RSM is inherently random, interpreting variable importance can be more challenging, whereas SWAG keeps track of which variables produce good performance and thus highlights how features co-occur across good models. Of course, this comes at a computational cost as SWAG by construction, is more computationally intensive than the classical RSM. A recent alternative very similar in spirit to the SWAG is the MPS of Kissel & Mentch (2024) which builds "forking-paths" from each of the pre-screened variables and evaluates (in a forward stepwise manner) which variables should be added to each of the previously selected variables.

By using a stability selection procedure in each step, the MPS builds a set of models of size $p_{\max}$. Compared to the latter, the SWAG is more computationally intensive but is less greedy in terms of nested structures since it allows to drop certain previous variable combinations if they do not improve the error metric in the following dimensions. Moreover, it does not only consider models of dimension $p_{\max}$ but also all smaller models with similarly good performance (although we think that the MPS could theoretically be adapted to consider this case as well). Based on these characteristics, while being more computationally burdensome, the SWAG would intuitively provide more diverse models in terms of size and variable combinations which can be an important aspect for practitioners.

To provide a practical comparison, we run some common variable selection methods (that only choose one model) on the same datasets and with the same model classes as those shown in Sec. 3. More in detail, in Tab. 3 we compare SWAG with: (i) AIC forward selection combined with the SIS screening procedure (*SIS + forwardAIC*); (ii) the iterative SIS coupled with the default MCP penalization method (*ISIS MCP*); and (iii) the RFE with Random Forests as the model class (*RFE RF*). These competitors include selection approaches of different nature, including penalized-based methods (e.g. Lasso, ISIS) and permutation importance based methods (RFE based on RF with MDA measures). We underline that we coupled classic forward selection with a screening procedure because in itself it is not suited for the high-dimensional datasets used in our work. This is because running a logistic regression requires a number of observations larger or equal to the number of variables, which is not the case for all our datasets. The main result of the applied analysis (see Tab. 3) is that, for all the datasets considered, the test error ϵ_{test} of each competitor lies within the test error range of the SWAG models (or even above the maximum value in some cases) for the relevant model class i.e., Lasso-logistic for *SIS + forwardAIC* and *ISIS MCP* and Random Forests (RF) for *RFE RF*. These results provide further evidence that SWAG searches the space of models in a comparable way to other forward/backward feature selection methods but, as opposed to the latter, outputs a library of almost-equally-good sparse models rather than a single one. This result also complements the one presented in the paper (see Fig. 1) where the test error of each full model, trained with all the variables, is always either within the SWAG range or sometimes even higher than the worst test error achieved by a SWAG model (except for one case). Throughout this analysis, we have made an effort to perform the most fair comparison possible. In particular, we have used 10-fold CV as the estimator of the out-of-sample prediction error whenever possible and we input our choice of $p_{\max}$ to *RFE RF* for each dataset to both speed-up and facilitate the search in the space of models of the considered comparisons.

	ϵ_{test} MeterA	ϵ_{test} LSVT	ϵ_{test} Ahus	ϵ_{test} Colon	ϵ_{test} Leukemia
SWAG range Lasso	**[0.000, 0.000]**	**[0.192, 0.231]**	**[0.065, 0.226]**	**[0.167, 0.417]**	**[0.000, 0.286]**
SIS + forwardAIC	0.111	0.192	0.129	0.417	0.143
ISIS MCP	0.333	0.231	0.065	0.250	0.143
SWAG range RF	**[0.000, 0.278]**	**[0.115, 0.269]**	**[0.032, 0.194]**	**[0.333, 0.417]**	**[0.071, 0.214]**
RFE RF	0.000	0.154	0.129	0.417	0.071

Table 3: Comparison of the test errors ϵ_{test} of forward/backward feature selection methods with the SWAG models test error range for the relevant model class i.e., Lasso-logistic for *SIS + forwardAIC* and *ISIS MCP* and Random Forests (RF) for *RFE RF*. The analysis has been done for all the datasets present in our work: MeterA, LSVT, Ahus, Colon and Leukemia.

To summarize, SWAG brings two main benefits compared to existing variable selection methods: (i) by using any test error metric, it can wrap around any model class without requiring specialized ranking metrics (each tied to a specific class of models); (ii) it finds multiple and *diverse* almost-equally-good (distinct) solutions at each subset size to address the Rashomon Effect. In terms of pros, empirical performance this far shows that

SWAG can match or exceed the accuracy of a (regularized) model fit on the whole data while selecting very few variables, and it allows for new avenues of interpretation (e.g. through a network of co-occurrences among variables in the best subsets). On the cons side, SWAG's forward-greedy expansion can be computationally demanding if the number of variables is extremely large or if the error metric ϵ is burdensome to compute, although parallelization can mitigate these burdens together with a smarter choice of the error metric if available (e.g. AIC or BIC). In this perspective, since the parameter $p_{\max}$ can work as a regularization parameter, it may be possible to simply evaluate the models based on their training error thereby avoiding major computational costs coming from cross-validations (to be investigated in future work). Moreover, it is still not known if and under what conditions the SWAG guarantees finding *every possible* almost-equally good model in the considered dimensions, though in practice it has proven to robustly capture diverse and accurate solutions for different model classes and datasets. To conclude, similarly to the MPS in Kissel & Mentch (2024), the SWAG fits into the forward selection tradition by however extending its scope: instead of identifying a single final subset it finds multiple sparse subsets with good performance and delivering advantages common when taking into account the Rashomon Effect.

B Computational Complexity

We firstly underline that the premise of the SWAG is to deliver a set of good sparse models for high-dimensional problems to be employed at the user's discretion for interpretations, predictions and decisions. Therefore it does not consider problems where the user needs to rapidly re-estimate or re-test new models and consequently assumes that the user is not constrained by time to obtain the SWAG outputs. This being said, the SWAG is a wrapper algorithm and therefore part of its complexity depends on the complexity in fitting the model class run within it. For this reason, let $C(n, \hat{p})$ represent the complexity of fitting a model $l \in \mathcal{L}$ to a dataset with n samples and $\hat{p}$ variables. Based on this, the first step of the SWAG evaluates each variable individually, therefore the complexity of this first step is $O\left(p\, C(n, 1)\right)$. The general step iteratively fits models with $2 \leq \hat{p} \leq p_{\max}$ variables where, for each dimension $\hat{p}$, the number of times the model is fitted is m with consequent complexity $O\left(m\, C(n, \hat{p})\right)$. Since the latter step iterates up to $p_{\max}$, the worst-case complexity of the general step is $O\left((p_{\max} - 1)\, m\, C(n, p_{\max})\right) = O\left(p_{\max}\, m\, C(n, p_{\max})\right)$. Therefore, combining the two steps of the SWAG delivers an overall worst-case complexity of $O\left(p\, C(n, 1) + p_{\max}\, m\, C(n, p_{\max})\right)$. In addition, if one uses r-repeated k-fold cross-validation to obtain the test error metric ϵ, then this worst-case complexity grows to $O\big(r\, k\, \big(p\, C(n, 1) + p_{\max}\, m\, C(n, p_{\max})\big)\big)$. As a consequence, since p is fixed by the data and $p_{\max}$ is fixed to be small by the user for sparsity/interpretation reasons, the computational complexity is driven mainly by the choice of m (i.e. the number of models to evaluate at each step) and, in particular, by the test error metric chosen. Indeed, the overall complexity of the SWAG can be increased $(r\, k)$-fold if choosing r-repeated k-fold cross-validation but can be considerably reduced if one is able to employ other more efficient metrics. As an example, if using a generalized linear model with complexity $O(n\hat{p}^2 + \hat{p}^3)$ and using AIC or BIC as the error metric ϵ, then the worst-case complexity becomes $O\left(np + mnp_{\max}^3 + mp_{\max}^4\right)$: this obviously consists in a considerable decrease compared to the general worst-case complexity, as also shown the computational runtimes further on. One could even further reduce computational complexity by directly using training error metrics since the parameter $p_{\max}$ could already be considered as a regularizer to limit over-fitting (see Efron et al., 2004, and discussions in Sec. 2.1). In addition to m however, in specific cases the parameter α can also play a role for complexity with respect to the number of variables p. Indeed, α determines the set of screened variables $\mathcal{S}^*$ whose size is p^*. In the general step the SWAG samples m possible combinations out of the $\binom{p^*}{\hat{p}}$ available: this total number of available combinations may be smaller than m due to a potentially small p^*, in which case the SWAG performs an exhaustive search and evaluation of all possible variable combinations for dimension $\hat{p}$ (i.e. best subset selection). If this occurs, then this will be the case also in larger dimensions up to $p_{\max}$ where the number of possible combinations will actually decrease with the dimension. As a result, the complexity coming from m will be reduced to $O\big(\binom{p^*}{\hat{p}}\big)$ for all $\{2 \leq \hat{p} \leq p_{\max} \mid \binom{p^*}{\hat{p}} < m\}$.

B.1 Example Runtime: Leukemia Dataset

We ran the SWAG on an Apple 8-core M1 pro 16GB RAM on the Leukemia dataset which has $n = 58$ training samples and $p = 7129$ variables with $m = 10160$, $\alpha = 0.01$ and $p_{\max} = 4$. Using 10-repeated 10-fold cross-validation to get the error metric ϵ, we get the results in Tab. 4 (rounded to the closest minute). These times will obviously be longer when the number of samples n increases, especially when employing model classes that do not scale too well with n. These runtimes can however be significantly reduced through vectorized operations and, in particular, parallelization (when possible) since we can fit and evaluate multiple models for each dimension $\hat{p}$ at the same time.

Lasso	L-SVM	R-SVM	RF
50 min	22 min	153 min	82 min

Table 4: Runtimes of SWAG on Leukemia dataset for different model classes.

Above we discussed how the use of a more computationally efficient error metric can drastically reduce SWAG runtimes. Let us therefore see an example when using a simple *logistic regression* in the SWAG and using the AIC as the error metric ϵ. Running it on a MacBook Pro with 1.4 GHz Quad-Core Intel Core i5 processor and 8 GB RAM, the SWAG is run in **0.32 min (roughly 20 seconds)**. Moreover, as a side note, this does not appear to affect the test error performance of the resulting SWAG models compared to the model classes and cross-validation approaches used for the other methods. In fact, the range of test errors for the SWAG models using this approach is $[0, 0.214]$ and the model dimension range is $[2, 4]$ therefore delivering results in line with the other model classes for this dataset. This suggests that if an efficient estimator of the test error is available, runtimes can be considerably reduced without significantly affecting the results.

C Hyper-parameter Choice

In this work, we make use of different model classes that we use either in their original versions or within the SWAG. For this reason, we make use of the same hyper-parameter selection rules in all settings, relying on the common/default choices (as per the `caret` R package):

1. **Lasso-logistic**[2]: the data is pre-processed (i.e., centered and rescaled) and a grid of 5 penalty terms (λ) for the Lasso (logistic) is used with $\lambda \in [0, \lambda_{max}]$ where $\lambda_{max} = \{0.05, 0.1\}$ depending on $p_{\max}$.

2. **Linear-** and **Radial-Kernel SVM**: in both cases, the data is pre-processed (i.e., centered and rescaled) and the Linear-Kernel SVM has a cost parameter $c = 1$ (default value), while the cost for Radial-Kernel SVM is automatically selected based on a grid of 5 values (i.e., `tuneLength` $= 5$);

3. **Random Forest**: the default value for the number of trees is kept (i.e., `ntree` $= 500$) as well as the common choice for the number of randomly sampled variables at each split (i.e., `mtry`$= \sqrt{p}$ for the original version and `mtry` $= \sqrt{\hat{p}}$ for the SWAG; see Genuer et al. (2010) for a detailed discussion.

D Sensitivity Analysis

We briefly discuss how the SWAG compares to Rashomon sets in the choice of their respective parameters (which was also mentioned in Sec. 2.2). Indeed, recalling the definition of Rashomon sets, the "performance" of the Rashomon set depends on the choice of a (good) reference model l^* and the parameter θ which may not always be easy to determine according to the choice of the loss function D (e.g. it may be difficult to have *a priori* knowledge of the potential range of loss values for a specific problem). Similarly, the performance of the Rashomon set can vary depending on choice of the model class $\mathcal{L}$ and across different datasets. Therefore, similarly to Rashomon sets, the choice of the meta-parameters is up to the user's discretion based on their needs. In particular, given that the parameter $p_{\max}$ is not chosen for performance but for user-preference on

[2]In this particular case, Algo. 1 is implemented with the `caret` package using classical (non-penalized) logistic regression

model dimensions, the main parameters to be chosen are m and α (and eventually post-processing parameters, e.g. δ). The parameter α plays a similar role to the parameter θ in Rashomon sets, except that it is not an absolute value but relative to the set of m models evaluated in a certain dimension. In some way, the choice of α could be considered more intuitive with respect to θ since the former represents the proportion of best performing models we want to use in the algorithm, whereas the choice of θ relies on the possible (unknown) range of losses with respect to a model class and dataset. Therefore, the "additional" parameter with respect to Rashomon sets would be the number of models to evaluate in each step of the SWAG, i.e. m.

With the above in mind, we perform some sensitivity analyses to understand how these parameters can affect the SWAG results in terms of test error (ϵ_{test}) and SWAG model dimensions ($|s_l|$). Given that $p_{\max}$ is determined by the user based on their interpretability requirements, as mentioned earlier, we focus on the impact of the parameters m (number of models considered for each dimension) and α (the percentile to determine the selected models for each dimension). We firstly run the SWAG for different model classes and different combinations of meta-parameters α and m for the Leukemia dataset: the results are presented in Tab. 5 and 6. We see that as the parameter α increases, so does the average test-error and its upper range, which is to be expected: by increasing the proportion of selected models we allow "worse-performing" models to be considered in the SWAG therefore increasing the upper bound (just like increasing θ for Rashomon sets). Across different values of m we generally also see an increase in the upper range of test errors as we increase m: nevertheless for a fixed value of α, the choice of m does not appear to considerably affect the range of SWAG model dimensions $|s_l|$ or the average test-error performance $mean(\epsilon_{test})$. Indeed, as we would expect, increasing m allows to explore more variable combinations and, by doing so, identifies more well-performing models as well as more "worse-performing" ones. Similar conclusions are made when focusing on the Meter A data using the Linear-Kernel SVM: see Tab. 7. All things being equal, these intuitive results suggest to pick a large m (conditional on computational time and resources) and then let the user pick the best performing ones within the final SWAG library. Nevertheless, as seen in these sensitivity results, similarly good-performing models are found even with smaller values of m. In fact, the issue of consistent performance of the SWAG in some way relies mainly on whether the SWAG is able to find some good performing models independently from these parameters, and the means and ranges of test errors across all model classes suggest that this is generally the case.

In support of the generally consistent performance across model classes and datasets, we extracted information on the frequency of variable selection (variable importance) across different classes within the same datasets analyzed in this work and looked at the correlation between these frequencies. Tab. 8 and 9 show these analyses for the Colon and Leukemia datasets respectively. For the Colon dataset, we see a high degree of correlation between Lasso and L-SVM which build linear decision boundaries while there is a less strong (but still reasonable) degree of correlation between R-SVM and RF which produce non-linear decision boundaries (whereas the correlations between these two groups of model classes are very low). We notice a similar pattern for the Leukemia dataset where however there is also a high degree of correlation of the R-SVM with the linear boundary models and a lower degree of correlation (but still non-negligible) with the non-linear model RF. What this seems to suggest is that, conditional on different non-linear patterns in the data, the identification of important predictive variables by the SWAG is relatively consistent across model classes (especially if these classes share some underlying assumptions): they select the same variables and consider the same variables to be important. Since these results are based on variable importance, there is a good reason to believe that these results would hold even for different choices of α and m since they do not greatly affect the detection of important variables once they are screened in the first step. In conclusion, returning to the choice of the SWAG meta-parameters, knowing that $p_{\max}$ is chosen by the user for sparsity preferences (and not for performance), α should generally be chosen small enough (depending also on the number of variables p) and m can be made as large as computational power allows: then the user can post-process the SWAG models as strictly as they want to pick the best performing ones. As stated earlier, similar reasoning must be applied, for example, also for the choice of the meta-parameters of Rashomon sets, i.e. reference model l^* and parameter θ, where the reference model is usually a pre-existing empirical risk minimizer chosen by the user.

	$m = 5,000$			$m = 10,000$		
	$\alpha = 0.1$	$\alpha = 0.05$	$\alpha = 0.01$	$\alpha = 0.1$	$\alpha = 0.05$	$\alpha = 0.01$
Lasso-Logistic						
ϵ_{test}	$[0, 0.286]$	$[0, 0.286]$	$[0, 0.286]$	$[0, 0.357]$	$[0, 0.357]$	$[0, 0.286]$
$\text{mean}(\epsilon_{\text{test}})$	0.134	0.109	0.059	0.110	0.100	0.075
$\lvert s_l \rvert$	$[2, 4]$	$[2, 4]$	$[2, 4]$	$[2, 4]$	$[2, 4]$	$[2, 4]$
Linear SVM						
ϵ_{test}	$[0, 0.357]$	$[0, 0.214]$	$[0, 0.214]$	$[0, 0.357]$	$[0, 0.357]$	$[0, 0.142$
$\text{mean}(\epsilon_{\text{test}})$	0.092	0.075	0.063	0.115	0.083	0.063
$\lvert s_l \rvert$	$[2, 4]$	$[2, 4]$	$[2, 4]$	$[2, 4]$	$[2, 4]$	$[2, 4]$
Radial SVM						
ϵ_{test}	$[0, 0.286]$	$[0, 0.286]$	$[0, 0.214]$	$[0, 0.286]$	$[0, 0.286]$	$[0, 0.214]$
$\text{mean}(\epsilon_{\text{test}})$	0.113	0.085	0.081	0.101	0.084	0.081
$\lvert s_l \rvert$	$[2, 4]$	$[2, 4]$	$[2, 4]$	$[2, 4]$	$[2, 4]$	$[2, 4]$
RF						
ϵ_{test}	$[0.071, 0.286]$	$[0.143, 0.286]$	$[0.143, 0.214]$	$[0, 0.286]$	$[0.143, 0.214]$	$[0.071, 0.214]$
$\text{mean}(\epsilon_{\text{test}})$	0.216	0.188	0.182	0.175	0.203	0.146
$\lvert s_l \rvert$	$[2, 4]$	$[2, 4]$	$[2, 4]$	$[2, 4]$	$[2, 4]$	$[2, 4]$

Table 5: Sensitivity analysis of meta-parameters m and α for the Leukemia dataset

Logistic						
	$m = 5,000$			$m = 10,000$		
	$\alpha = 0.1$	$\alpha = 0.05$	$\alpha = 0.01$	$\alpha = 0.1$	$\alpha = 0.05$	$\alpha = 0.01$
ϵ_{test}	$[0, 0.42857]$	$[0, 0.28571]$	$[0, 0.21429]$	$[0, 0.5]$	$[0,\ 0.35714]$	$[0, 0.21429]$
$\text{mean}(\epsilon_{\text{test}})$	0.12519	0.09505	0.06505	0.12548	0.10005	0.06505
$\lvert s_l \rvert$	$[2, 4]$	$[2, 4]$	$[2, 4]$	$[2, 4]$	$[2, 4]$	$[2, 4]$

Table 6: Sensitivity analysis of meta-parameters m and α for the Leukemia dataset with simple logistic and AIC for test error estimation

Table 7: Sensitivity analysis of meta-parameters m and α for the Meter A dataset with Linear-Kernel SVM.

	$m = 10,000$			$m = 40,000$		
	$\alpha = 0.2$	$\alpha = 0.1$	$\alpha = 0.05$	$\alpha = 0.2$	$\alpha = 0.1$	$\alpha = 0.05$
ϵ_{test}	$[0.056, 0.500]$	$[0, 0.111]$	$[0, 0.056]$	$[0.056, 0.722]$	$[0, 0.667]$	$[0, 0.056]$
$\lvert s_l \rvert$	$[5, 6]$	$[2, 6]$	$[2, 6]$	$[4, 6]$	$[2, 6]$	$[2, 6]$

Table 8: Colon dataset: Correlation Matrix of Variable Importance

	Lasso	L-SVM	R-SVM	RF
Lasso	1.00000000	0.74568151	-0.05654834	-0.05374952
L-SVM	0.74568151	1.00000000	-0.03513071	-0.06136117
R-SVM	-0.05654834	-0.03513071	1.00000000	0.28045757
RF	-0.05374952	-0.06136117	0.28045757	1.00000000

Table 9: Leukemia dataset: Correlation Matrix of Variable Importance

	Lasso	L-SVM	R-SVM	RF
Lasso	1.00000000	0.89133907	0.7865345	-0.04117896
L-SVM	0.89133907	1.00000000	0.8179987	-0.01653499
R-SVM	0.78653446	0.81799867	1.0000000	0.13001335
RF	-0.04117896	-0.01653499	0.1300133	1.00000000

E Empirical Results

Here we give further details regarding the empirical results presented in Sec. 3. More specifically, we provide (i) more details regarding the logic used to determine the SWAG meta-parameters, (ii) the training error (ϵ_{train}) results, number models and Jaccard index description for the considered datasets, and (iii) the details regarding the variables included in the SWAG networks for the Meter A and LSVT data.

E.1 SWAG Parameter Choice

The choice of the SWAG meta-learning parameters is made in line with the rules of thumb described in Sec. 2.1 and with the intent of reducing overall computational time while ensuring a reasonable exploration of the variable space. More specifically, we start by defining $p_{\max}$ based on the EPV rule discussed in van der Ploeg et al. (2014) and, for each dataset, we verify that the selected $p_{\max}$ reaches at least an EPV > 4 as suggested by Vittinghoff & McCulloch (2007). Based on this, we define m with the aim to at least explore all the subspace of two-dimensional models generated by $p^\star$ (i.e., $\binom{p^\star}{2}$). To determine this value in practice, we first evaluate $\binom{p}{2}$ and, given the computational power at our disposal, we select a given percentage of this quantity that we may call m_2 since it refers only to two-dimensional models. This choice then automatically determines a value $p^\star$ that satisfies our original aim since $p^\star = {}^{1+\sqrt{1+8m_2}}/_{2}$. The value of α follows automatically and it is set to $\alpha = {}^{p^\star}/_{p}$ or, if this value is not appropriate for the user, the parameter m_2 can be varied in order to obtain the desired value of α. In particular, we find that the range $0.01 \leq \alpha \leq 0.05$ is a good compromise between fixing an α as small as possible (in order to select strong models) and the possibility of exploring a considerable portion of the variable space. Taking this into account, in order to determine the final value of m, we just need to consider the growing dimension of the model up to $p_{\max}$. To do so, we multiply the evaluation made at the two-dimensional space of models by $p_{\max}$ and obtain our final $m = m_2\, p_{\max}$. It implies that, following our choices, m is a linear function of $p_{\max}$.

E.2 SWAG Training Error

For additional information purposes, we provide results on the training error (ϵ_{train}) observed for the SWAG models and their original versions which are represented in Fig. 4. As can be observed, the training error ranges of the SWAG models are generally lower than that of the original model versions. We expect this since the SWAG models are selected based exactly on this cross-validation criterion (which is not the case for the original versions).

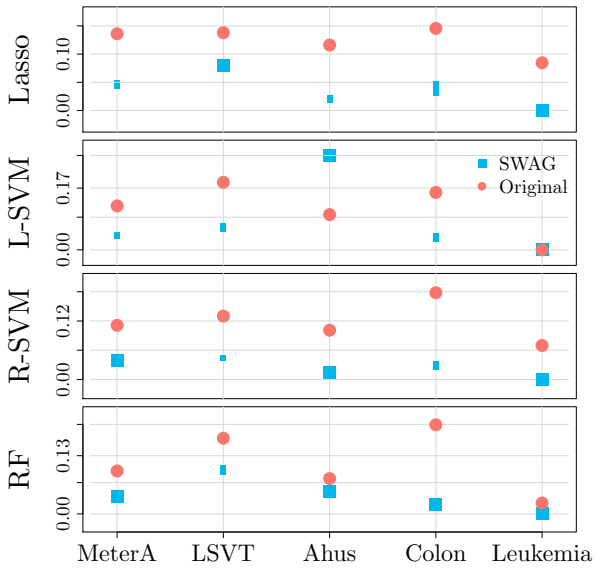


Figure 4: SWAG training error range ϵ_{train} (blue rectangles) and original learning method training error (red dots). The blue *squares* imply that the SWAG models all have the same training error.

E.3 Number of SWAG Models

Tab. 10 collects the information on the number of SWAG models $|\widetilde{\mathcal{M}}|$ per model class and dataset shown in this work (see Tab. 1). As we can see, the number of models varies between datasets and model classes, indicating whether there can exist sparse models as a result of the Rashomon Effect. For example, a number that stands out is the number of models based on the Lasso class, i.e. 5, which can suggest that models belonging to its Rashomon set are of larger dimensions than $p_{\max}$. This is consistent with Fig. 1 where we can see that the original model performs better than the corresponding SWAG models.

Table 10: Number of models in the SWAG libraries

	MeterA	LSVT	Ahus	Colon	Leukemia
Lasso	32	5	720	64	157
L-SVM	74	417	447	66	98
R-SVM	26	28	1'137	70	40
RF	26	27	637	59	20

E.4 Jaccard Index Computation

We briefly explain how the Jaccard index was computed to present the results in Tab. 2. Generally speaking, we pick two models belonging to the SWAG library and then apply the Jaccard index to the sets of variables included in these two models: we then do so for all pairs of SWAG models. More specifically, for sets A and B the Jaccard index is given by:

$$J(A, B) = \frac{A \cap B}{A \cup B}.$$

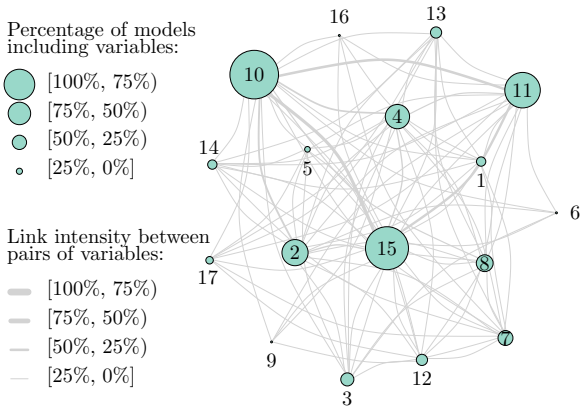
Therefore, as an example, if $A = \{2, 3\}$ is the index set of variables in one SWAG model (i.e. this model contains variables 2 and 3) and if $B = \{2, 3, 12\}$ is the index set of variables in another SWAG model, then the Jaccard index is $^2/_3 \approx 0.67$ denoting a considerable overlap in variables between the two models. This is computed for each pair of models in the SWAG sets therefore generating a collection of Jaccard indexes which we summarize with their median value and range for each dataset and model class.

E.5 SWAG Network Details

In this section, we reproduce the same analyses and interpretations included in the main manuscript (for ease of readability) and add the details regarding the variables included in the SWAG networks for the Meter A and LSVT datasets.

E.5.1 Meter A

The Meter A data is analyzed in Gyamfi et al. (2018) and collects measurements on ultrasonic flowmeter diagnostics. Achieving good diagnostics regarding the health of a flowmeter is of extreme importance for condition-based maintenance in many industrial sectors such as the oil and gas industry since incorrect measurement can entail considerable economic and material losses (see, e.g., TUV-NEL, 2012). In this data, the goal is to classify the health of a meter into two classes: "Healthy" (Class 1) or "Installation effects" (Class 2). Given that the variables are measurements of physical nature, we decide to consider all first-order interactions which finally delivers a total variable size of $p = 666$ (36 original variables plus $\binom{36}{2} = 630$ interactions). Using the results from Lasso-based SWAG, we can see how the variables most frequently included in the selected models can be arranged into a network where the edges represent the most common connections between these variables as can be seen in Fig. 5. Therefore, in order to understand the mechanics and perform diagnostics for this ultrasonic flowmeter, among the 666 variables, a researcher could for example focus their attention on the interaction (i) between flatness ratio and gain as well as (ii) between the speed of sound and gain at the first end (of the fifth path). In particular, a consistent flatness ratio across various paths indicates structural integrity and alignment, crucial for optimal performance. Variations in the flatness ratio can suggest issues such as misalignment or internal obstructions, which can significantly impair the gain readings and, consequently, the flow measurements. The flatness ratio can affect wave propagation, with deviations leading to distorted signals. This distortion, which manifests in reduced gain measurements, indicates latent defects that could progress into more severe operational failures if not addressed (Tang et al., 2015). In addition, changes in the speed of sound can indicate alterations in fluid properties—such as temperature, pressure, or contamination levels (which may not be immediately observable). Monitoring the speed of sound alongside gain measurements can uncover underlying problems within the fluid system that affect measurement accuracy (Bombarda et al., 2021). As a final note, given that cost-based maintenance can have asymmetric costs according to the decision taken on the flowmeter, the SWAG could allow to select models based on the corresponding (non-convex) cost-function instead of a symmetric kind of loss (i.e., each type of error is weighted equally) that model classes are usually trained on (see, e.g., Crone, 2002; Masnadi-Shirazi & Vasconcelos, 2007).

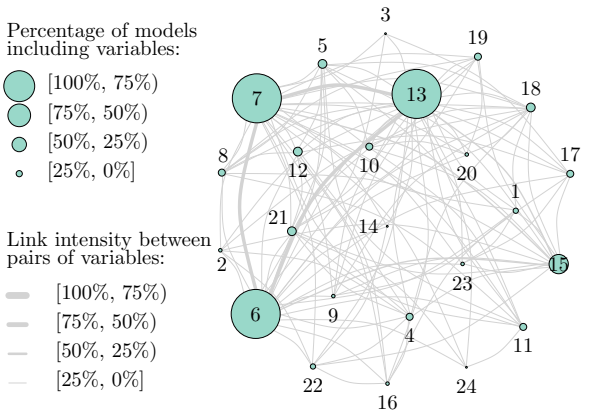


	Meter A variables
1	Gain at first end of 2nd path
2	Gain at first end of 5th path
3	Int. Flatness ratio, Gain at first end of 1st path
4	Int. Flatness ratio, Gain at first end of 7th path
5	Int. Flatness ratio, Gain at second end of 2st path
6	Int. Flatness ratio, Gain at second end of 4th path
7	Int. Flatness ratio, Gain at second end of 7th path
8	Int. Speed of sound 3nd path, Gain at first end of 5th path
9	Int. Speed of sound 3rd path, Gain at first end of 5th path
10	Int. Speed of sound 4th path, Gain at first end of 5th path
11	Int. Speed of sound 6th path, Gain at first end of 5th path
12	Int. Speed of sound 7th path, Gain at first end of 5th path
13	Int. Speed of sound 8th path, Gain at first end of 5th path
14	Int. Average speed of sound in all paths, Gain at first end of 5th path
15	Int. Gain at first end of 2nd path, Gain at first end of 5th path
16	Int. Gain at first end of 5th path, Gain at second end of 5rd path
17	Int. Gain at first end of 5th path, Gain at second end of 5th path

Figure 5: Network of variable importance and pairwise link-intensity in the set of models $\widetilde{\mathcal{M}}$ for the Meter A dataset using the Lasso-based SWAG results. "Int." denotes the interaction between two variables

E.5.2 LSVT

The voice rehabilitation dataset is analyzed in Tsanas et al. (2013) in order to assess the effectiveness of a computer program called "Lee Silverman voice treatment (LSVT) Companion" which allows patients with Parkinson's disease to independently progress through a rehabilitative treatment session. Taking data on 126 samples from 14 patients who followed the latter treatment, 310 dysphonia measures are taken on each of them (plus information on sex and age of the patients) and used to understand if they could correctly predict whether the patients' voices are "acceptable" or "unacceptable" after this treatment. In their analysis, a robust feature selection is used to select 8 variables (based on the first eight variables classified by the feature selection method) and subsequently R-SVM is tested (along with RF) to obtain around 90% accuracy in classifying patients' progress.



	LSVT variables		LSVT variables
1	Jitter F0 abs 0th perturb	13	entropy log 4 coef
2	Jitter F0 TKEO prc5	14	entropy log 6 coef
3	Jitter F0 TKEO prc25	15	entropy log 6 coef
4	Jitter F0 range 5 95 perc	16	entropy log 10 coef
5	Jitter pitch percent	17	det TKEO mean 6 coef
6	MFCC 2nd coef	18	det TKEO mean 8 coef
7	MFCC 3rd coef	19	det TKEO std 5 coef
8	prc95 5 F0 series F0 expTitze	20	det TKEO std 6 coef
9	entropy shannon 6 coef	21	det TKEO std 7 coef
10	entropy shannon 7 coef	22	det TKEO std 8 coef
11	entropy shannon 9 coef	23	det TKEO std 10 coef
12	entropy shannon 10 coef	24	entropy log 3 9 coef

Figure 6: Network of variable importance and pairwise link-intensity in the set of models $\widetilde{\mathcal{M}}$ for the LSVT dataset using the Radial-Kernel SVM SWAG results.

There is also scientific interest in determining the variables (and combinations thereof) that most contribute to the definition, in this case, of a Parkinson's disease (PD) speech treatment as being acceptable or not. Also in this case, the SWAG models (based on the R-SVM) can be arranged into a network in order to allow for interpretation as seen in Fig. 6. Based on the SWAG network, researchers interested in improving speech treatment should focus on the 2nd and 3rd Mel-Frequency Cepstral coefficients (MFCC) and on the entropy with base-4 logarithmic coefficients (as well as the interactions between these three variables as highlighted by their frequent presence in the same SWAG models). In particular, MFCCs such as the 2nd and 3rd coefficients, are pivotal in characterizing the spectral properties of sound produced by the vocal tract. They provide details about vowel sounds and overall speech quality, making them useful for assessing the impact of LSVT therapy. Moreovwer, various jitter measurements (such as Jitter F0 absolute perturbation and Jitter F0 range across the 5th and 95th percentiles), are crucial in assessing vocal stability: with heightened jitter often correlating with less stable vocal production where this instability can signal underlying physiological issues affecting vocal fold function, suggesting a need for targeted therapeutic interventions. For instance, a patient exhibiting low jitter values and consistent MFCC readings alongside stable entropy measures is likely progressing well in their treatment. Conversely, high jitter or low entropy may indicate challenges that require further intervention. These attributes serve not only as indicators of progress but as quantifiable measures for tailoring personalized therapy sessions, enhancing the overall management of voice characteristics affected by PD. Each attribute interrelates, influencing the others: for example, stabilization in jitter can lead to improved MFCC values as patients develop stronger vocal control through therapy.