Composite Feature Selection Using Deep Ensembles

Fergus Imrie*
University of California, Los Angeles
imrie@ucla.edu

Alexander Norcliffe*
University of Cambridge
alin2@cam.ac.uk

Pietro Liò
University of Cambridge
pl219@cam.ac.uk

Mihaela van der Schaar
University of Cambridge
The Alan Turing Institute
University of California, Los Angeles
mv472@cam.ac.uk

Abstract

In many real world problems, features do not act alone but in combination with each other. For example, in genomics, diseases might not be caused by any single mutation but require the presence of multiple mutations. Prior work on feature selection either seeks to identify individual features or can only determine relevant groups from a predefined set. We investigate the problem of discovering groups of predictive features without predefined grouping. To do so, we define predictive groups in terms of linear and non-linear interactions between features. We introduce a novel deep learning architecture that uses an ensemble of feature selection models to find predictive groups, without requiring candidate groups to be provided. The selected groups are sparse and exhibit minimum overlap. Furthermore, we propose a new metric to measure similarity between discovered groups and the ground truth. We demonstrate the utility our model on multiple synthetic tasks and semi-synthetic chemistry datasets, where the ground truth structure is known, as well as an image dataset and a real-world cancer dataset.

1 Introduction

Feature selection is a key problem permeating statistics, machine learning and broader science. Typically in high-dimensional datasets, the majority of features will not be responsible for the target response and thus an important goal is to identify which variables are truly predictive. For example, in healthcare there may be many features (such as age, sex, medical history, etc.) that could be considered, while only a small subset might in fact be relevant for predicting the likelihood of developing a specific disease. By eliminating irrelevant variables, feature selection algorithms can be used to drive discovery, improve model generalisation/robustness, and improve interpretability [18].

However, features often do not act alone but instead in *combination*. In genetics, for instance, it has been noted that understanding the origins of many diseases may require methods able to identify more complex genetic models than single variants [65]. More generally, often there are multiple groups of variables which act (somewhat) independently of each other. For example, in medicine or biology, a number of diseases can manifest from different mechanisms or pathways. Examples include cancer [63], amyotrophic lateral sclerosis (ALS) and frontotemporal dementia (FTD) [6], inflammatory bowel disease [31], cardiovascular disease [42], and diabetes [64].

While feature selection might be able to identify a set of features associated with a particular response, the underlying structure of how features interact is not captured. Further, the resulting

*Equal contribution

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

predictive models can be complex, hard to interpret, and not amenable to the generation of hypotheses that can be experimentally tested [46]. This limits the impact such models can have in furthering scientific understanding across many domains where variables are known to interact, such as genetics [65, 69, 60], medicine [89, 13], and economics [8].

Group feature selection is a generalisation of standard feature selection, where instead of selecting individual features, groups of features are either entirely chosen or entirely excluded. A primary application of group feature selection is when features are jointly measured, for example by different instruments. In such scenarios, groups are readily defined as features measured by the same instrument. A natural question is which instruments give the most meaningful measurements. Group feature selection has also been applied in situations where there is extensive domain knowledge regarding the group structure [72] or where groups are defined by the correlation structure between features (e.g. neighbouring pixels in images are highly correlated). The pervasive issue with current group feature selection methods is that a predetermined grouping *must* be provided, and the groups are selected from the given candidates. In reality, we may not know how to group the variables.

In this paper we seek to solve a related but ultimately different and more challenging problem, which we call *Composite Feature Selection*. We wish to find groups of variables *without* prior knowledge, where each group acts as a separate predictive subset of the features and the overall predictive power is greatest when all groups are used in unison. We call each group of features a composite feature.[1] By imposing this structure on the discovered features, we attempt to isolate pathways from features to the response variable. Discovering *groups* of features offers deeper insights into *why* specific features are important than standard feature selection.

Contributions. (1) We formalise *composite* feature selection as an extension of standard feature selection, defining composite features in terms of linear and non-linear interactions between variables (Sec. 3). (2) We propose a new deep learning architecture for composite feature selection using an ensemble-based approach (Sec. 4). (3) To assess our solution, we introduce a metric for assessing composite feature similarity based on Jaccard similarity (Sec. 5). (4) We demonstrate the utility of our model on a range of synthetic and semi-synthetic tasks where the ground truth group features are known (Sec. 5). We see that our model not only frequently recovers the relevant features, but also often discovers the underlying group structure. We further illustrate our approach on an image dataset and a real-world cancer dataset, corroborating discovered features and feature interactions in the scientific literature.

2 Related Work

Significant attention has been placed on feature selection with a range of solutions including traditional methods (e.g. [56, 47, 34]) and deep learning approaches [54, 91, 7, 49] (see Appendix B for further discussion of standard feature selection). Several approaches have been extended to select predefined groups of variables instead of individual features. For example, LASSO [76, 83] is a linear method that uses an L1 penalty to impose sparsity among coefficients. Group LASSO [93] generalises this to allow predefined groups to be selected or excluded jointly, rather than single features, by replacing the L1 penalty with L2 penalties on each group. Other feature selection methods, such as SLOPE [12], have been similarly extended to group feature selection to give Group-SLOPE [14]. Further examples of group feature selection using adapted loss functions are SCAD-L2 [95] and hierarchical LASSO [100]. Similarly, Bayesian approaches to feature selection [28] have also been generalised to the group setting [35]. Finally, the Knockoff procedure [9, 17, 39, 57, 73, 80] is a generative procedure that creates fake covariates (knockoffs), obeying certain symmetries under permutations of real and knockoff features. By subsequently carrying out feature selection on the combined real and knockoff data, it is possible to obtain guarantees on the False Discovery Rate of the selected features. Generalisations of the Knockoff procedure to the group setting also exist [23, 101], where symmetries under permutations of entire groups must exist.

The key commonality is that none of these methods *discover* groups, but instead can only *select* groups from a set of predefined candidates. Therefore, while they may be applicable when we can split inputs into groups, they are not able to find groups of predictors on their own. Our work differs from these methods by considering the challenge of finding such groups in the absence of prior knowledge. Additionally, unlike prior work, we do not make assumptions about correlations

[1] We will often refer to composite features as groups for brevity; in this paper, they refer to the same thing.

between features or place restrictions on groups, such as requiring the candidate groups to partition the features.

3 Problem Description

Let $\mathbf{X} \in \mathcal{X}^p$ be a p-dimensional signal (such as gene expressions or patient covariates) and $Y \in \mathcal{Y}$ be a response (such as disease traits). Informally, we wish to group features into the maximum number of subsets, $\mathcal{G}_i \subset [p]$, where the predictive power of any single group significantly decreases when any feature is removed, allowing us to separate the groups into different pathways from the signal to the response. Note that we do not enforce assumptions on the groups such as non-overlapping groups or every feature being in at least one group. In this section, we begin with a description of traditional feature selection before formalizing composite feature selection.

3.1 Feature Selection

The goal of traditional feature selection is to select a subset, $\mathcal{S} \subset [p]$, of features that are relevant for predicting the response variable. In particular, in the case of embedded feature selection [32], this is conducted jointly with the model selection process.

Let $*$ denote any point not in $\mathcal{X}$ and define $\mathcal{X}_\mathcal{S} = (\mathcal{X} \cup \{*\})^p$. Then, given $\mathbf{X} \in \mathcal{X}^p$, the selected subset of features can be denoted as $\mathbf{X}_\mathcal{S} \in \mathcal{X}_\mathcal{S}$ where $x_{\mathcal{S},k} = x_k$ if $k \in \mathcal{S}$ and $x_{\mathcal{S},k} = *$ if $k \notin \mathcal{S}$. Let $f : \mathcal{X}_\mathcal{S} \to \mathcal{Y}$ be a function in some space $\mathcal{F}$ (such as the space of neural networks) taking subset $\mathbf{X}_\mathcal{S}$ as input to yield Y. Then, selecting relevant features for predicting a response can be achieved by solving the following optimization problem:

$$\underset{f \in \mathcal{F},\ \mathcal{S} \subset [p]}{\text{minimize}} \quad \mathbb{E}_{\mathbf{x}, y \sim p_{XY}} \Big[\ell_Y \big(y, f(\mathbf{x}_\mathcal{S}) \big) \Big] \ \text{ subject to } \ |\mathcal{S}| \leq \delta, \tag{1}$$

where δ constrains the number of selected features and $\ell_Y(y, y')$ is a task-specific loss function.

This can be solved by introducing a selection vector $\mathbf{M} = (M_1, \cdots, M_p) \in \{0, 1\}^p$, consisting of binary random variables governed by distribution p_M, with realization $\mathbf{m}$ indicating selection of the corresponding features. Then, the selected features given vector $\mathbf{m}$ can be written as

$$\tilde{\mathbf{x}} \triangleq \mathbf{m} \odot \mathbf{x} + (1 - \mathbf{m}) \odot \hat{\mathbf{x}}, \tag{2}$$

where $\odot$ indicates element-wise multiplication and $\hat{\mathbf{x}}$ are the values assigned to features that are not selected (typically $\hat{\mathbf{x}} \equiv 0$ or $\bar{\mathbf{x}}$). Eq. (1) can be (approximately) solved by jointly learning the model f and the selection vector distribution p_M based on the following optimization problem:

$$\underset{f,\ p_M}{\text{minimize}} \ \ \mathbb{E}_{\mathbf{x}, y \sim p_{XY}} \mathbb{E}_{\mathbf{m} \sim p_M} \Big[\ell_Y \big(y, f(\tilde{\mathbf{x}}) \big) + \beta \|\mathbf{m}\|_0 \Big], \tag{3}$$

where β is a balancing coefficient that controls the number of features to be selected.

3.2 Composite Feature Selection

The goal of composite feature selection is not only to find the predictive features, but also to group them based on *how* they are predictive. For example, assume features x_1 and x_2 are only predictive when both are known by the model, but have the same influence on the outcome independent of x_3. Then we wish to group x_1, x_2 separately from x_3. In this section, we define the embedded composite feature selection problem; that is, we want to find a valid model f and groups $\{\mathcal{G}_1, \ldots, \mathcal{G}_N\}$ in parallel. A model is only valid when the group representations are combined in a way where we can view each group as contributing an independent piece of information for the final prediction. A valid model acts on a *set* of groups [94], thus when combining groups, we require order not to matter. Therefore, we must combine the representations using a permutation invariant aggregator.

Let $A : (\prod_i \mathbb{R}^n) \to \mathbb{R}^N$ be a general permutation invariant aggregation function. It is well established that for a specific choice of $\phi : \mathbb{R}^n \to \mathbb{R}^m$ and $\rho : \mathbb{R}^m \to \mathbb{R}^N$, A can be decomposed as $\rho[\sum_i \phi(\cdot)]$ (see [94] for examples). This gives $f(\mathbf{x}) = g\big(\rho\big[\sum_i \phi(f_i(\mathbf{x}_{\mathcal{G}_i}))\big]\big)$, where f_i encodes group i, ρ and ϕ give the permutation invariant aggregation, and g is any final non-linear function, for instance softmax. The function composition of ϕ and f_i can be relabelled as $\tilde{f}_i = \phi \circ f_i$, and the composition of g and ρ can be relabelled as $\tilde{\rho} = g \circ \rho$. This leads to $f(\mathbf{x}) = \tilde{\rho}\big[\sum_i \tilde{f}_i(\mathbf{x}_{\mathcal{G}_i})\big]$, giving the following definition for a valid model structure in composite feature selection.

Definition 3.1. The most general valid model for acting on N composite features is given by:

$$f(\mathbf{x}) = \rho\Big[\sum_{i=1}^{N} f_i(\mathbf{x}_{\mathcal{G}_i})\Big]. \tag{4}$$

That is, the groups must interact exactly once, all groups must be included, and the interaction is a summation; all other interactions can (and often should) be non-linear.

Depending on the task, a specific permutation invariant aggregation may be chosen (e.g. `Max()`). However, any permutation invariant aggregator can be (approximately) expressed in the form of Def. 3.1; thus, when learning from data, the general structure of Def. 3.1 means that this is not necessary.

The embedded composite feature selection problem can now be phrased in an analogous way to traditional feature selection. Let $*$ denote some point not in $\mathcal{X}$ and define $\mathcal{X}_{\mathcal{G}_i} = (\mathcal{X} \cup \{*\})^p$. Then, given $\mathbf{X} \in \mathcal{X}^p$, the selected group of features is denoted as $\mathbf{X}_{\mathcal{G}_i} \in \mathcal{X}_{\mathcal{G}_i}$ where $x_{\mathcal{G}_i,k} = x_k$ if $k \in \mathcal{G}_i$ and $x_k = *$ if $k \notin \mathcal{G}_i$. Let $f_i : \mathcal{X}_{\mathcal{G}_i} \to \mathcal{Z}$ be a function in $\mathcal{F}$ that takes as input the subset $\mathbf{X}_{\mathcal{G}_i}$ and outputs a latent representation $\mathbf{z}_i$. Then, finding the groups of features can be achieved by solving the optimization problem:

$$\underset{\rho, f_i \in \mathcal{F},\ \mathcal{G}_i \subset [p]}{\text{minimize}} \quad \mathbb{E}_{\mathbf{x},y\sim p_{XY}}\left[\ell_Y\left(y, \rho\big[\sum_{i=1}^{N} f_i(\mathbf{x}_{\mathcal{G}_i})\big]\right)\right] \qquad \text{subject to} \quad \begin{array}{l} |\mathcal{G}_i| \leq \delta_i \ \ \forall i, \\ N \geq \Delta, \end{array} \tag{5}$$

where $\boldsymbol{\delta}$ constrains the number of selected features in each group and Δ gives the minimum number of groups. This objective leads to multiple smaller groups, rather than one group containing all features, which is consistent with our motivation of the problem.

Continuing to expand from traditional feature selection, we can also extend the solution to the composite setting. For N groups we can introduce a selection *matrix* $\mathbf{M} \in \{0,1\}^{N\times p}$, governed by distribution p_M. For a realization $\mathbf{M}$, the selected features from group i are given by

$$\tilde{\mathbf{x}}_i \triangleq \mathbf{m}_i \odot \mathbf{x} + (1 - \mathbf{m}_i) \odot \hat{\mathbf{x}}, \tag{6}$$

where $\mathbf{m}_i$ is the ith row of $\mathbf{M}$. We can approximately solve Eq. (5) by solving the optimization problem:

$$\underset{f,\ p_M}{\text{minimize}} \ \ \mathbb{E}_{\mathbf{x},y\sim p_{XY}}\mathbb{E}_{\mathbf{M}\sim p_M}\Big[\ell_Y\big(y, f(\mathbf{x})\big) + R_e(\mathbf{M})\Big], \tag{7}$$

where $f(\mathbf{x})$ obeys Def. (4) and R_e is a regularisation term which controls how features are selected in each group. R_e should capture both group size (i.e. encourage as few features as possible to be selected) but also the relationships between groups (i.e. groups should be distinct and not redundant).

3.3 Challenges

There are various challenges in solving the composite feature selection problem. While the ultimate task is to find predictive groups of features, there first remains the necessity simply to identify predictive features, which is already an NP-hard problem [2]. Composite feature selection not only inherits this property but introduces additional complexity since we can think of each group as solving a separate feature selection problem. Consider the number of potential solutions: in traditional feature selection (assuming not all features are selected), there are $2^n - 2$ ways of selecting a subset from n features; even restricting to at most $m << n$ quickly becomes unfeasible for even modest values of m. In composite feature selection, *every group* has the same number of solutions as traditional feature selection, drastically increasing the total number of possible solutions. A challenge specific to composite feature selection arises when the ground truth group structure contains groups with overlapping features (e.g. feature x_1 interacts independently with both x_2 and x_3). In this scenario, it is difficult to separate these two effects while penalizing the inclusion of additional features.

4 Method: CompFS

In this section, we propose a novel architecture for finding predictive groups of features, which we refer to as **Comp**osite **F**eature **S**election (CompFS). In order to discover groups of features, our model is composed of a set of group selection models and an aggregate predictor. Our approach resembles

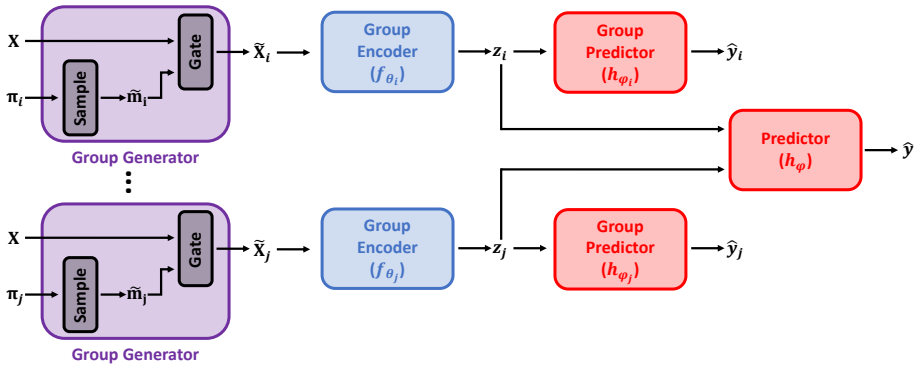


Figure 1: An illustration of CompFS. We use an ensemble of group selection models to discover composite features and an aggregate predictor to combine these features when issuing predictions.

an ensemble of "weak" feature selection models, where each learner attempts to solve the task using a sparse set of features (Figure 1). These models are then trained in such a way as to discover distinct predictive groups. We first consider the group selection models in more detail before describing how the group selection models are combined and the training procedure.

4.1 Group Selection Models

CompFS is composed of a set of group selection models, each of which primarily aims to solve the traditional feature selection problem specified by Eq. (1). We achieve this by solving Eq. (3) using a neural network-based approach with stochastic gating of the input features. Each group selection model consists of the following three components (Figure 1):

- *Group Selection Probability*, $\boldsymbol{\pi}_i = (\pi_{1,i}, \cdots, \pi_{p,i}) \in [0,1]^p$, which is a trainable vector that governs the Bernoulli distribution used to generate the gate vector $\mathbf{m}_i$. Each element of the selection probability $\pi_{k,i}$ indicates the importance of the corresponding feature to the target.
- *Group Encoder*, $f_{\theta_i} : \mathcal{X}^p \to \mathcal{Z}$, that takes as input the selected subset of features $\tilde{\mathbf{x}}_i$ and outputs latent representations $\mathbf{z}_i \in \mathcal{Z}$.
- *Group Predictor*, $h_{\phi_i} : \mathcal{Z} \to \mathcal{Y}$, that takes as input the latent representations of the selected subset of features, $\mathbf{z}_i = f_{\theta_i}(\tilde{\mathbf{x}}_i)$, and outputs predictions on the target outcome.

Solving Eq. (3) directly is not possible since the sampling step has no differentiable inverse. Instead, we use the relaxed Bernoulli distribution [59, 38] and apply the reparameterization trick as follows.

Formally, given selection probability $\boldsymbol{\pi} = (\pi_1, \cdots, \pi_p)$ and independent Uniform$(0,1)$ random variables $(U_1, \cdots, U_p)$, we can generate a relaxed gate vector $\tilde{\mathbf{m}} = (\tilde{m}_1, \cdots, \tilde{m}_p) \in (0,1)^p$ based on the following reparameterization trick [59]:

$$\tilde{m}_k = \sigma\Big(\frac{1}{\tau}\big(\log \pi_k - \log(1-\pi_k) + \log U_k - \log(1-U_k)\big)\Big), \tag{8}$$

where $\sigma(x) = (1+\exp(-x))^{-1}$ is the sigmoid function. This relaxation is parameterized by $\boldsymbol{\pi}$ and temperature $\tau \in (0, \infty)$. Further, as $\tau \to 0$, the gate vectors $\tilde{m}_k$ converge to Bernoulli(π_k) random variables. Crucially, this is differentiable with respect to $\boldsymbol{\pi}$.

Given group selection probability $\boldsymbol{\pi}_i$, we first sample relaxed Bernoulli random variable $\tilde{\mathbf{m}}_i$ according to Eq. (8) and then use $\tilde{\mathbf{m}}_i$ in a gating procedure to select the group of features. The output of the gate is:

$$\tilde{\mathbf{x}}_i = \text{gate}_i(\mathbf{x}) = \tilde{\mathbf{m}}_i \odot \mathbf{x} + (1 - \tilde{\mathbf{m}}_i) \odot \bar{\mathbf{x}}, \tag{9}$$

where we replace the variables that were not selected by their mean value $\bar{\mathbf{x}}$. The mean is used because in particular tasks a feature having a value of 0 may be particularly meaningful. However, any (arbitrary) value could be used for non-selected features. The gate output $\tilde{\mathbf{x}}_i$ is then fed into the group encoder f_{θ_i} to yield representation $\mathbf{z}_i = f_{\theta_i}(\tilde{\mathbf{x}}_i)$. This representation is finally passed to the group predictor h_{ϕ_i} to produce the prediction for an individual learner, $\hat{y}_i = h_{\phi_i}(\mathbf{z}_i)$.

4.2 Group Aggregation

The final component necessary for CompFS is a way to aggregate the individual group selection models. This is achieved via an overall *predictor*, $h_\phi : \mathcal{Z} \to \mathcal{Y}$, that takes as input the set of latent representations $\{\mathbf{z}_1, \dots, \mathbf{z}_N\}$ produced by the individual learners and outputs predictions on the target outcome. For simplicity, we apply a linear prediction head to the latent representations and use element-wise summation to aggregate. Thus, the prediction of the ensemble is given by:

$$\hat{y} = h_\phi(\{\mathbf{z}_1, \dots, \mathbf{z}_N\}) = \rho\left[\sum_{i=1}^{N} \mathbf{W}_i \mathbf{z}_i + \mathbf{b}_i\right], \tag{10}$$

where N is the number of members of the ensemble (i.e. the number of groups) and ρ is a suitable transformation (e.g. softmax). Note that by using element-wise summation, our model satisfies Def. (4) for acting on composite features.

4.3 Loss Functions

Group Selection Models. The individual learners can be trained to perform (traditional) feature selection (Eq. (1)) by minimizing the following loss function:

$$\mathcal{L}_{\mathcal{G}_i} = \mathbb{E}_{\mathbf{x}, y \sim p_{XY}}\left[\ell_Y\big(y, h_{\phi_i}(f_{\theta_i}(\text{gate}_i(\mathbf{x})))\big) + \beta \langle \boldsymbol{\pi}_i \rangle^2\right], \tag{11}$$

where ℓ_Y is a suitable loss function for the prediction task (e.g. cross-entropy for classification tasks and MSE for regression tasks) and $\beta \geq 0$ balances the two terms. Note the selections probabilities $\boldsymbol{\pi}_i$ are not regularized with the typical L1 penalty. Instead, we apply an L2 penalty to the mean selection probability $\langle \boldsymbol{\pi}_i \rangle$ for each individual learner. This is justified as follows. Recall the optimization problem given by Eq. (5). We desire a solution with the maximal number of predictive groups N while minimizing the number of selected features per group $\sum_{i=1}^{N} |\mathcal{G}_i|$. The standard L1 penalty term does not achieve this goal since adding an additional feature to either group $\mathcal{G}_i$ or $\mathcal{G}_j$ incurs the same penalty. In contrast, the L2 penalty imposed on $\langle \boldsymbol{\pi}_i \rangle$ penalizes adding extra features to already large groups, favoring the construction of smaller groups over larger ones.

Aggregate Predictor. The aggregate predictor can be trained jointly with the group selection models by minimizing a standard prediction loss (where ℓ_Y is the same as in Eq. (11)):

$$\mathcal{L}_E = \mathbb{E}_{\mathbf{x}, y \sim p_{XY}}\left[\ell_Y\big(y, h_\phi(\{\mathbf{z}_1, \dots, \mathbf{z}_n\})\big)\right]. \tag{12}$$

Additional Regularization. If we simply apply the losses given by Eqs. (11), (12), there will be limited (or even no) differentiation among the individual learners and the optimal solution would be for each learner to simply solve the traditional feature selection problem (Eq. (1)). This results in all learners selecting the same features, which does not achieve our aim of discovering groups of predictive features. In order to encourage differentiation between the models, we introduce an additional loss that penalizes the selection of the same features in multiple groups:

$$\mathcal{L}_R = \mathbb{E}_{\mathbf{x}, y \sim p_{XY}}\left[\sum_{i=1}^{N} \sum_{j>i} \boldsymbol{\pi}_i \cdot \boldsymbol{\pi}_j\right]. \tag{13}$$

Overall Loss. Combining the above, our overall loss function therefore can be written as follows:

$$\mathcal{L} = \sum_{i=1}^{N} \mathcal{L}_{\mathcal{G}_i} + \beta_E \mathcal{L}_E + \beta_R \mathcal{L}_R, \tag{14}$$

where $\beta_E, \beta_R \geq 0$ are hyperparameters to balance the losses.

Training CompFS with the loss given by Eq. (14) is designed to achieve the following: (1) The overall ensemble network should be a good predictor ($\mathcal{L}_E$). (2) Each individual learner should solve the traditional feature selection problem ($\mathcal{L}_{\mathcal{G}_i}$), which requires the group predictor to be accurate while selecting minimal features. However, the individual learners should not be maximally predictive by

definition (hence why we compare individual group feature selection models to weak learners). (3) Finally, we want the groups to be distinct and thus discourage highly similar groups ($\mathcal{L}_R$). However, note that we do not exclude the possibility of some overlap of features between groups. The model is end-to-end differentiable, so we train with gradient descent.

Evaluation. During evaluation, only the gating procedure changes. The way features can be selected is chosen by the user. A standard solution which we adopt in this paper is using a threshold λ and computing gate vectors $\mathbf{m}_i$ as follows: $m_{i,k} = 1$, if $\pi_{i,k} > \lambda$ and 0 otherwise.

5 Experiments

We evaluate CompFS using several synthetic and semi-synthetic datasets where ground truth feature importances and group structure are known. In addition, we illustrate our method on an image dataset (MNIST) and a real-world cancer dataset (METABRIC). Specific architectural details are given in App. C. Additional information regarding experiments, benchmarks, and datasets can be found in App. D. Additional ablations and sensitivity analysis are in App. A. The code for our method and experiments is available on Github. [2] [3]

Benchmarks. The primary goal of our experiments is to demonstrate the utility of discovering composite features over traditional feature selection. Our main benchmark is an oracle feature selection method ("Oracle") that perfectly selects the ground truth features but provides no structure, giving all features as one group. By definition, this is the strongest standard feature selection baseline for the scenarios where the ground truth features are known. We also include comparisons to a linear feature selection method (LASSO) [83] and two non-linear, state of the art approaches, Stochastic Gates (STG) [91] and Supervised Concrete Autoencoder (Sup-CAE) [7]. Finally, we compare with Group LASSO [93], where we enumerate all groups with 1 or 2 features as predefined groups. Note this represents a significant simplification of the task for Group Lasso. We include additional baselines in App. G.

Metrics. When the ground truth feature groups $\mathcal{G}_1, \ldots, \mathcal{G}_N$ are known, we use True Positive Rate (TPR) and False Discovery Rate (FDR) to assess the discovered features against the ground truth. To assess composite features, i.e. grouping, we define the Group Similarity (G_{sim}) as the normalized Jaccard similarity between ground truth feature groups and the most similar proposed group:

$$\text{G}_{\text{sim}} = \frac{1}{\max(N, K)} \sum_{i=1}^{N} \max_{j \in [K]} \mathcal{J}(\mathcal{G}_i, \hat{\mathcal{G}}_j), \tag{15}$$

where $\mathcal{J}$ is the Jaccard index [37] and $\hat{\mathcal{G}}_1, \ldots, \hat{\mathcal{G}}_K$ are the discovered groups. $\text{G}_{\text{sim}} \in [0, 1]$, where $\text{G}_{\text{sim}} = 1$ corresponds to perfect recovery of the ground truth groups, while $\text{G}_{\text{sim}} = 0$ when none of the correct features are discovered (see App. E for additional details together with examples). We assess the models by seeing if the ground truth features have been correctly discovered, using TPR and FDR. We then see if the underlying grouping has been uncovered (and correct features) using G_{sim}. Finally, we assess the predictive power of the discovered features using accuracy or area under the receiver operating curve (AUROC).

5.1 Synthetic Experiments.

Dataset Description. We begin by evaluating our method on a range of synthetic datasets where the ground truth feature importance is known (Table 1). We generate synthetic datasets by sampling from the Gaussian distribution with initially no correlations among features ($X \sim \mathcal{N}(0, I)$). We construct binary classification tasks, where the class y is determined by the following decision rules:

- **(Syn1)** $y = 1$ if $x_1 > 0.55$ or $x_2 > 0.55$, 0 otherwise. The ground truth groups are $\{\{1\}, \{2\}\}$. This task assesses whether the model can separate two features rather than group them together.
- **(Syn2)** $y = 1$ if $x_1 x_2 > 0.30$ or $x_3 x_4 > 0.30$, 0 otherwise. The ground truth groups are $\{\{1, 2\}, \{3, 4\}\}$. This task requires identifying groups consisting of more than one variable.
- **(Syn3)** $y = 1$ if $x_1 x_2 > 0.30$ or $x_1 x_3 > 0.30$, 0 otherwise. The ground truth groups are $\{\{1, 2\}, \{1, 3\}\}$. This task investigates whether a model can split the features into two *overlapping* groups of two, rather than one group with all three features.

[2] `https://github.com/a-norcliffe/Composite-Feature-Selection`

[3] `https://github.com/vanderschaarlab/Composite-Feature-Selection`

Table 1: Performance on Synthetic Datasets, values are recorded with their standard deviations.

DATASET	MODEL	TPR	FDR	G_{SIM}	NO. GROUPS	ACCURACY (%)
SYN1	COMPFS(5)	100.0 ± 0.0	0.0 ± 0.0	0.91 ± 0.14	2.2 ± 0.4	98.9 ± 0.5
	ORACLE	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	LASSO	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	81.8 ± 2.0
	GROUP LASSO	100.0 ± 0.0	0.0 ± 0.0	0.67 ± 0.00	3.0 ± 0.0	83.8 ± 1.4
	STG	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	97.8 ± 1.4
	SUP-CAE	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	97.8 ± 1.4
SYN2	COMPFS(5)	95.0 ± 15.0	0.0 ± 0.0	0.90 ± 0.20	1.8 ± 0.4	95.5 ± 5.4
	ORACLE	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	LASSO	0.0 ± 0.0	0.0 ± 0.0	0.00 ± 0.00	0.0 ± 0.0	52.6 ± 2.9
	GROUP LASSO	0.0 ± 0.0	0.0 ± 0.0	0.00 ± 0.00	0.0 ± 0.0	52.2 ± 0.9
	STG	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	93.9 ± 2.2
	SUP-CAE	37.5 ± 31.7	42.5 ± 44.2	0.24 ± 0.20	1.0 ± 0.0	61.9 ± 12.8
SYN3	COMPFS(5)	100.0 ± 0.0	0.0 ± 0.0	0.68 ± 0.05	1.3 ± 0.5	97.4 ± 1.1
	ORACLE	100.0 ± 0.0	0.0 ± 0.0	0.67 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	LASSO	0.0 ± 0.0	0.0 ± 0.0	0.00 ± 0.00	0.0 ± 0.0	56.5 ± 4.0
	GROUP LASSO	0.0 ± 0.0	0.0 ± 0.0	0.00 ± 0.00	0.0 ± 0.0	54.6 ± 1.3
	STG	100.0 ± 0.0	0.0 ± 0.0	0.67 ± 0.00	1.0 ± 0.0	95.3 ± 1.7
	SUP-CAE	23.3 ± 31.6	66.7 ± 47.1	0.23 ± 0.31	1.0 ± 0.0	62.6 ± 12.6
SYN4	COMPFS(5)	90.0 ± 12.2	51.9 ± 13.8	0.47 ± 0.20	2.5 ± 0.7	95.8 ± 1.8
	ORACLE	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	LASSO	0.0 ± 0.0	0.0 ± 0.0	0.00 ± 0.00	0.0 ± 0.0	51.8 ± 3.2
	GROUP LASSO	0.0 ± 0.0	10.0 ± 31.6	0.00 ± 0.00	0.1 ± 0.3	53.0 ± 1.1
	STG	100.0 ± 0.0	66.7 ± 0.0	0.17 ± 0.00	1.0 ± 0.0	94.2 ± 2.1
	SUP-CAE	72.5 ± 14.2	16.7 ± 14.7	0.39 ± 0.08	1.0 ± 0.0	72.2 ± 13.2

- **(Syn4)** $y = 1$ if $x_1x_4 > 0.30$ or $x_7x_{10} > 0.30$, 0 otherwise. The ground truth groups are $\{\{1, 4\}, \{7, 10\}\}$. This task is equivalent to **Syn2**, however, here the features exhibit strong correlation in collections of 3, i.e. features 1, 2, and 3 are highly correlated, features 4, 5, and 6 are highly correlated, and so on. This task demonstrates the difficulty of carrying out group feature selection (and indeed standard feature selection) when the features are highly correlated.

The decision rules are created such that there is minimal class imbalance. We use signals with 500 dimensions to demonstrate the utility in the high dimensional regime. We use 20,000 samples to train and 200 to test. Each experiment is repeated 10 times.

Analysis. On both Syn1 and Syn2, CompFS achieves high TPR with no false discoveries (0% FDR) and significantly higher G_{sim} than the Oracle. Despite allowing CompFS to discover up to 5 groups, CompFS typically finds the correct number of groups (2), demonstrating that it is not necessary for the number of potential composite features to match the ground truth, which is vital in real-world use cases where this is unknown. Syn3 is significantly more challenging due to the overlapping structure and we observe essentially the same performance as Oracle. Despite finding all the correct features and no false discoveries, CompFS typically finds the union $\{1, 2, 3\}$ rather than the underlying group structure $\{\{1, 2\}, \{1, 3\}\}$. Finally, for Syn4, while CompFS has a relatively high FDR, it frequently finds the ground truth relevant features and groups with similar G_{sim} to Oracle. This is a challenging task with significant correlation between features. Despite this, CompFS is able to uncover the underlying group structure, providing additional insight over traditional feature selection. STG typically performs reasonably in terms of traditional feature selection, but scores poorly in terms of G_{sim} due to not providing any group information.

5.2 Semi-Synthetic Experiments.

Dataset Description. Next, we assess our ability to identify composite features using semi-synthetic molecular datasets. These tasks are analogs of real-world problems, such as identifying biologically active chemical groups; however, the labels are determined by a synthetic "binding logic" so that the ground truth feature relevance is known. We use several of the datasets constructed by [62], some of which were also used by [75].[4] The synthetic "binding logics" are expressed as a combination of molecular fragments that must either be present or absent for binding to occur and are used to label molecules from the ZINC database [36]. Each logic includes up to four functional groups (Table 6).

[4]Data from `https://github.com/google-research/graph-attribution/raw/main/data/all_16_logics_train_and_test.zip`.

Molecules are featurized using a set of 84 functional groups, where feature $x_i = 1$ if the molecule contains functional group i and 0 otherwise. The specific binding logics are given in App. F.

Table 2: Performance on Chemistry Datasets, values are recorded with their standard deviations.

DATASET	MODEL	TPR	FDR	G_{SIM}	NO. GROUPS	ACCURACY (%)
CHEM1	COMPFS(5)	100.0 ± 0.0	0.0 ± 0.0	0.82 ± 0.20	1.9 ± 0.5	100.0 ± 0.0
	ORACLE	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	LASSO	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	75.8 ± 0.0
	GROUP LASSO	100.0 ± 0.0	0.0 ± 0.0	0.67 ± 0.00	3.0 ± 0.0	100.0 ± 0.0
	STG	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	SUP-CAE	62.5 ± 13.2	23.3 ± 17.5	0.37 ± 0.07	1.0 ± 0.0	77.8 ± 11.0
CHEM2	COMPFS(5)	100.0 ± 0.0	0.0 ± 0.0	0.72 ± 0.24	2.2 ± 0.6	100.0 ± 0.0
	ORACLE	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	LASSO	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	81.6 ± 0.0
	GROUP LASSO	100.0 ± 0.0	0.0 ± 0.0	0.40 ± 0.00	5.0 ± 0.0	81.6 ± 0.0
	STG	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	SUP-CAE	66.7 ± 0.0	0.0 ± 0.0	0.42 ± 0.00	1.0 ± 0.0	80.9 ± 9.5
CHEM3	COMPFS(5)	100.0 ± 0.0	7.3 ± 11.7	0.62 ± 0.17	2.4 ± 0.5	100.0 ± 0.0
	ORACLE	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	LASSO	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	87.4 ± 5.2
	GROUP LASSO	100.0 ± 0.0	20.0 ± 0.0	0.20 ± 0.00	10.0 ± 0.0	91.5 ± 0.0
	STG	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	SUP-CAE	62.5 ± 13.2	23.3 ± 17.5	0.37 ± 0.07	1.0 ± 0.0	77.8 ± 11.0

Analysis. All methods are able to identify the ground truth relevant features; however, only CompFS provides deeper insights. Unlike for Syn1-4, LASSO correctly selects the ground truth features since the dataset consists of binary variables and thus it is possible to find performant linear models. However, while discovering the correct features, Group LASSO selects all possible combinations of these features, adding no benefit over standard feature selection.

For Chem1-2, CompFS perfectly recovers the group structure in the majority of experiments, leading to high G_{sim} far exceeding traditional feature selection. On Chem3, we occasionally discover additional features that are not part of the binding logic. However, a number of molecular fragments are strongly correlated with the binding logic, even though they are not themselves included. In fact, some features contain information about *multiple* functional groups. For example, esters contain a carbonyl and an ether; both are in the binding logic for Chem3, while ester is not, despite being highly informative, and thus occasionally CompFS incorrectly selects this feature. In spite of this, CompFS achieves significantly higher G_{sim} than even Oracle. This demonstrates the benefit of the grouping discovered by CompFS, even with a modest number of false discoveries. As before, CompFS typically finds the correct number of groups (2), despite being able to discover up to 5 groups, further demonstrating that the number of composite features need not be known *a priori*, which is the case in real-world applications.

5.3 MNIST

Dataset Description. We investigate CompFS on the MNIST dataset [48]. While this well-known dataset consists of 28x28 images and typically fixed pixel locations do not have specific meaning, it has been extensively used in the feature selection literature due to the handwritten digits being centered and thus each of the 784 pixels can be (somewhat) meaningfully treated as a separate feature. While the ground truth group structure is unknown, a benefit of MNIST is that we can readily visualise selected features.

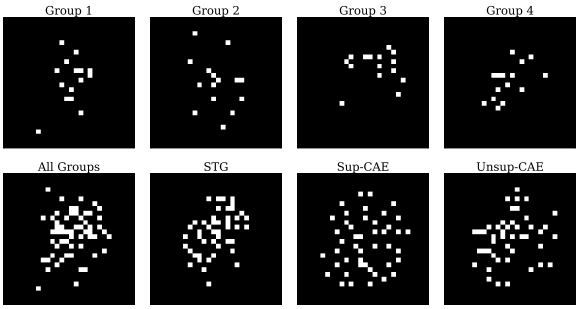


Figure 2: Pixels selected by CompFS and baselines.

Analysis. The features discovered by CompFS (using 4 groups), STG, Sup-CAE and Unsup-CAE are shown in Figure 2. As expected, all selected pixels are central and relatively spread out. However, the four groups discovered by CompFS appear to have slightly different focus, in particular Group 3.

To investigate the impact of these differences, we evaluate the predictive power of each of the groups. We find that the individual groups have relatively low accuracies (72%-81%), in part due to only using 15 pixels (<2% of total). However, the union of these features achieve significantly greater accuracy of 95%, equalling the performance of STG and (Sup-,Unsup-)CAE. This illustrates that the grouping does not seem to drastically affect performance, despite enforcing constraints on how the features are used by the model.

Finally, we consider the per-class accuracy of each of these groups. Interestingly, the variance in performance between classes is significantly higher for the groups than any of the overall methods (Fig. 4). For example, Group 2 struggles to identify digits 4 and 5, while Group 3 performs poorly on digits 2, 3, and 8. This highlights the distinct information contained in each group. Further details and analysis is provided in App. G.3.

5.4 Real-World Data: METABRIC

Dataset Description. Finally, we assess CompFS on a real-world dataset, METABRIC [21, 68], where the ground truth group structure is *unknown*. METABRIC contains gene expression, mutation, and clinical data for 1,980 primary breast cancer samples. We evaluated the ability to predict the progesterone receptor (PR) status of the tissue based on the gene expression data, which consists of measurements for 489 genes.

Table 3: METABRIC performance. We compare CompFS and STG using 25 features to an MLP using all 489 features.

Model	AUROC
MLP (All features)	0.869
CompFS(5)	0.830
STG	0.843

Analysis. CompFS suffers limited performance degradation compared to using all features, despite only using 5% of the features (Table 3). Despite imposing a more rigid structural form on how features can interact in the predictive model, STG only had marginally greater predictive power than CompFS. However, CompFS provides greater insight into how the features interact than STG.

We found supporting evidence in the scientific literature for all but 1 of the genes discovered by CompFS (Table 10). In addition, within each group, we found further evidence of the interactions between genes, demonstrating the ability for CompFS to learn informative groups of features. For example, in Group 1, CXCR1 and PEN-2 (the protein encoded by PSENEN) are known to interact [5]. In Group 2, BMP6 encodes a member of the TGF-β superfamily of proteins, and TGF-β triggers activation of SMAD3 [19]. In the same group, MAPK1 activity is dependent on the activity of PRKCQ in breast cancer cells [15], while MAPK1 is also known to interact with MAPT [51], SMAD3 [26], and BMP6 [96]. Additional supporting evidence can be found in Appendix H.

6 Conclusion

In this paper, we introduced CompFS, an ensemble-based approach that tackles the newly proposed challenge of composite feature selection. Using synthetic and semi-synthetic data, we assess our ability to go beyond traditional feature selection and recover deeper underlying connections between variables. CompFS is not without limitations: as with other methods, points of difficulty arise when features are highly correlated, or if predictive composites contain overlapping features. Future work may overcome this by using correlated gates. Further, as with many traditional feature selection methods, there are no guarantees on false discovery rate. This could be tackled by first proposing candidate composite features, and then using the Group Knockoff procedure. Additionally, to discover groups, CompFS requires the introduction of additional hyperparameters which could be challenging to tune in practice. More broadly, as with standard feature selection, groups found under composite feature selection must be verified by domain experts (both features but additionally interactions). However, we believe the additional structure provided by composite feature selection could be of significant benefit to a wide variety of practitioners.

Acknowledgements

We thank the anonymous reviewers for their comments and suggestions. Fergus Imrie and Mihaela van der Schaar are supported by the National Science Foundation (NSF, grant number 1722516). Mihaela van der Schaar is additionally supported by the Office of Naval Research (ONR). Alexander Norcliffe is supported by a GlaxoSmithKline grant.

References

[1] Wail Al Sarakbi, Sara Reefy, Wen G. Jiang, Terry Roberts, Robert F. Newbold, and Kefah Mokbel. Evidence of a tumour suppressor function for DLEC1 in human breast cancer. *Anticancer Research*, 30(4):1079–1082, 2010.

[2] Edoardo Amaldi and Viggo Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209(1):237–260, 1998.

[3] M. Ampuja, E.L. Alarmo, P. Owens, R. Havunen, A.E. Gorska, H.L. Moses, and A. Kallioniemi. The impact of bone morphogenetic protein 4 (BMP4) on breast cancer metastasis in a mouse xenograft model. *Cancer Letters*, 375(2):238–244, 2016.

[4] Sharon Arcuri, Georgia Pennarossa, Fulvio Gandolfi, and Tiziana A. L. Brevini. Generation of trophoblast-like cells from hypomethylated porcine adult dermal fibroblasts. *Frontiers in Veterinary Science*, 8, 2021.

[5] Martina Bakele, Amelie S. Lotz-Havla, Anja Jakowetz, Melanie Carevic, Veronica Marcos, Ania C. Muntau, and Dominik Gersting, Soeren W.and Hartl. An interactive network of elastase, secretases, and PAR-2 protein regulates CXCR1 receptor surface expression on neutrophils. *Journal of Biological Chemistry*, 289(30):20516–20525, 2014.

[6] Rubika Balendra and Adrian M. Isaacs. C9orf72-mediated ALS and FTD: multiple pathways to disease. *Nature Reviews Neurology*, 14(9):544–558, Sep 2018.

[7] Muhammed Fatih Balın, Abubakar Abid, and James Zou. Concrete autoencoders: Differentiable feature selection and reconstruction. In *International Conference on Machine Learning (ICML)*, 2019.

[8] Hatice Ozer Balli and Bent E. Sørensen. Interaction effects in econometrics. *Empirical Economics*, 45(1):583–603, 2013.

[9] Rina Foygel Barber and Emmanuel J Candès. Controlling the false discovery rate via knockoffs. *The Annals of Statistics*, 43(5):2055–2085, 2015.

[10] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[11] Alexis Bellot and Mihaela van der Schaar. Conditional independence testing using generative adversarial networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[12] Malgorzata Bogdan, Ewout van den Berg, Weijie Su, and Emmanuel Candes. Statistical estimation and testing via the sorted L1 norm. *arXiv preprint arXiv:1310.1969*, 2013.

[13] Terry Brown. Silica exposure, smoking, silicosis and lung cancer—complex interactions. *Occupational Medicine*, 59(2):89–95, 03 2009.

[14] Damian Brzyski, Alexej Gossmann, Weijie Su, and Małgorzata Bogdan. Group SLOPE - adaptive selection of groups of predictors. *Journal of the American Statistical Association*, 114(525):419–433, 2019.

[15] Jessica Byerly, Gwyneth Halstead-Nussloch, Koichi Ito, Igor Katsyv, and Hanna Y. Irie. PRKCQ promotes oncogenic growth and anoikis resistance of a subset of triple-negative breast cancer cells. *Breast Cancer Research*, 18(1):95, 2016.

[16] Jessica H. Byerly, Elisa R. Port, and Hanna Y. Irie. PRKCQ inhibition enhances chemosensitivity of triple-negative breast cancer by regulating Bim. *Breast Cancer Research*, 22(1):72, 2020.

[17] Emmanuel Candes, Yingying Fan, Lucas Janson, and Jinchi Lv. Panning for gold: 'model-X' knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(3):551–577, 2018.

[18] Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

[19] Bijun Chen, Ruoshui Li, Silvia C. Hernandez, Anis Hanna, Kai Su, Arti V. Shinde, and Nikolaos G. Frangogiannis. Differential effects of smad2 and smad3 in regulation of macrophage phenotype and function in the infarcted myocardium. *Journal of Molecular and Cellular Cardiology*, 171:1–15, 2022.

[20] Tianyi Cheng, Peiying Chen, Jingyi Chen, Yingtong Deng, and Chen Huang. Landscape analysis of matrix metalloproteinases unveils key prognostic markers for patients with breast cancer. *Frontiers in Genetics*, 12, 2022.

[21] Christina Curtis, Sohrab P. Shah, Suet-Feung Chin, Gulisa Turashvili, Oscar M. Rueda, Mark J. Dunning, Doug Speed, Andy G. Lynch, Shamith Samarajiwa, Yinyin Yuan, Stefan Gräf, Gavin Ha, Gholamreza Haffari, Ali Bashashati, Roslin Russell, Steven McKinney, METABRIC Group, Anita Langerød, Andrew Green, Elena Provenzano, Gordon Wishart, Sarah Pinder, Peter Watson, Florian Markowetz, Leigh Murphy, Ian Ellis, Arnie Purushotham, Anne-Lise Børresen-Dale, James D. Brenton, Simon Tavaré, Carlos Caldas, and Samuel Aparicio. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403):346–352, 2012.

[22] Kun Dai, Hong-Yi Yu, and Qing Li. A semisupervised feature selection with support vector machine. *Journal of Applied Mathematics*, 64:141–158, 2013.

[23] Ran Dai and Rina Barber. The knockoff filter for FDR control in group-sparse and multitask regression. In *International Conference on Machine Learning (ICML)*, 2016.

[24] Mei Dong, Tam How, Kellye C. Kirkbride, Kelly J. Gordon, Jason D. Lee, Nadine Hempel, Patrick Kelly, Benjamin J. Moeller, Jeffrey R. Marks, and Gerard C. Blobe. The type III TGF-β receptor suppresses breast cancer progression. *The Journal of Clinical Investigation*, 117(1):206–217, 2007.

[25] Gary Doran, Krikamol Muandet, Kun Zhang, and Bernhard Schölkopf. A permutation-based kernel conditional independence test. In *Uncertainty in Artificial Intelligence (UAI)*, pages 132–141, 2014.

[26] Wei Bin Fang, Iman Jokar, An Zou, Diana Lambert, Prasanthi Dendukuri, and Nikki Cheng. CCL2/CCR2 chemokine signaling coordinates survival and motility of breast cancer cells through smad3 protein- and p42/44 mitogen-activated protein kinase (MAPK)-dependent mechanisms. *Journal of Biological Chemistry*, 287(43):36593–36608, 2012.

[27] Michael Y. Fessing, Ruzanna Atoyan, Ben Shander, Andrei N. Mardaryev, Vladimir V. Botchkarev Jr., Krzysztof Poterlowicz, Yonghong Peng, Tatiana Efimova, and Vladimir A. Botchkarev. BMP signaling induces cell-type-specific changes in gene expression programs of human keratinocytes and fibroblasts. *Journal of Investigative Dermatology*, 130(2):398–404, 2010.

[28] Edward I. George and Robert E. McCulloch. Approaches for bayesian variable selection. *Statistica Sinica*, pages 339–373, 1997.

[29] Christophe Ginestier, Suling Liu, Mark E. Diebel, Hasan Korkaya, Ming Luo, Marty Brown, Julien Wicinski, Olivier Cabaud, Emmanuelle Charafe-Jauffret, Daniel Birnbaum, Jun-Lin Guan, Gabriela Dontu, and Max S. Wicha. CXCR1 blockade selectively targets human breast cancer stem cells in vitro and in xenografts. *The Journal of Clinical Investigation*, 120(2):485–497, 2010.

[30] Santiago M. Gómez Bergna, Abril Marchesini, Leslie C. Amorós Morales, Paula N. Arrías, Hernán G. Farina, Víctor Romanowski, M. Florencia Gottardo, and Matias L. Pidre. Exploring the metastatic role of the inhibitor of apoptosis BIRC6 in breast cancer. *bioRxiv*, 2021.

[31] Daniel B. Graham and Ramnik J. Xavier. Pathway paradigms revealed from the genetics of inflammatory bowel disease. *Nature*, 578(7796):527–539, Feb 2020.

[32] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(1):1157–1182, 2003.

[33] Kai Han, Yunhe Wang, Chao Zhang, Chao Li, and Chao Xu. Autoencoder inspired unsupervised feature selection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[34] Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2005.

[35] Daniel Hernández-Lobato, José Miguel Hernández-Lobato, and Pierre Dupont. Generalized spike-and-slab priors for bayesian group feature selection using expectation propagation. *Journal of Machine Learning Research*, 14(7), 2013.

[36] John J. Irwin, Teague Sterling, Michael M. Mysinger, Erin S. Bolstad, and Ryan G. Coleman. ZINC: A free tool to discover chemistry for biology. *Journal of Chemical Information and Modeling*, 52(7):1757–1768, 2012.

[37] Paul Jaccard. The distribution of the flora in the alpine zone. *The New Phytologist*, 11(2):37–50, 1912.

[38] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations (ICLR)*, 2017.

[39] James Jordon, Jinsung Yoon, and Mihaela van der Schaar. KnockoffGAN: Generating knockoffs for feature selection using generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[40] Päivi Järvensivu, Taija Heinosalo, Janne Hakkarainen, Pauliina Kronqvist, Niina Saarinen, and Matti Poutanen. HSD17B1 expression induces inflammation-aided rupture of mammary gland myoepithelium. *Endocrine-Related Cancer*, 25(4):393 – 406, 2018.

[41] Tadashi Kato, Atsushi Yamada, Mikiko Ikehata, Yuko Yoshida, Kiyohito Sasa, Naoko Morimura, Akiko Sakashita, Takehiko Iijima, Daichi Chikazu, Hiroaki Ogata, and Ryutaro Kamijo. FGF-2 suppresses expression of nephronectin via JNK and PI3K pathways. *FEBS Open Bio*, 8(5):836–842, 2018.

[42] Matthew Kelly and Christopher Semsarian. Multiple mutations in genetic cardiovascular disease. *Circulation: Cardiovascular Genetics*, 2(2):182–190, 2009.

[43] Pora Kim, Feixiong Cheng, Junfei Zhao, and Zhongming Zhao. ccmGDB: a database for cancer cell metabolism genes. *Nucleic Acids Research*, 44(D1):D959–D968, 2015.

[44] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[45] Kenji Kira and Larry A. Rendell. A practical approach to feature selection. In *Machine Learning Proceedings*, pages 249–256. 1992.

[46] Theo A. Knijnenburg, Gunnar W. Klau, Francesco Iorio, Mathew J. Garnett, Ultan McDermott, Ilya Shmulevich, and Lodewyk F. A. Wessels. Logic models to predict continuous outputs based on binary inputs with an application to personalized cancer therapy. *Scientific Reports*, 6(1):36812, 2016.

[47] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1):273–324, 1997.

[48] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[49] Changhee Lee, Fergus Imrie, and Mihaela van der Schaar. Self-supervision enhanced feature selection with correlated gates. In *International Conference on Learning Representations (ICLR)*, 2022.

[50] Ismael Lemhadri, Feng Ruan, and Rob Tibshirani. LassoNet: Neural networks with feature sparsity. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

[51] Chad Leugers, Ju Yong Koh, Willis Hong, and Gloria Lee. Tau in MAPK activation. *Frontiers in Neurology*, 4, 2013.

[52] Xinghua Li, Weijiang Liang, Junling Liu, Chuyong Lin, Shu Wu, Libing Song, and Zhongyu Yuan. Transducin (β)-like 1 X-linked receptor 1 promotes proliferation and tumorigenicity in human breast cancer via activation of beta-catenin signaling. *Breast Cancer Research*, 16(5):465, 2014.

[53] Yifeng Li Li, Chih-Yu Chen, and Wyeth W. Wasserman. Deep feature selection: theory and application to identify enhancers and promoters. *Journal of Computational Biology*, 23(5):322–336, 2016.

[54] Faming Liang, Qizhai Li, and Lei Zhou. Bayesian neural networks for selection of drug sensitive genes. *Journal of the American Statistical Association*, 113(523):955–972, 2018.

[55] Ofir Lindenbaum, Uri Shaham, Erez Peterfreund, Jonathan Svirsky, Nicolas Casey, and Yuval Kluger. Differentiable unsupervised feature selection based on a gated laplacian. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[56] Huan Liu and Rudy Setiono. A probabilistic approach to feature selection - a filter solution. In *International Conference on Machine Learning (ICML)*, 1996.

[57] Ying Liu and Cheng Zheng. Deep latent variable models for generating knockoffs. *Stat*, 8(1):e260, 2019.

[58] Huanyu Lu, Yue Guo, Gaurav Gupta, and Xingsong Tian. Mitogen-activated protein kinase (MAPK): New insights in breast cancer. *Journal of Environmental Pathology, Toxicology and Oncology*, 38(1):51–59, 2019.

[59] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The Concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations (ICLR)*, 2017.

[60] Ramamurthy Mani, Robert P. St.Onge, John L. Hartman, Guri Giaever, and Frederick P. Roth. Defining genetic interaction. *Proceedings of the National Academy of Sciences*, 105(9):3461–3466, 2008.

[61] Pulak R. Manna, Ahsen U. Ahmed, Shengping Yang, Madhusudhanan Narasimhan, Joëlle Cohen-Tannoudji, Andrzej T. Slominski, and Kevin Pruitt. Genomic profiling of the steroidogenic acute regulatory protein in breast cancer: In silico assessments and a mechanistic perspective. *Cancers*, 11(5), 2019.

[62] Kevin McCloskey, Ankur Taly, Federico Monti, Michael P. Brenner, and Lucy J. Colwell. Using attribution to decode binding mechanism in neural network models for chemistry. *Proceedings of the National Academy of Sciences*, 116(24):11624–11629, 2019.

[63] Lisiane B. Meira, Antonio M.C. Reis, David L. Cheo, Dorit Nahari, Dennis K. Burns, and Errol C. Friedberg. Cancer predisposition in mutant mice defective in multiple genetic pathways: uncovering important genetic interactions. *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, 477(1):51–58, 2001.

[64] Jordi Merino, Marta Guasch-Ferré, Jun Li, Wonil Chung, Yang Hu, Baoshan Ma, Yanping Li, Jae H. Kang, Peter Kraft, Liming Liang, Qi Sun, Paul W. Franks, JoAnn E. Manson, Walter C. Willet, Jose C. Florez, and Frank B. Hu. Polygenic scores, diet quality, and type 2 diabetes risk: An observational study among 35,759 adults from 3 US cohorts. *PLOS Medicine*, 19(4):1–20, 04 2022.

[65] Sofia Papadimitriou, Andrea Gazzo, Nassim Versbraegen, Charlotte Nachtegael, Jan Aerts, Yves Moreau, Sonia Van Dooren, Ann Nowé, Guillaume Smits, and Tom Lenaerts. Predicting disease-causing variant combinations. *Proceedings of the National Academy of Sciences*, 116(24):11878–11887, 2019.

[66] Ui-Hyun Park, Mi Ran Kang, Eun-Joo Kim, Young-Soo Kwon, Wooyoung Hur, Seung Kew Yoon, Byoung-Joon Song, Jin Hwan Park, Jin-Taek Hwang, Ji-Cheon Jeong, and Soo-Jong Um. ASXL2 promotes proliferation of breast cancer cells by linking ERα to histone methylation. *Oncogene*, 35(28):3742–3752, 2016.

[67] Hanna M. Peltonen, Annakaisa Haapasalo, Mikko Hiltunen, Vesa Kataja, Veli-Matti Kosma, and Arto Mannermaa. Γ-secretase components as predictors of breast cancer outcome. *PLOS ONE*, 8(11), 2013.

[68] Bernard Pereira, Suet-Feung Chin, Oscar M. Rueda, Hans-Kristian Moen Vollan, Elena Provenzano, Helen A. Bardwell, Michelle Pugh, Linda Jones, Roslin Russell, Stephen-John Sammut, Dana W. Y. Tsui, Bin Liu, Sarah-Jane Dawson, Jean Abraham, Helen Northen, John F. Peden, Abhik Mukherjee, Gulisa Turashvili, Andrew R. Green, Steve McKinney, Arusha Oloumi, Sohrab Shah, Nitzan Rosenfeld, Leigh Murphy, David R. Bentley, Ian O. Ellis, Arnie Purushotham, Sarah E. Pinder, Anne-Lise Børresen-Dale, Helena M. Earl, Paul D. Pharoah, Mark T. Ross, Samuel Aparicio, and Carlos Caldas. The somatic mutation profiles of 2,433 breast cancers refine their genomic and transcriptomic landscapes. *Nature Communications*, 7(1):11479, 2016.

[69] Patrick C. Phillips. Epistasis — the essential role of gene interactions in the structure and evolution of genetic systems. *Nature Reviews Genetics*, 9(11):855–867, 2008.

[70] Michael W. Pickup, Laura D. Hover, Eleanor R. Polikowsky, Anna Chytil, Agnieszka E. Gorska, Sergey V. Novitskiy, Harold L. Moses, and Philip Owens. BMPR2 loss in fibroblasts promotes mammary carcinoma metastasis via increased inflammation. *Molecular Oncology*, 9(1):179–191, 2015.

[71] Barbara Maria Piskór, Andrzej Przylipiak, Emilia Dąbrowska, Iwona Sidorkiewicz, Marek Niczyporuk, Maciej Szmitkowski, and Sławomir Ławicki. Plasma level of MMP-10 may be a prognostic marker in early stages of breast cancer. *Journal of Clinical Medicine*, 9(12), 2020.

[72] Franck Rapaport, Emmanuel Barillot, and Jean-Philippe Vert. Classification of arrayCGH data using fused SVM. *Bioinformatics*, 24(13):i375–i382, 2008.

[73] Yaniv Romano, Matteo Sesia, and Emmanuel Candès. Deep knockoffs. *Journal of the American Statistical Association*, 115(532):1861–1872, 2020.

[74] Jakob Runge. Conditional independence testing based on a nearest-neighbor estimator of conditional mutual information. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 938–947, 2018.

[75] Benjamin Sanchez-Lengeling, Jennifer Wei, Brian Lee, Emily Reif, Peter Wang, Wesley Qian, Kevin McCloskey, Lucy Colwell, and Alexander Wiltschko. Evaluating attribution for graph neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[76] Fadil Santosa and William W. Symes. Linear inversion of band-limited reflection seismograms. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1307–1330, 1986.

[77] Rajat Sen, Ananda Theertha Suresh, Karthikeyan Shanmugam, Alexandros G Dimakis, and Sanjay Shakkottai. Model-powered conditional independence test. In *Advances in neural information processing systems (NeurIPS)*, 2017.

[78] Razieh Sheikhpour, Mehdi Agha Sarram, Sajjad Gharaghani, and Mohammad Ali Zare Chahooki. A survey on semi-supervised feature selection methods. *Pattern Recognition*, 64:141–158, 2017.

[79] Prajjal K. Singha, Srilakshmi Pandeswara, Hui Geng, Rongpei Lan, Manjeri A. Venkatachalam, Albert Dobi, Shiv Srivastava, and Pothana Saikumar. Increased smad3 and reduced smad2 levels mediate the functional switch of TGF-β from growth suppressor to growth and metastasis promoter through TMEPAI/PMEPA1 in triple negative breast cancer. *Genes & cancer*, 10(5-6):134, 2019.

[80] Mukund Sudarshan, Wesley Tansey, and Rajesh Ranganath. Deep direct likelihood knockoffs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[81] Mina Takahashi, Fumio Otsuka, Tomoko Miyoshi, Hiroyuki Otani, Junko Goto, Misuzu Yamashita, Toshio Ogura, Hirofumi Makino, and Hiroyoshi Doihara. Bone morphogenetic protein 6 (BMP6) and BMP7 inhibit estrogen-induced proliferation of breast cancer cells by suppressing p38 mitogen-activated protein kinase activation. *Journal of Endocrinology*, 199(3):445 – 455, 2008.

[82] Jiliang Tang, Salem Alelyani, and Huan Liu. Feature selection for classification: A review. *Data classification: Algorithms and applications*, page 37, 2014.

[83] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[84] Nicholas Turner, Alex Pearson, Rachel Sharpe, Maryou Lambros, Felipe Geyer, Maria A. Lopez-Garcia, Rachael Natrajan, Caterina Marchio, Elizabeth Iorns, Alan Mackay, Cheryl Gillett, Anita Grigoriadis, Andrew Tutt, Jorge S. Reis-Filho, and Alan Ashworth. FGFR1 amplification drives endocrine therapy resistance and is a therapeutic target in breast cancer. *Cancer Research*, 70(5):2085–2094, 2010.

[85] Dongfeng Wang, Jian Li, Fengling Cai, Zhi Xu, Li Li, Huanfeng Zhu, Wei Liu, Qingyu Xu, Jian Cao, Jingfeng Sun, and Jinhai Tang. Overexpression of MAPT-AS1 is associated with better patient survival in breast cancer. *Biochemistry and Cell Biology*, 97(2):158–164, 2019.

[86] Dongsheng Wang, Chenglong Zhao, Liangliang Gao, Yao Wang, Xin Gao, Liang Tang, Kun Zhang, Zhenxi Li, Jing Han, and Jianru Xiao. NPNT promotes early-stage bone metastases in breast cancer by regulation of the osteogenic niche. *Journal of Bone Oncology*, 13:91–96, 2018.

[87] Chang-Yuan Wei, Qi-Xing Tan, Xiao Zhu, Qing-Hong Qin, Fei-Bai Zhu, Qin-Guo Mo, and Wei-Ping Yang. Expression of CDKN1A/p21 and TGFBR2 in breast cancer and their prognostic significance. *International Journal of Clinical and Experimental Pathology*, 8(11):14619, 2015.

[88] Michael K. Wendt, Molly A. Taylor, Barbara J. Schiemann, Khalid Sossey-Alaoui, and William P. Schiemann. Fibroblast growth factor receptor splice variants are stable markers of oncogenic transforming growth factor $\beta 1$ signaling in metastatic breast cancers. *Breast Cancer Research*, 16(2):R24, 2014.

[89] Tim Wilson, Tim Holt, and Trisha Greenhalgh. Complexity and clinical care. *BMJ*, 323(7314):685–688, 2001.

[90] Xinxing Wu and Qiang Cheng. Algorithmic stability and generalization of an unsupervised feature selection algorithm. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[91] Yutaro Yamada, Ofir Lindenbaum, Sahand Negahban, and Yuval Kluger. Feature selection using stochastic gates. In *International Conference on Machine Learning (ICML)*, 2020.

[92] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. Invase: Instance-wise variable selection using neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

[93] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[94] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[95] Lingmin Zeng and Jun Xie. Group variable selection via SCAD-L2. *Statistics*, 48(1):49–66, 2014.

[96] Xin-Yue Zhang, Hsun-Ming Chang, Elizabeth L. Taylor, Rui-Zhi Liu, and Peter C. K. Leung. BMP6 downregulates GDNF expression through SMAD1/5 and ERK1/2 signaling pathways in human granulosa-lutein cells. *Endocrinology*, 159(8):2926–2938, 2018.

[97] Yong-ping Zhang, Wen-ting Na, Xiao-qiang Dai, Ruo-fei Li, Jian-xiong Wang, Ting Gao, Wei-bo Zhang, and Cheng Xiang. Over-expression of SRD5A3 and its prognostic significance in breast cancer. *World Journal of Surgical Oncology*, 19(1):260, 2021.

[98] Jidong Zhao, Ke Lu, and Xiaofei He. Locality sensitive semi-supervised feature selection. *Neurocomputing*, 71:1842—-1849, 2008.

[99] Ting Zhong, Feifei Xu, Jinhui Xu, Liang Liu, and Yun Chen. Aldo-keto reductase 1C3 (AKR1C3) is associated with the doxorubicin resistance in human breast cancer via PTEN loss. *Biomedicine & Pharmacotherapy*, 69:317–325, 2015.

[100] Nengfeng Zhou and Ji Zhu. Group variable selection via a hierarchical lasso and its oracle property. *arXiv preprint arXiv:1006.2871*, 2010.

[101] Guangyu Zhu and Tingting Zhao. Deep-gKnock: Nonlinear group-feature selection with deep neural networks. *Neural Networks*, 135:139–147, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Conclusion for details.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] In the Conclusion, we caution that as with any feature selection method, discovered features must be verified or evaluated by domain experts. This verification or evaluation might be costly, and should the method perform poorly, could result in wasted resources. In addition, without additional oversight (primarily in dataset construction but also when validating features), features that contain bias could remain and be identified by feature selection algorithms.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code is available at either of the following GitHub repositories: `https://github.com/a-norcliffe/Composite-Feature-Selection`, `https://github.com/vanderschaarlab/Composite-Feature-Selection`.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Hyperparameters for each experiment are provided in Table 4. Architecture details are provided in Appendix C and further experimental details are provided in Appendix D.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] All experiments are repeated 10 times and results are reported along with standard deviations.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] All experiments can be run easily on a commercially-available laptop. We provide further details of the compute resources used in Appendix D.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We used several existing methods and datasets (see Experiments). All benchmark methods and datasets are clearly cited.
   (b) Did you mention the license of the assets? [Yes] Licenses of assets (benchmark methods and datasets) is provided in Appendix D and F.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Code is available at either of the following GitHub repositories: `https://github.com/a-norcliffe/Composite-Feature-Selection`, `https://github.com/vanderschaarlab/Composite-Feature-Selection`.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] We only use publicly available, anonymized datasets. See Broader Impact - Datasets.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] We only use publicly available, anonymized datasets. See Broader Impact - Datasets.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

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

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

Broader Impact

Applications. Our work is focused on imposing additional structure on the traditional feature selection procedure. Feature selection and group feature selection help to improve a model's interpretability, from this we do not foresee negative impacts. One potentially significant issue is if biased data is used, feature selection and group feature selection may select features that are based on or serve to reinforce societal bias. Therefore, as mentioned in 6 (Conclusion), discovered features must be verified by domain experts to avoid this.

On the other hand, methods for understanding the importance of and interplay between features (such as our work) have potential for good. For example, if one were to discover the interactions between genes that leads to a particular disease this could greatly aid both screening and drug design. Other examples may be understanding complex interactions in economics, helping to understand how to tackle inequality; or interactions in climate science, helping to further our understanding of climate change. With respect to both positive and negative impacts, our work focuses more on introducing the problem of group feature selection, thus we do not see significant negative impact as a direct result of this paper.

Datasets. Three of our datasets do not contain sensitive data. These are the synthetic datasets, semi-synthetic chemistry datasets, and MNIST. We also use METABRIC, a dataset consisting of patients with breast cancer, containing genetic and medical information. Whilst this does contain sensitive information, this is a public dataset with anonymous data.

A Ablation Study

To explore the behavior of CompFS further, we conducted a sensitivity analysis of several hyperparameters. In particular, we investigated how many groups are used as the numbers of learners, β_R and β vary. As discussed in Section 5, the true number of composite features or groups is likely not known *a priori*, and thus models for composite feature selection must not depend too heavily on this. We examined how many learners are used (i.e. how many learners discover non-empty groups) on the Syn2 dataset. When ablating over β_R, we trained for 35 epochs, with β fixed at 1.0. When ablating over β we trained for 35 epochs keeping β_R fixed at 1.2. Each experiment was repeated 3 times. The results are shown in Figure 3. For larger values of β_R and β, the number of groups remains low, even when the number of learners far exceeds the true number of groups. As a consequence, if more learners are employed, larger values of β_R and β will likely be necessary.

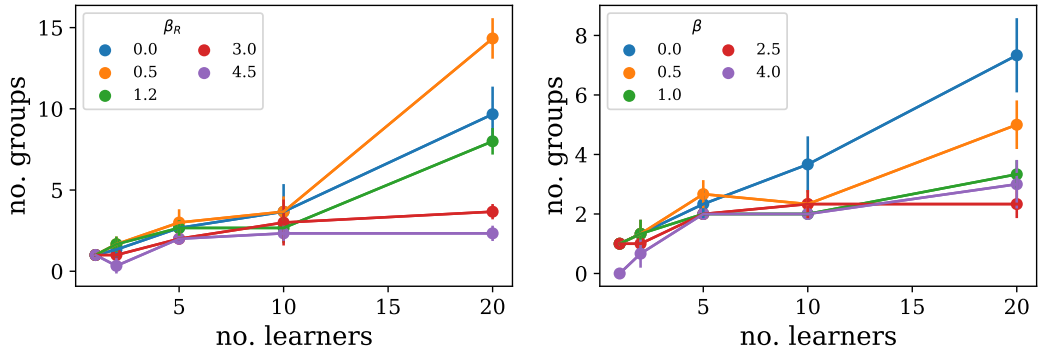


Figure 3: Ablation investigating how the number of learners in the ensemble and β_R & β hyperparameters affect the number of groups learnt by CompFS.

B Extended Related Work

Feature selection has an extensive history and proposed solutions can be generally categorized into three types based on how they interact with the learning method: wrapper [47], filter [56, 45], or embedded methods [76]. For more detailed discussion on traditional approaches to feature selection,

we refer the reader to the following review articles [82, 18]. In this section, we focus on advances in standard feature selection, which, while not the aim of our paper, provides useful background.

Recently, a number of deep learning embedded feature selection methods have been developed which exploit end-to-end learning to jointly learn a prediction network that only uses a subset of the available features. For example, the popular LASSO penalization [83] has been extended to neural networks in LassoNet [50]. One key challenge in deep learning approaches to feature selection is that selecting features is inherently a non-differentiable process. Methods have explored different approximation schemes, such as Lasso or elastic net penalization (DeepFS) [53], MCMC sampling (BNNsel) [54], continuous relaxation using independent Gaussian random variables (STG) [91], and reinforcement learning [92]

In the unsupervised domain, differentiable unsupervised feature selection (DUFS) [55] introduces a trainable Bernoulli gating mechanism into the Laplacian score [34], while [90] extends autoencoder-based feature selection [33] with algorithmic stability. Concrete Autoencoders (CAE) were proposed in [7], where autoencoders were employed to identify a pre-specified number of features that are sufficient for reconstructing the data. They employ a concrete selector layer based on Concrete random variables [59, 38]. By selecting stochastic linear combinations of the input variables during training, the layer converges to a discrete set of features, with the number of feature defined by the user. While CAEs typically use a reconstruction objective (i.e. unsupervised), they can also be trained in a supervised manner.

Additionally, several semi-supervised feature selection methods have been proposed [78] as extensions of traditional methods (e.g. [98, 22]), as well as self-supervised methods [49].

Finally, while not the focus of this paper, we briefly mention an alternate strand of literature related to feature selection, that of hypothesis testing and, in particular, conditional independence testing [25, 77, 74, 11].

[11]

C Architecture Details

CompFS consists of 4 different network components: (i) group selection probabilities $\boldsymbol{\pi}_i$, (ii) group encoders f_{θ_i}, (iii) group predictors h_{ϕ_i}, and (iv) aggregate or ensemble predictor h_ϕ. We use a fully-connected network as the baseline architecture for network components (ii)-(iv). Note that compared to traditional feature selection, the computational complexity/number of parameters scales linearly with the maximum permitted number of groups (in the same way that an ensemble scales with the number of members). Typically the number of groups will be relatively small, which should alleviate any practical concerns. The specific architectural details of CompFS are provided below:

- Stochastic Gating: For the relaxed Bernoulli sampling we use a temperature of 0.1 and for evaluation we use a threshold $\lambda = 0.7$ to determine whether a feature is relevant for the group.
- Group Encoders f_{θ_i}: Multi-layer perceptron with ReLU activations, and two hidden layers of a given hidden width.
- Group Predictors h_{ϕ_i}: Single linear layer, given by a separate weight matrix and bias for each group.
- Ensemble Predictor h_ϕ: The latent representation of each group encoder is passed through a separate linear layer, given by a weight matrix and bias term. These are then summed element-wise, yielding a permutation invariant aggregation. For simplicity we do not apply any further transformations in our experiments.

Further details regarding hyperparameters in each of the experiments can be found in Table 4.

D Experimental Details

Benchmark implementations. LASSO was implemented in PyTorch. Group LASSO was implemented using the *group-lasso* public library [5], which is provided under the MIT License. This requires hyperparameter β, determining the trade-off between sparse features and predictive power.

[5]The documentation can be found at: `https://group-lasso.readthedocs.io/en/latest/`

Group LASSO is trained with the fast iterative soft thresholding algorithm (FISTA) [10]. In order to run Group LASSO, we first form all possible combinations of groups of features, enumerating all groups of size 1 and 2. This allows us to treat the groups independently and run Group LASSO. This scales poorly requiring us to keep groups limited to a maximum size of 2. Additionally we eliminate the majority of the noise variables and use only 60 features in the synthetic experiments, giving Group LASSO a considerably easier task to tackle. Despite this, Group LASSO is still outperformed by CompFS. We use the publicly available implementations for both STG[6] and CAE[7]. Both STG and CAE are provided under MIT Licenses.

Hyperparameters. The hyperparameters used by each model in each experiment are given in Table 4. In Table 4, Reg refers to the regularisation parameter as follows. For CompFS, Reg refers to β. For both LASSO and Group LASSO, Reg refers to the scale factor that the $L1$ norm of the weights is multiplied by in the loss (often referred to as λ). For STG, Reg refers to λ, the scale factor for the $L1$ norm. For CAE, Reg refers to the number of features to use. Additionally for CompFS, β and β_R in this table are multiplied by the square root of the number of features for a given experiment. This allows the importance of having small groups and minimal overlap to increase with the dimensionality of the problem, but not too quickly. As a result, we observe that the hyperparameters are relatively stable across our experiments. We fixed the number of possible groups for CompFS as five for all experiments. This is *more* than the true number of groups, which means that CompFS must not only learn to select (composite) features, but also not to discover redundant features. STG uses two hidden layers. CAE uses two hidden layers in the decoder.

We train every model apart from Group LASSO with the ADAM optimizer [44], with the learning rates and decays given in Table 4. CompFS(5) is trained for 35 epochs, CompFS(1) is trained for 20 epochs, LASSO is trained for 8 epochs and Group LASSO is trained for 1000 iterations or until convergence. The experiments were all repeated 10 times to produce means and standard deviations. For all experiments we use a threshold of 0.7 to determine whether a feature is relevant for CompFS. For LASSO, feature i is relevant if $|w_i| > 0.01$, where $\mathbf{w}$ is the weight vector. If irrelevant, w_i is set to 0 during evaluation. For Group LASSO, group i is relevant if $||w_{\mathcal{G}_i}||_2 > 0.005$. If irrelevant, $w_{\mathcal{G}_i}$ is set to 0 during evaluation. When evaluating the composite features discovered by CompFS, we disregard repeated or empty groups, only considering unique non-empty ones.

Reproducibility. Code is available at either of the following GitHub repositories: `https://github.com/a-norcliffe/Composite-Feature-Selection`, `https://github.com/vanderschaarlab/Composite-Feature-Selection`. Most experiments were run locally on a machine with 8GB RAM, Intel Core i7-7500 HQ, NVIDIA GeForce MX150. The remainder (Group LASSO, STG, CAE, MNIST) were run on the free version of Google Colab.

[6] `https://github.com/runopti/stg`

[7] `https://github.com/mfbalin/Concrete-Autoencoders`

Table 4: Hyperparameters for each experiment.

TASK	MODEL	REG	β_E	β_R	BATCH SIZE	HIDDEN WIDTH	LEARNING RATE	LR DECAY
SYN1	COMPFS(5)	4.5	1.0	1.2	50	20	0.003	0.99
	COMPFS(1)	0.35	1.0	-	100	30	0.003	0.99
	LASSO	0.4	-	-	50	-	0.003	0.99
	GROUP LASSO	0.1	-	-	FULL	-	-	-
	STG	0.1	-	-	250	20	0.001	1.00
	CAE	2	-	-	250	200	0.001	1.00
SYN2	COMPFS(5)	4.5	1.0	1.2	50	20	0.003	0.99
	COMPFS(1)	0.025	1.0	-	100	30	0.003	0.99
	LASSO	0.4	-	-	50	-	0.003	0.99
	GROUP LASSO	0.04	-	-	FULL	-	-	-
	STG	0.1	-	-	250	20	0.001	1.00
	CAE	4	-	-	250	200	0.001	1.00
SYN3	COMPFS(5)	4.5	1.0	1.2	50	20	0.003	0.99
	COMPFS(1)	0.025	1.0	-	100	30	0.003	0.99
	LASSO	0.4	-	-	50	-	0.003	0.99
	GROUP LASSO	0.04	-	-	FULL	-	-	-
	STG	0.1	-	-	250	20	0.001	1.00
	CAE	3	-	-	250	200	0.001	1.00
SYN4	COMPFS(5)	4.5	1.0	1.2	50	20	0.003	0.99
	COMPFS(1)	0.35	1.0	-	100	30	0.003	0.99
	LASSO	0.4	-	-	50	-	0.003	0.99
	GROUP LASSO	0.04	-	-	FULL	-	-	-
	STG	0.1	-	-	250	20	0.001	1.00
	CAE	4	-	-	250	200	0.001	1.00
CHEM1	COMPFS(5)	2.0	1.0	1.2	20	20	0.003	0.99
	COMPFS(1)	0.4	1.0	-	20	30	0.003	0.99
	LASSO	0.4	-	-	20	-	0.003	0.99
	GROUP LASSO	0.04	-	-	FULL	-	-	-
	STG	0.1	-	-	250	20	0.001	1.00
	CAE	2	-	-	250	200	0.001	1.00
CHEM2	COMPFS(5)	3.4	1.0	1.2	20	20	0.003	0.99
	COMPFS(1)	0.4	1.0	-	20	30	0.003	0.99
	LASSO	0.2	-	-	20	-	0.003	0.99
	GROUP LASSO	0.04	-	-	FULL	-	-	-
	STG	0.1	-	-	250	20	0.001	1.00
	CAE	3	-	-	250	200	0.001	1.00
CHEM3	COMPFS(5)	2.0	1.0	1.2	20	20	0.003	0.99
	COMPFS(1)	0.7	1.0	-	20	30	0.003	0.99
	LASSO	0.2	-	-	20	-	0.003	0.99
	GROUP LASSO	0.04	-	-	FULL	-	-	-
	STG	0.1	-	-	250	20	0.001	1.00
	CAE	4	-	-	250	200	0.001	1.00

E Jaccard Index

The Jaccard Index (or Jaccard Similarity) [37] is a metric to estimate the similarity between two sets A and B, which returns a scalar value between 0 and 1. It is defined as the ratio of the number of elements in the intersection and the number of elements in the union of the two sets:

$$\mathcal{J}(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{16}$$

When carrying out standard feature selection this would work as a metric to compare the set of ground truth features to the set of discovered features. However, in the new problem of composite feature selection both the ground truth and the output are a set of sets, i.e. a set of groups of features. If we were to use this metric on the sets $\{\{1, 2\}, \{3, 4\}\}$ and $\{\{1, 2, 3\}, \{4\}\}$, Eq. (16) would return 0, because the two sets of sets share no elements, where the elements are the composite features. This is clearly undesirable behaviour as we can see qualitatively the learnt group structure is similar to the true group structure. Therefore, we generalise the Jaccard Similarity to the Group Similarity denoted by $\mathrm{G}_{\mathrm{Sim}}$:

$$\mathrm{G}_{\mathrm{sim}} = \frac{1}{\max(N, K)} \sum_{i=1}^{N} \max_{j \in [K]} \mathcal{J}(\mathcal{G}_i, \hat{\mathcal{G}}_j) \tag{17}$$

Here we calculate the Jaccard Similarity between all the learnt composite features and the true composite features. For each true composite feature we look for the learnt composite feature that matches it best, and take the sum for all the true composite features. We then normalise this sum by dividing by either the number of true composites or the number of discovered composites, whichever is larger. This firstly keeps $\mathrm{G}_{\mathrm{Sim}}$ between 0 and 1, with $G_{\mathrm{Sim}} = 0$ when none of the correct features are discovered, and 1 for perfect recovery of *only* the true group structure. However, if the true group structure is incidentally discovered, (if many composites are discovered for example), then the normalisation term punishes any additional structure which is incorrect. This prevents a method finding all combinations of true individual features. Further to this, it will reward the group structures that are similar more than those that are significantly different. To help understand the group similarity metric further we provide some examples in Table 5.

Table 5: Examples of Group Similarity calculations.

True Composites	**Candidate Composites**	**Sum**	**max(N, K)**	**$\mathrm{G}_{\mathrm{Sim}}$**
$\{1, 2\}, \{3, 4\}$	$\{1, 2\}, \{3, 4\}$	2	2	1
	$\{1, 2, 3, 4\}$	1	2	1/2
	$\{1, 2, 3\}, \{1, 4\}$	1	2	1/2
	$\{1\}, \{2\}, \{3\}, \{4\}$	1	4	1/4
	$\{1, 2\}, \{3, 4\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}$	2	6	1/3
$\{1\}, \{2\}, \{3, 4, 5\}$	$\{1, 2\}$	1	3	1/3
	$\{3\}, \{1, 3, 5\}$	5/6	3	5/18
	$\{6\}, \{7\}, \{8, 9, 10\}$	0	3	0
$\{1, 2\}, \{1, 3\}$	$\{1, 2\}, \{1, 3\}$	2	2	1
	$\{1, 2, 3\}$	4/3	2	2/3
	$\{1\}, \{2\}, \{3\}$	1	3	1/3

In order to provide a more complete picture of a model's group predictions, we also consider the feature level true positive and false discovery rates. This is analogous to using accuracy for classification as well as other metrics such as AUROC.

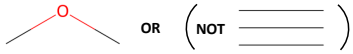
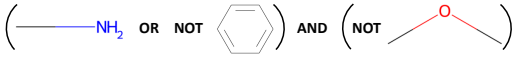
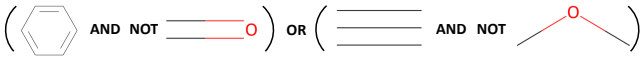
F Chemistry Datasets

The semi-synthetic molecular datasets employed are based on several of the datasets constructed by [62], which were released under the Apache 2.0 license.[8] The synthetic "binding logics" are expressed as a combination of molecular fragments that must either be present or absent for binding to occur and this is used to label molecules from the ZINC database [36]. The synthetic binding logics used in our paper are shown in Table 6 and cover a range of ground truth group structures. The logics correspond to the following combinations of molecular fragments: (1) Ether OR NO Alkyne; (2) (Primary Amine or NO Benzene) AND NO Ether (3) (Benzene AND NO Carbonyl) OR (Alkyne AND NO Ether). These were chosen from the larger set of synethic binding logics considered by [62] since they exhibit a group structure.

In order to reduce possible biases, [62] constructed each logic dataset to be balanced across the set of all binary combinations. For example, given a logic $A\&B$, the datasets contain an equal number of examples for every logic combination: $A\&B$, $A\&\neg B$, $\neg A\&B$ and $\neg A\&\neg B$. However, this induces significant class imbalance for the more complex binding logics. We used the same training and test sets determined by [62]. The number of examples varies for the different binding logics, with the training sets employed ranging from 3,862 to 14,769 molecules, and the test sets from 467 to 1832.

Molecules are featurized using a set of 84 functional groups, where feature $x_i = 1$ if the molecule contains functional group i and 0 otherwise. Ground truth indices in Table 6 correspond to the relevant indices in the molecular featurization.

Table 6: Binding Logics for Chemistry Datasets.

DATASET	BINDING LOGIC (VISUALIZATION)	GROUND TRUTH
CHEM1		{40}, {1}
CHEM2		{56, 18}, {40}
CHEM3		{18, 29}, {1, 40}

G Additional Baselines

Here we run further baselines on our experiments, including implementations details and hyperparameters as well as results.

Reduced CompFS. We include an ablation of our method, which uses only one group (CompFS(1)), which tests how well the method performs as a standard feature selection algorithm. Hyperparameters are given in Table 4. Note that due to the loss function used to train our method (in particular Eq. 11), we do not expected CompFS(1) to outperform state of the art tradition feature selection methods.

Unsupervised Concrete Autoencoder. We consider the unsupervised version of concrete autoencoder (unsup-CAE), where the minimum subset of features are used to reconstruct the entire input. Unsup-CAE uses the same hyperparameters as Sup-CAE, we evaluate model performance by looking at the reconstruction MSE.

Random Forests. Attempting to create grouping we use Random Forests. This model consists of a collection of 250 decision trees, with maximum depth 5. After training, the top 10 performing trees are considered. For each one the chosen features are those with a feature importance greater than 0.1. This collection of features is considered a group and all unique groups are used as the predicted composites. This was implemented using the scikitlearn implementation of Random Forest.

[8] Data from `https://github.com/google-research/graph-attribution/raw/main/data/all_16_logics_train_and_test.zip`.

Gradient Boosted Decision Trees. This is the same as the random forests benchmark, but here Gradient Boosted Decision Trees are used rather than simple Decision Trees. We refer to this method as GBDT. The same hyperparameters are used as Random Forests. This was implemented using the scikitlearn implementation of Gradient Boosted Decision Trees.

Clustering on Oracle Features. For the synthetic and semi-synthetic tasks, we take the true features and apply clustering to them (Oracle+Cluster). This attempts to group them based on their correlations rather than interactions. In each case we enforce the same number of clusters as number of true composites. The clustering is done using scikitlearn Feature Agglomeration.

Ensemble of STGs. Our final baseline is an ensemble of STGs (Ensemble STG). Here we use a collection of independent STGs. However, rather than their outputs being directly used as logits, they are instead all concatenated. Then this new larger hidden representation is passed through an MLP. The output of this is used to predict the logits which are used to train the ensemble. The regularisation loss remains the same. The features selected by each STG are then a composite feature. Our implementation uses two hidden layers for each individual STG and one hidden layer for the final MLP, ReLU activations are used. We use a hidden width of 50. For the tasks we train with 4 STGs, using $\lambda = 0.1$ for synthetic tasks and $\lambda = 4.0$ for the Chemistry tasks. We train using the ADAM optimizer with batchsizes of 100 for synthetic tasks and 500 for chemistry tasks, the learning rates are 0.001 for synthetic tasks and 0.01 for chemistry tasks.

Next we give the results of these new baselines alongside our main ones. Additional baselines are run 5 times.

G.1 Synthetic Task

We test all our benchmarks on the synthetic tasks from Section 5. The results are given in Table 7, we present them along with the main results for easy comparison.

CompFS(1) performs well on the synthetic tasks, showing it is able to carry out feature selection. STG performs as well as the Oracle on Syn1-3, demonstrating it as a state of the art feature selection method, however has high FDR on Syn4. Sup-CAE significantly underperforms other non-linear methods on Syn2 and Syn3, although performs well on Syn4, achieving the lowest FDR of all methods with non-zero TPR, due to having to select how many features are found. Unsup-CAE performs poorly across Syn1-3 because all features are independent, so it is impossible to reconstruct the full input from a subset. As expected, the reconstruction MSE almost exactly equals the variance of the standard normal distribution from which features are sampled. It performs slightly better on Syn4 due to correlation between features. Random Forests achieve perfect TPR, and on Syn 1-3 achieve reasonable group similarities compared to Oracle, showing it is possible to provide some grouping. However, due to the high correlation performs poorly on Syn 4. GBDT performs as well as Oracle on all tasks, However, whilst this is a strength through the lens of standard feature selection, it is a weakness in composite feature selection because no grouping has been provided. Oracle+Cluster trivially achieves the same TPR and FDR as Oracle. Additionally, it also occasionally achieve higher group similarity, this is because it does enforce some group structure. However, due to a lack of correlation between features, the grouping is inferior to that provided by CompFS. This is to be expected, as this groups based on correlations between features rather than their interactions. Ensemble STG performs the same as STG, showing that it has not been able to separate out the groups.

Noticeably, CompFS still performs very well, performing either best or close to best in terms of discovering group structure as measured by G_{Sim}, as well as on TPR and FDR.

Table 7: Performance on Synthetic Datasets including all benchmarks, values are recorded with their standard deviations. Performance is recorded as accuracy except for Unsup-CAE, where the reconstruction MSE is given.

Dataset	Model	TPR	FDR	G_{sim}	No. Groups	Accuracy (%) Or MSE
Syn1	CompFS(5)	100.0 ± 0.0	0.0 ± 0.0	0.91 ± 0.14	2.2 ± 0.4	98.9 ± 0.5
	Oracle	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	CompFS(1)	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	99.1 ± 0.8
	LASSO	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	81.8 ± 2.0
	Group LASSO	100.0 ± 0.0	0.0 ± 0.0	0.67 ± 0.00	3.0 ± 0.0	83.8 ± 1.4
	STG	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	97.8 ± 1.4
	Sup-CAE	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	97.8 ± 1.4
	Unsup-CAE	0.0 ± 0.0	100.0 ± 0.0	0.00 ± 0.00	1.0 ± 0.0	0.999 ± 0.005
	Random Forests	100.0 ± 0.0	0.0 ± 0.0	0.68 ± 0.03	2.8 ± 0.4	96.6 ± 3.9
	GBDT	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	Oracle+Cluster	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	2.0 ± 0.0	100.0 ± 0.0
	Ensemble STG	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	93.5 ± 0.8
Syn2	CompFS(5)	95.0 ± 15.0	0.0 ± 0.0	0.90 ± 0.20	1.8 ± 0.4	95.5 ± 5.4
	Oracle	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	CompFS(1)	85.0 ± 22.9	2.0 ± 6.0	0.48 ± 0.08	1.0 ± 0.0	90.2 ± 8.1
	LASSO	0.0 ± 0.0	0.0 ± 0.0	0.00 ± 0.00	0.0 ± 0.0	52.6 ± 2.9
	Group LASSO	0.0 ± 0.0	0.0 ± 0.0	0.00 ± 0.00	0.0 ± 0.0	52.2 ± 0.9
	STG	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	93.9 ± 2.2
	Sup-CAE	37.5 ± 31.7	42.5 ± 44.2	0.24 ± 0.20	1.0 ± 0.0	61.9 ± 12.8
	Unsup-CAE	0.0 ± 0.0	100.0 ± 0.0	0.00 ± 0.00	1.0 ± 0.0	0.995 ± 0.005
	Random Forests	100.0 ± 0.0	0.0 ± 0.0	0.28 ± 0.08	6.8 ± 1.5	62.9 ± 3.7
	GBDT	100.0 ± 0.0	0.0 ± 0.0	0.54 ± 0.08	2.2 ± 0.4	99.0 ± 0.0
	Oracle+Cluster	100.0 ± 0.0	0.0 ± 0.0	0.58 ± 0.00	2.0 ± 0.0	100.0 ± 0.0
	Ensemble STG	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	86.9 ± 0.7
Syn3	CompFS(5)	100.0 ± 0.0	0.0 ± 0.0	0.68 ± 0.05	1.3 ± 0.5	97.4 ± 1.1
	Oracle	100.0 ± 0.0	0.0 ± 0.0	0.67 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	CompFS(1)	96.7 ± 10.0	2.5 ± 7.5	0.65 ± 0.05	1.0 ± 0.0	94.5 ± 3.7
	LASSO	0.0 ± 0.0	0.0 ± 0.0	0.00 ± 0.00	0.0 ± 0.0	56.5 ± 4.0
	Group LASSO	0.0 ± 0.0	0.0 ± 0.0	0.00 ± 0.00	0.0 ± 0.0	54.6 ± 1.3
	STG	100.0 ± 0.0	0.0 ± 0.0	0.67 ± 0.00	1.0 ± 0.0	95.3 ± 1.7
	Sup-CAE	23.3 ± 31.6	66.7 ± 47.1	0.23 ± 0.31	1.0 ± 0.0	62.6 ± 12.6
	Unsup-CAE	0.0 ± 0.0	100.0 ± 0.0	0.00 ± 0.00	1.0 ± 0.0	0.999 ± 0.005
	Random Forests	100.0 ± 0.0	0.0 ± 0.0	0.54 ± 0.06	3.6 ± 0.5	51.3 ± 0.6
	GBDT	100.0 ± 0.0	0.0 ± 0.0	0.67 ± 0.00	1.0 ± 0.0	98.5 ± 0.0
	Oracle+Cluster	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	2.0 ± 0.0	100.0 ± 0.0
	Ensemble STG	100.0 ± 0.0	0.0 ± 0.0	0.67 ± 0.00	1.0 ± 0.0	84.6 ± 1.5
Syn4	CompFS(5)	90.0 ± 12.2	51.9 ± 13.8	0.47 ± 0.20	2.5 ± 0.7	95.8 ± 1.8
	Oracle	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	CompFS(1)	75.0 ± 19.4	63.1 ± 6.0	0.18 ± 0.03	1.0 ± 0.0	92.8 ± 5.9
	LASSO	0.0 ± 0.0	0.0 ± 0.0	0.00 ± 0.00	0.0 ± 0.0	51.8 ± 3.2
	Group LASSO	0.0 ± 0.0	10.0 ± 31.6	0.00 ± 0.00	0.1 ± 0.3	53.0 ± 1.1
	STG	100.0 ± 0.0	66.7 ± 0.0	0.17 ± 0.00	1.0 ± 0.0	94.2 ± 2.1
	Sup-CAE	72.5 ± 14.2	16.7 ± 14.7	0.39 ± 0.08	1.0 ± 0.0	72.2 ± 13.2
	Unsup-CAE	32.5 ± 26.5	67.5 ± 26.5	0.15 ± 0.14	1.0 ± 0.0	0.983 ± 0.007
	Random Forests	100.0 ± 0.0	63.5 ± 2.1	0.12 ± 0.04	9.8 ± 0.4	82.1 ± 0.7
	GBDT	100.0 ± 0.0	0.0 ± 0.0	0.44 ± 0.00	3.0 ± 0.0	98.5 ± 0.0
	Oracle+Cluster	100.0 ± 0.0	0.0 ± 0.0	0.58 ± 0.00	2.0 ± 0.0	100.0 ± 0.0
	Ensemble STG	100.0 ± 0.0	66.7 ± 0.0	0.17 ± 0.00	1.0 ± 0.0	81.2 ± 0.4

G.2 Chemistry Task

Next we test all our benchmarks on the chemistry tasks from Section 5. The results are given in Table 8, we present them along with the main results for easy comparison.

Table 8: Performance on Chemistry Datasets, values are recorded with their standard deviations. Performance is recorded as accuracy except for Unsup-CAE, where the reconstruction MSE is given.

Dataset	Model	TPR	FDR	G_{sim}	No. Groups	Accuracy (%) Or MSE
Chem1	CompFS(5)	100.0 ± 0.0	0.0 ± 0.0	0.82 ± 0.20	1.9 ± 0.5	100.0 ± 0.0
	Oracle	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	CompFS(1)	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	LASSO	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	75.8 ± 0.0
	Group LASSO	100.0 ± 0.0	0.0 ± 0.0	0.67 ± 0.00	3.0 ± 0.0	100.0 ± 0.0
	STG	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	Sup-CAE	50.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	75.8 ± 0.0
	Unsup-CAE	0.0 ± 0.0	100.0 ± 0.0	0.00 ± 0.00	1.0 ± 0.0	0.042 ± 0.003
	Random Forests	100.0 ± 0.0	20.0 ± 16.3	0.50 ± 0.06	1.6 ± 0.5	98.37 ± 2.0
	GBDT	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	Oracle+Cluster	100.0 ± 0.0	0.0 ± 0.0	1.00 ± 0.00	2.0 ± 0.0	100.0 ± 0.0
	Ensemble STG	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
Chem2	CompFS(5)	100.0 ± 0.0	0.0 ± 0.0	0.72 ± 0.24	2.2 ± 0.6	100.0 ± 0.0
	Oracle	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	CompFS(1)	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	LASSO	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	81.6 ± 0.0
	Group LASSO	100.0 ± 0.0	0.0 ± 0.0	0.40 ± 0.00	5.0 ± 0.0	81.6 ± 0.0
	STG	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	Sup-CAE	66.7 ± 0.0	0.0 ± 0.0	0.42 ± 0.00	1.0 ± 0.0	80.9 ± 9.5
	Unsup-CAE	0.0 ± 0.0	100.0 ± 0.0	0.00 ± 0.00	1.0 ± 0.0	0.040 ± 0.003
	Random Forests	100.0 ± 0.0	25.0 ± 0.0	0.47 ± 0.04	2.4 ± 0.5	85.1 ± 0.5
	GBDT	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	Oracle+Cluster	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	2.0 ± 0.0	100.0 ± 0.0
	Ensemble STG	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
Chem3	CompFS(5)	100.0 ± 0.0	7.3 ± 11.7	0.62 ± 0.17	2.4 ± 0.5	100.0 ± 0.0
	Oracle	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	CompFS(1)	100.0 ± 0.0	3.3 ± 10.0	0.48 ± 0.05	1.0 ± 0.0	100.0 ± 0.0
	LASSO	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	87.4 ± 5.2
	Group LASSO	100.0 ± 0.0	20.0 ± 0.0	0.20 ± 0.00	10.0 ± 0.0	91.5 ± 0.0
	STG	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	Sup-CAE	62.5 ± 13.2	23.3 ± 17.5	0.37 ± 0.07	1.0 ± 0.0	77.8 ± 11.0
	Unsup-CAE	25.0 ± 16.7	72.5 ± 17.1	0.11 ± 0.07	1.0 ± 0.0	0.029 ± 0.002
	Random Forests	100.0 ± 0.0	46.8 ± 5.2	0.15 ± 0.01	8.0 ± 0.9	95.9 ± 1.7
	GBDT	100.0 ± 0.0	0.0 ± 0.0	0.50 ± 0.00	1.0 ± 0.0	100.0 ± 0.0
	Oracle+Cluster	100.0 ± 0.0	0.0 ± 0.0	0.58 ± 0.00	2.0 ± 0.0	100.0 ± 0.0
	Ensemble STG	100.0 ± 0.0	0.0 ± 0.0	0.58 ± 0.00	2.0 ± 0.0	100.0 ± 0.0

STG performs as well as Oracle, further demonstrating its ability as a standard feature selection method. However, like Oracle, it does not offer any further insight into the structure of the discovered features. Sup-CAE does not perform well as it often fails to find all ground truth features. As in the synthetic tasks, Unsup-CAE performs poorly because the ability to reconstruct (we see from the MSE that this is achieved) does not equate to being able to predict, and therefore it finds non-predictive features. Due to high correlation between the functional gorups, Random Forests has the highest FDR (apart from Unsup-CAE), but it also achieves perfect TPR. As a result of many incorrect features being found, Random Forests has low group similarity. GBDT does as well as Oracle, however, again it only finds a single group, providing no information on the grouping. Orace+Cluster is able to achieve a groups similarity higher than Oracle on Chem3 due to adding grouping, however it is not as high as CompFS due to grouping according to correlations rather than interactions. Ensemble STG performs in the same way as on the synthetic tasks. It achieves the same results as STG, but is unable to provide any grouping, except on Chem3, where some grouping was discovered, In this case

it grouped all 4 features in one group and 3 of these in a second group, which is not the correct group structure, hence has lower G_{sim} than CompFS.

Just as in the synthetic tasks, we see that CompFS consistently achieves the highest G_{sim}, showing it is able to separate features into groups based on their interactions.

These additional benchmarks show us that even when it is possible to construct a model that can return groups of features, the groups are often not correct. However, CompFS is constantly able to achieve high G_{sim}, showing the particular loss function and architecture work well for discovering composite features.

G.3 MNIST

We investigate CompFS on the MNIST dataset [48], where the ground truth group structure is *unknown*. Typically in images, pixels do not have any specific meaning; this is because we can translate an image slightly and the image is effectively unchanged to a human, but the pixels have entirely different values. We use MNIST as a qualitative example because the images are all centered and at the same scale. We consider CompFS, STG, Supervised CAE and Unsupervised CAE, all discovered pixels are given in Figure 2.

Implementation. To train CompFS, we use 4 learners, with a hidden width of 256, temperature of 0.1, $\beta_R = \beta = 0.18$. To select the features in each groups, we choose the top 15 features, those with the 15 highest π values. The model was trained for 50 epochs with batchsize 500 and learning rate 0.001. The learnt groups each contain 15 features respectively with only 2 shared between any of the four, making 58 in total. For STG, we use $\lambda = 4.0$ a batchsize of 500 and train with the ADAM optimizer with learning rate 0.001. STG discovered 46 pixels. For CAE we limit it to find 45 pixels, we train with a batchsize of 250 using the ADAM optimizer with learning rate 0.001.

We see what we would expect, all feature selection models find pixels near the centre and spread out, to sample as much of the image as possible.

Evaluation. To fairly evaluate the discoveries, we train new models that use the discovered pixels as input only. For the individual groups, their union and the baselines we use an MLP with ReLU activations, two hidden layers and a hidden width of 200. For CompFS, we use the CompFS architecture as described in Appendix C. We use a hidden width of 50 to make the number of parameters comparable to the other baselines. We train using the ADAM optimizer with a learning rate of 0.001 and batchsize of 250 for 30 epochs. The final accuracies are given in Table 9.

Table 9: Accuracy on the MNIST dataset.

Model	No. Pixels	Accuracy
Group 1	15	78.00%
Group 2	15	74.00%
Group 3	15	72.00%
Group 4	15	81.00%
CompFS	58	94.00%
Union	58	95.00%
STG	46	95.00%
Sup-CAE	45	95.00%
Unsup-CAE	45	95.00%

We see that the individual groups have low accuracies with the highest being group 4 at 81%. Models trained on the union of these features, as well as those found by STG and CAE all achieve an accuracy of 95%. Noteably, CompFS performs almost as well as the other baselines, despite having fewer parameters and a significantly constrained architecture. This demonstrates that the grouping does not drastically affect performance, despite enforcing constraints on how the features are used by the model.

Finally, we consider the class level accuracies of each of these models. We plot the total test accuracies and the class-level test accuracies in Figure 4. We see that in all groups there are digits which are not accurately classified. Whereas CompFS, STG and CAE all are able to recognise digits relatively

uniformly. This shows that despite individual groups performing poorly, each one can be seen as providing some new information, and the ensemble combines this to make an overall prediction.

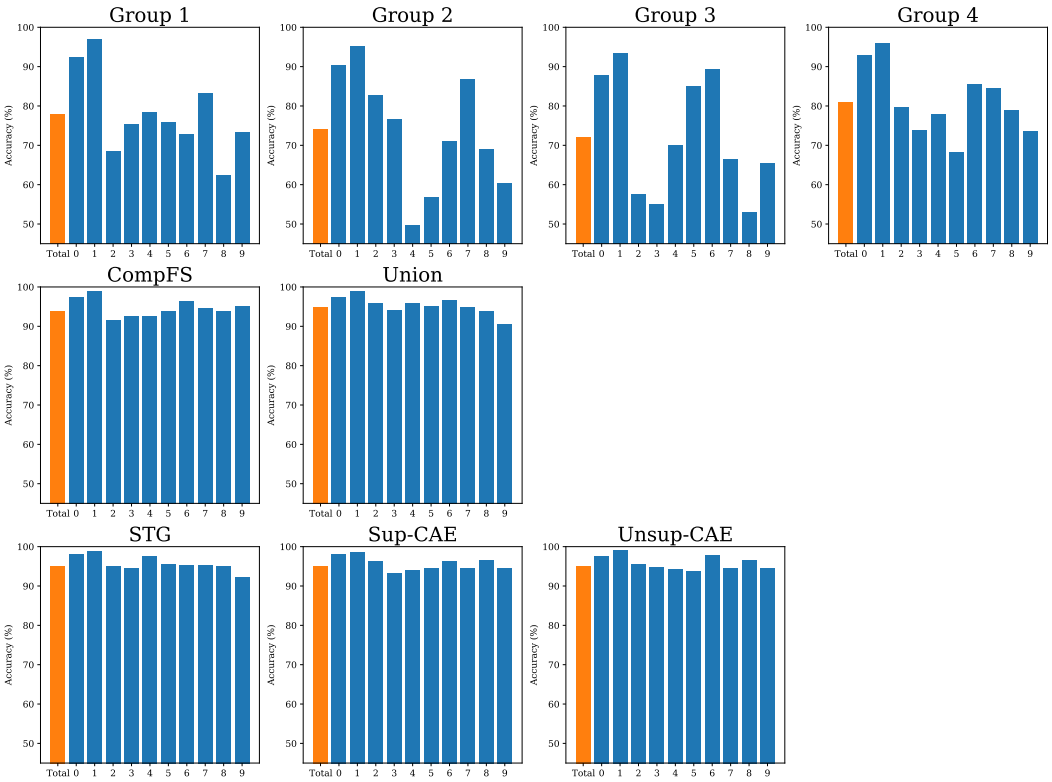


Figure 4: Class-level and overall MNIST accuracies of classifiers trained with the features found by each model. Note the scale of the y-axis does not start at 0, but at 45.

H METABRIC

In this section we provide additional details validating the features identified by CompFS on the METABRIC dataset.[9]

Individual features. We first provide validation for the individual features discovered by CompFS in Table 10. We found supporting evidence for 24 of the 25 features discovered by CompFS.

Table 10: Groups discovered by CompFS on the METABRIC dataset.

GROUP	FEATURE	REF.	GROUP	FEATURE	REF.	GROUP	FEATURE	REF.
GROUP 1	PSENEN	[67]	GROUP 2	BMP6	[81]	GROUP 3	BMPR2	[70]
	CXCR1	[29]		MAPK1	[58]		MMP12	[20]
	DLEC1	[1]		SMAD3	[79]		ASXL2	[66]
	MMP15	[20]		MAPT	[85]		BIRC6	[30]
	SRD5A3	[97]		PRKCQ	[16]		STAR	[61]
GROUP 4	CDKN1A	[87]	GROUP 5	BMP4	[3]			
	FGFR1	[84]		MMP10	[71]			
	TGFBR3	[24]		TBL1XR1	[52]			
	NPNT	[86]		USH2A				
	AKR1C3	[99]		HSD17B1	[40]			

Crucially, within each group, we found evidence of the joint importance of features, validating the groups discovered by CompFS. Discovered relationships are provided in Table 11.

Table 11: Supporting evidence for interactions between features within groups discovered by CompFS on the METABRIC dataset.

GROUP	FEATURES	REF.
GROUP 1	PSENEN, CXCR1	[5]
GROUP 2	BMP6, SMAD3	[19]
	BMP6, MAPK1	[96]
	MAPK1, SMAD3	[26]
	MAPK1, MAPT	[51]
	MAPK1, PRKCQ	[15]
GROUP 3	ASXL2, BIRC6	[43]
GROUP 4	FGFR1, TGFBR3	[88]
	FGFR1, NPNT	[41]
GROUP 5	BMP4, MMP10	[27]
	BMP4, HSD17B1	[4]

[9] Data from https://www.kaggle.com/datasets/raghadalharbi/breast-cancer-gene-expression-profiles-metabric