Composite Feature Selection Using Deep Ensembles

Anonymous Author(s)
Affiliation
Address
email

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; additionally, we can rank the groups based on their predictive power for downstream decision making. Furthermore, we propose a new metric to measure similarity between discovered groups and the ground truth. We test our model on multiple synthetic tasks, semi-synthetic chemistry datasets and image datasets to demonstrate its utility.

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 [9].

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 [30]. 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 predictive models can be complex, hard to interpret, and not amenable to the generation of hypotheses that can be experimentally tested [20]. This limits the impact such models can have in furthering scientific understanding across many domains where variables are known to interact, such as genetics [30, 31, 28], medicine [38, 6], and economics [3].

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

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

selection has also been applied in situations where there is extensive domain knowledge regarding the group structure [32] 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. Doing so offers deeper insights into *why* specific features are important than standard feature selection, with the goal of improving data-driven discovery.

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. (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.

2 Related Work

Significant attention has been placed on feature selection, with a range of solutions including traditional methods (e.g. [25, 21, 13]) and deep learning approaches (e.g. [24, 39, 2]). Several of these approaches have been extended to select predefined groups instead of individual features. For example, LASSO [35, 37] is a linear feature selection method that uses an L1 penalty to impose sparsity among coefficients. Group LASSO [40] generalises this to allows 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 [5, 23], have been similarly extended to group feature selection to give Group-SLOPE [7]. Further examples of group feature selection using adapted loss functions are SCAD-L2 [42] and hierarchical LASSO [43].

Feature selection for linear regression can also be approached from a Bayesian perspective by employing latent variables $w_i \in \{0, 1\}$ to indicate whether feature i is relevant to the model or not [11]. This class of approaches has been generalised to the group setting such that the latent variable $\boldsymbol{w}$ does not encode the relevance of each feature, but of each group from a predefined set [14].

Finally, the Knockoff procedure [4, 8] is a separate feature selection method that provides guarantees on the False Discovery Rate that has been utilized by a number of feature selection methods [18, 26, 33, 36]. Generalizations of the Knockoff procedure to the group setting also exist [10, 44].

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 between features or restrictions on groups, such as requiring that the candidate groups 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 the features into the maximum

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

number of subsets, $\mathcal{G}_i \subset [p]$, where complex interactions may happen within groups but only simple interactions between groups, 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 [12], 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 $\mathcal{F}$ 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 $\bar{\mathbf{x}}$). (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 to not only 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, but make the same prediction independent of x_3. Then we wish to group x_1, x_2 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. The model is valid when only simple interactions happen between groups and interactions within groups can be arbitrarily complex.

This raises the question, what is a simple interaction? We can view each group as contributing a different piece of information to the final prediction. When combining groups, we require order not to matter, as otherwise more complex interactions exist between groups. Therefore we must combine the representations using a *permutation invariant aggregator*. Separately, we can also view our model as acting on a *set* of groups [41], which naturally leads to permutation invariant aggregation.

It is well established that any permutation invariant aggregation function can be decomposed as $\rho[\sum \phi(\cdot)]$, for an appropriate choice of ρ and ϕ. 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, softmax for instance. 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 $f(\mathbf{x}) = \tilde{\rho}\big[\sum_i \tilde{f}_i(\mathbf{x}_{\mathcal{G}_i})\big]$, this gives a definition for what a valid model structure can be 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 only 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* $M \in \{0,1\}^{N \times p}$, 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 M. We can approximately solve (5) by solving the optimisation 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 remains the necessity simply to identify predictive features, which is already an NP-hard problem [1]. 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 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.

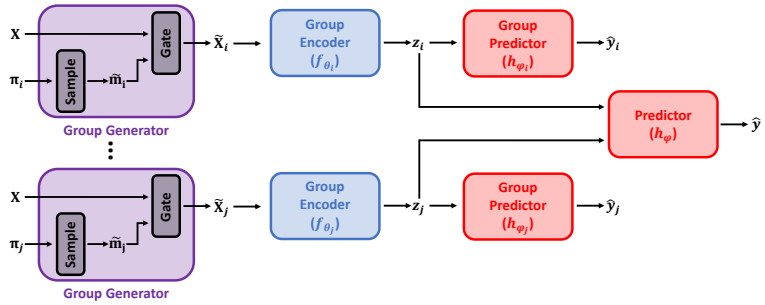

Figure 1: An illustration of CompFS. We use an ensemble of group selection models to discover composite features and an aggregate predictor to combine these features when issuing predictions.

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 (1). We achieve this by solving (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 (3) directly is not possible since the sampling step has no differentiable inverse. Instead, we use the relaxed Bernoulli distribution [27, 17] 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 [27]:

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

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 (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. 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 $\{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ 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:

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

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

4.3 Loss Functions

Group Selection Models. The individual learners can be trained to perform (traditional) feature selection (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 (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 feature selection models by minimizing a standard prediction loss:

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

where ℓ_Y is the same as in (11).

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 (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 (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 attempt to solve the traditional feature selection problem ($\mathcal{L}_{\mathcal{G}_i}$), which requires the group predictor to be accurate while selecting a minimal number of features. (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 calculating the gate vector $\mathbf{m}_i$ as follows: $m_{k,i} = 1$, if $\pi_{k,i} > \lambda$ 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. Specific architectural details are given in App. B. Additional information regarding experiments, benchmarks, and datasets can be found in App. C. Additional ablations and sensitivity analysis are in App. A.

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. As a result, we do not conduct an exhaustive comparison with traditional feature selection, since the upper bound on the performance of any such method is Oracle. However, for illustrative purposes, we include comparisons to a linear feature selection method ("LASSO", [37]) and an ablation of CompFS where we only allow a single group ("CompFS(1)").[2] Additional comparisons to STG [39] and Concrete Autoencoder [2] can be found in App. F. Finally, we compare with Group LASSO [40], 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.

Metrics. Since 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 [16] 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. D for additional details together with examples.

TPR and FDR assess whether the ground truth features have been correctly identified but do not take into account the group structure, which is our main goal. On the other hand, G_{sim} takes both into account and requires both the correct features to be identified and the underlying grouping to be uncovered. Finally, models should have strong predictive performance, which we assess via accuracy.

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 across the 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 the model when groups overlap, and whether it can split into two overlapping groups of two features each, rather than one group with three features.
- **(Syn4)** $y = 1$ if $x_1 x_4 > 0.30$ or $x_7 x_{10} > 0.30$, 0 otherwise. The ground truth groups are $\{\{1, 4\}, \{7, 10\}\}$. This task is equivalent to **Syn2**, however, here the features exhibit strong correlation. We enforce a block diagonal covariance matrix, with 1 on the diagonals, and 0.99 on the off diagonal entries in the blocks. The block matrices are 3×3 and cover the first 12 features. This task demonstrates the difficulty of carrying out group feature selection (and indeed standard feature selection) when the features are highly correlated.

[2] We use the notation CompFS(N) to refer to CompFS where N weak learners are used in the ensemble.

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
	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
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
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
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

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 20000 samples during train and 200 to test. Each experiment is repeated 10 times. For all experiments we use a threshold of $\lambda = 0.7$ to determine whether a feature is included by CompFS.

Analysis. On both Syn1 and Syn2, CompFS achieves high TPR with no false discoveries (0% FDR) and significantly higher G_{sim} than oracle feature selection. 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 allowed 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, outperforming the ablation of our method (CompFS(1)) by all metrics 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.

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 [29], some of which were also used by [34].[3]

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 [15]. Each logic includes up to four functional groups (Table 5). 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. E. Finally, note that many molecular fragments correlate with the binding logic but are not themselves responsible for binding.

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

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
	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
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
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

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.

MNIST. Finally, we investigate CompFS on MNIST [22], where the ground truth group structure is *unknown*. CompFS achieves overall accuracy of 95.2%, with four groups using 38 features ($< 5\%$ of pixels) on average. This is comparable to other studies [2], with individual learners achieving c. 80% accuracy and only minor overlap. Thus, while individual learners might not be able to solve the problem themselves, they provide different pieces of information to an overall ensemble. We visualise the pixels that each learner uses in Figure 2. We see that learners target different aspects of the problem; in particular, groups 1 and 2 have distinct features, with group 1 focusing more on the left and bottom to detect loops, while group 2 on the top right thus cannot make this distinction. See App. F.2 for additional details and comparisons.

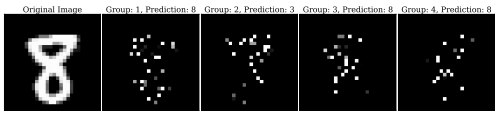

Figure 2: Groups discovered by CompFS predicting an MNIST 8. Qualitatively, group 1, 3 and 4 can detect loops, distinguishing 8 from 3.

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. As with any feature selection method, discovered features must be verified or evaluated by domain experts; however, we believe the additional structure provided by composite feature selection could be of significant benefit to a wide variety of practitioners.

References

[1] 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.

[2] Muhammed Fatih Balın, Abubakar Abid, and James Zou. Concrete autoencoders: Differentiable feature selection and reconstruction. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 444–453. PMLR, 09–15 Jun 2019.

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

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

[5] 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.

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

[7] 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.

[8] 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.

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

[10] Ran Dai and Rina Barber. The knockoff filter for fdr control in group-sparse and multitask regression. In *International conference on machine learning*, pages 1851–1859. PMLR, 2016.

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

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

[13] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. *In Proceedings of the 18th Conference on Neural Information Processing Systems (NIPS 2005)*, 2005.

[14] 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.

[15] 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, Jul 2012.

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

[17] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *In Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*, 2017.

[18] 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*, 2018.

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

[20] 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, Nov 2016.

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

[22] 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.

[23] Ma lgorzata Bogdana, Ewout van den Bergb, Weijie Suc, and Emmanuel J Candesc. Statistical estimation and testing via the ordered l1 norm. 2013.

[24] 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.

[25] Huan Liu and Rudy Setiono. A probabilistic approach to feature selection - a filter solution. *In Proceedings of the 13th International Conference on Machine Learning (ICML 1996)*, 1996.

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

[27] C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: A continuous relaxation of discrete random variables. *In Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*, 2017.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

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

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

[34] 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 H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5898–5910. Curran Associates, Inc., 2020.

[35] 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.

[36] Mukund Sudarshan, Wesley Tansey, and Rajesh Ranganath. Deep direct likelihood knockoffs. *Advances in neural information processing systems*, 33:5036, 2020.

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

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

[39] Y. Yamada, O. Lindenbaum, S. Negahban, and Y. Kluger. Feature selection using stochastic gates. *In Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, 2020.

[40] 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.

[41] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets. *arXiv preprint arXiv:1703.06114*, 2017.

[42] Lingmin Zeng and Jun Xie. Group variable selection via scad-l 2. *Statistics*, 48(1):49–66, 2014.

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

[44] 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] We provide all code needed to reproduce all results in the supplemental material.
   (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 3. Architecture details are provided in Appendix B and further experimental details are provided in Appendix C.
   (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 C.
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 C and E.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] The code to run our experiments is included in the supplementary material. The code will also be released publicly after the review period.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
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]