Instance-wise Feature Grouping

*Aria Masoomi[1], *Chieh Wu[1], Tingting Zhao[1], Zifeng Wang[1], Peter Castaldi[2], and Jennifer Dy[1]

masoomi.a@northeastern.edu, wu.chie@northeastern.edu, t.zhao@northeastern.edu
zifengwang@ece.neu.edu, repjc@channing.harvard.edu, jdy@ece.neu.edu
[1]*Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, US*
[2]*Channing Division of Network Medicine, Brigham and Women's Hospital, Boston, MA, US*

Abstract

In many learning problems, the domain scientist is often interested in discovering the groups of features that are redundant and are important for classification. Moreover, the features that belong to each group, and the important feature groups may vary per sample. But what do we mean by feature redundancy? In this paper, we formally define two types of redundancies using information theory: *Representation* and *Relevant redundancies*. We leverage these redundancies to design a formulation for instance-wise feature group discovery and reveal a theoretical guideline to help discover the appropriate number of groups. We approximate mutual information via a variational lower bound and learn the feature group and selector indicators with Gumbel-Softmax in optimizing our formulation. Experiments on synthetic data validate our theoretical claims. Experiments on MNIST, Fashion MNIST, and gene expression datasets show that our method discovers feature groups with high classification accuracies.

1 Introduction

Data samples are typically represented by features that domain experts assume to be important for a learning problem; however, not all features are important. The goal of feature selection is to select which features are needed to improve learning performance. Moreover, knowing which features are important helps in understanding learning algorithms.

Traditionally, *Feature Selection* algorithms find a *global* set of features for the entire data [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. While knowing the most important global features are useful, feature importance may vary across the entire population. For example in images, while one set of pixels may help us identify a shoe, a vastly different set of pixels would be required to identify a shirt. From this observation, there is an additional need for *Feature Selection* to be on a case-by-case basis, an approach also known as *Instance-wise Feature Selection*. A novel concept that has only been recently investigated in the context of explaining black-box models [11, 12, 13, 14]. Learning saliency maps [15] in some ways also provide some form of instance-wise feature importance by highlighting (weighting) important pixels in an image.

While *Instance-wise Feature Selection* focuses on each feature's relationship to its labels, it ignores the interaction among features. Multiple features may be equally important and yet redundant in relation to each other. Traditional feature selection algorithms (such as LASSO [16]) tend to select just one of these redundant features. However, in some domains such as gene expression applications, we are interested not only in which genes (features) are important but also in which genes interact together for disease prediction. Therefore, in addition to *Instance-wise Feature Selection*, we wish to also group the features based on their relationship with each other and to the label. There exist

[*] Signifies equal contribution.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

methods like group Lasso (GLasso) [17] that selects which feature groups are important given a predefined grouping. Yet, in many applications the feature groups are unknown. Thus, methods that learn feature groups have been proposed [18, 19, 20, 21]. While these methods perform group feature selection, the groups are global and not instance-wise. In contrast to these approaches, this paper introduces *instance-wise* methods that can learn the feature group structure and identify its importance for prediction from an information theory perspective. We refer to this approach as *Instance-wise Feature Grouping*.

Our Contribution. We introduce a novel method for learning instance-wise feature grouping, the *group Interpreter (gI)*. Our formulation is made possible by our theoretical contribution of defining the concept of redundancy in this setting. Leveraging mutual information's ability to measure dependency, we formally define two types: *Representation Redundancy* captures the dependency between features while *Relevant Redundancy* captures the dependency between features and its corresponding labels. We prove how these redundancies can be captured and describe the mechanisms by which information is preserved. Our analysis leads to a lower bound to identify the number of groups for each sample. Moreover, we provide a practical algorithm that approximates mutual information (MI) through a variational lower bound. The algorithm also learns a mapping function that identifies the most important feature groups on a sample by sample basis. Finally, our theories are experimentally verified on both synthetic and real data from ongoing research. Indeed, our method reveals the difference in gene expression based on smoking status. We make the source code publicly available at `https://github.com/ariahimself/Instance-wise-Feature-Grouping`.

Related Work. Many traditional *global* feature selection utilizes MI as criterion for selection (as it is a natural criterion for measuring dependency among random variables). However, in global feature selection, the goal is to find the minimal subset of features relevant for prediction [4, 5, 6, 22, 1, 23]. A way to achieve finding this minimal subset is to maximize feature relevance while minimizing feature redundancy [24, 25]. Note that they wish to *remove* redundancy. In contrast, our goal is to learn which features group (i.e., cluster) together, where we define similarity of features based on *redundancy*. For example, mRMR [25] maximizes feature relevance while minimizing feature redundancy. If features F1 and F2 are highly dependent and relevant to prediction, only one will be chosen. In contrast, gI would select both as a group, *highlighting to domain scientists that these two features are both relevant and redundant to each other*. Unlike traditional feature clustering, our goal is to learn feature similarity not just based on their redundancy with each other (representation redundancy) but also on their redundancy based on their prediction ability (relevant redundacy). We formally define these concepts in this paper.

Among other global feature group learning methods, Chormunge and Jena [19] learn feature groups based on k-means clustering then apply gLasso; Bilevel Learning [20] learns the feature groups through a multi-task learning setting using bilevel optimization; OSCAR [18] automatically learns the feature groups by encouraging equality in the magnitude of each pair of variables. All these group feature selection methods are *global*; whereas, our proposed method gI learns feature groups *instance-wise*.

2 Ingredients of Feature Group Learning

Overall Framework. We summarize the overall network framework of our method in Figure 1, followed by a description of each component in this section.

Given data set $\mathbf{X} \in \mathbb{R}^{n \times d}$ with n samples and d features, and let $\mathbf{Y} \in \mathbb{R}^{n}$ be its corresponding labels; the i^{th} sample input and its label are denoted as $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. Our goal is to separate the features into k non-overlapping groups and select the m most important groups for each sample. We learn for each sample a matrix G to indicate each feature's group membership. The G matrix is specifically constrained such that $G \in \{0,1\}^{k \times d}$ where $G_{i,j} = 1$ if the j^{th} feature of a sample belongs to the i^{th} group. After compressing the features into k groups, we also learn a vector $\mathbf{s} \in \{0,1\}^k$ where $\mathbf{s}_\mu = 1$ if the μ group is among the m most important groups.

The Group Membership Matrix G**.** We denote $\mathcal{G}$ as a random variable over a set of all possible G matrices where $\mathcal{G} \in \{G \in \{0,1\}^{k \times d} | \sum_{i=1}^{k} G_{ij} = 1\}$. This allows us to generate instance-wised G matrices for each sample by learning $P(\mathcal{G}|\mathbf{X})$, and use its most likely outcome as G where $G = \arg\max_{\mathcal{G}} P(\mathcal{G}|\mathcal{X})$. To learn G, we propose to train a non-traditional *autoencoder* ψ_{θ_G} that

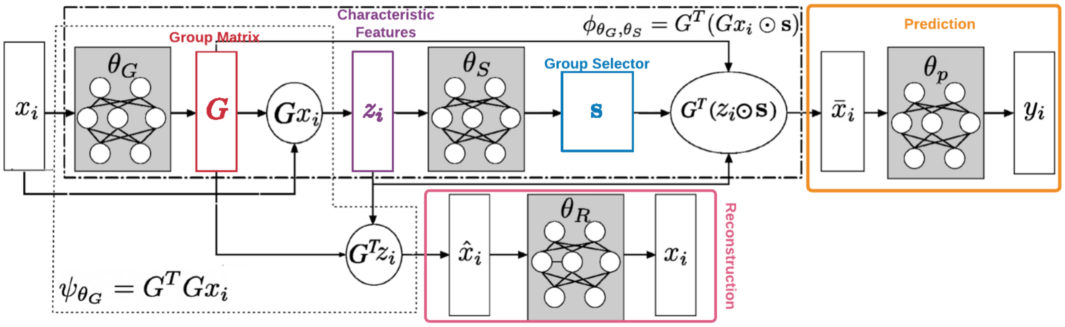

Figure 1: Flowchart of our instance-wise feature grouping framework.

maps the data $\mathbf{X}$ into a low dimensional embedding $\mathbf{Z} \in \mathbb{R}^{n\times k}$ with $\hat{\mathbf{X}} \in \mathbb{R}^{n\times d}$ as its decoded output where $\psi_{\theta_G}(\mathbf{X}) = \hat{\mathbf{X}}$. The *encoder* and *decoder* functions are denoted as $T_G : \mathbb{R}^d \to \mathbb{R}^k$ and $T_G^+ : \mathbb{R}^k \to \mathbb{R}^d$ where for a given sample i, $z_i = T_G(x_i) = Gx_i$, $\hat{x}_i = T_G^+(z_i) = G^T z_i$, and $\hat{x}_i = \psi_{\theta_G}(x_i) = T_G^+ \circ T_G(x_i) = G^T G x_i$. Note that each feature of z_i is a summation of only features of the same group, therefore, each feature encapsulates the characteristics of its corresponding group; accordingly, we refer to them as *characteristic features*.

The Group Selector s. Each G matrix is coupled with its own m-hot vector $\mathbf{s}$ that indicates the m most important groups. By defining the random variable $\mathcal{S} \in \{\mathbf{s} \in \{0,1\}^k | |\mathbf{s}| = m\}$, we also learn $\mathbf{s}$ indirectly by learning the distribution $P(\mathcal{S}|\mathbf{Z})$, where $\mathbf{s} = \arg\max_{\mathbf{S}} P(\mathcal{S}|\mathbf{Z})$. This is accomplished given a 2nd autoencoder $\phi_{\theta_S,\theta_G}(x_i) = G^T(Gx_i \odot \mathbf{s})$ which selects m *characteristic features* that corresponds to the m most important groups, where $\odot$ is an element wise product or Hadamard product. Hence, given $\mathbf{X}$ we have $\phi_{\theta_S,\theta_G}(\mathbf{X}) = \bar{\mathbf{X}}$ where the i_{th} row is $\bar{x}_i$.

Defining Feature Redundancy. Intuitively, features can be redundant if it is highly dependent on another set of features, we call this *Representation Redundancy*. Simultaneously, features can also be redundant if their inclusion does not improve the data/label dependency, i.e., given the occurrence of a feature, additional features may not provide any extra label-predicting information; we call this *Relevant Redundancy*. Formally, let $\mathcal{X}_j$ be a random variable representing the j^{th} feature, and let $\mathcal{X} = \{X_1, \dots, X_d\}$ be a set of all features with the cardinality of $|\mathcal{X}|$. By leveraging mutual information (MI, I), we define the two redundancies below.

Definition 1. *Feature X_j is Representation Redundant with respect to a set of random variables $\mathbf{Z}$ iff*

$$I(X_j; \mathcal{X}) \neq 0 \quad \text{and} \quad I(X_j; \mathcal{X}|\mathbf{Z}) = 0. \tag{1}$$

Definition 2. *Feature X_j is Relevant Redundant with respect to a set of random variables $\mathbf{Z}$ iff*

$$I(X_j; \mathbf{Y}) \neq 0 \quad \text{and} \quad I(X_j; \mathbf{Y}|\mathbf{Z}) = 0. \tag{2}$$

Note that in Def. (1), while condition $I(X_j; \mathcal{X}) \neq 0$ is always true since $X_j \in \mathcal{X}$, it is nevertheless included to preserve the symmetry with Def. (2). Following these definitions, we present our method, the Group Interpreter (gI), which implicitly learns G and $\mathbf{s}$ by maximizing

$$\max_{\theta_G,\theta_S} I(\hat{\mathbf{X}}; \mathbf{X}) + \lambda I(\bar{\mathbf{X}}; \mathbf{Y}), \quad \text{s.t:} \quad \hat{\mathbf{X}} = \psi_{\theta_G}(\mathbf{X}), \quad \bar{\mathbf{X}} = \phi_{\theta_S,\theta_G}(\mathbf{X}). \tag{3}$$

This objective is theoretically motivated by Defs. 1 and 2. Indeed, the $\hat{\mathbf{X}}$ that maximizes $I(\hat{\mathbf{X}}; \mathbf{X})$ captures *Representation Redundancy* while $I(\bar{\mathbf{X}}; \mathbf{Y})$ identifies the optimal $\bar{\mathbf{X}}$ to capture *Relevant Redundancy*. The control parameter λ then balances the two criteria. We formally prove these claims in the following two theorems with their proof included in App. C.

Theorem 1. *The maximum mutual information $I(\hat{\mathbf{X}}; \mathbf{X})$ is achieved if and only if its characteristic features $\mathbf{Z}$ induced by the model makes $\mathbf{X}$ representative redundant based on Def. (1), i.e.*

$$\begin{gathered} \max_G I(\hat{\mathbf{X}}; \mathbf{X}) = I(\mathbf{X}; \mathbf{X}) \iff \min_G I(\mathbf{X}; \mathbf{X}|\mathbf{Z}) = 0, \\ \textit{s.t.} \;\; G \in \{0,1\}^{k\times d}, \; \sum_{i=1}^{k} G_{ij} = 1, \; \mathbf{Z} = T_G(\mathbf{X}), \; \hat{\mathbf{X}} = \psi_{\theta_G}(\mathbf{X}). \end{gathered} \tag{4}$$

Theorem 2. *The maximum mutual information $I(\bar{\mathbf{X}}; \mathbf{X})$ is achieved if and only if its m-selected characteristic features $\mathbf{Z} \odot \mathbf{s}$ induced by the model makes $\mathbf{X}$ relevant redundant based on Def. (2), i.e.*

$$\max_G I(\bar{\mathbf{X}}; \mathbf{Y}) = I(\mathbf{X}; \mathbf{Y}) \iff \min_G I(\mathbf{X}; \mathbf{Y} | \mathbf{Z} \odot \mathbf{s}) = 0,$$
$$\textit{s.t.} \;\; G \in \{0,1\}^{k \times d}, \; \sum_{i=1}^{k} G_{ij} = 1, \; \mathbf{Z} = T_G(\mathbf{X}), \;\; \bar{\mathbf{X}} = \phi_{\theta_S,\theta_G}(\mathbf{X}), \mathbf{s} \in \{0,1\}^k, \; |\mathbf{s}| = m. \tag{5}$$

Approximating Mutual Information. Since the various distributions required to compute MI are difficult to obtain, we instead maximize MI's variational lower bound [11] as a surrogate. We provide here a summary of the key formulations while leaving the detail derivations to App. G. First, we solve Eq. (3) by first simplifying it into expectations

$$\max_{\theta_G,\theta_S} \quad E_{\mathbf{X},\hat{\mathbf{X}}}[\log(P(\mathbf{X}|\hat{\mathbf{X}})] + \lambda E_{\mathbf{Y},\bar{\mathbf{X}}}[\log(P(\mathbf{Y}|\bar{\mathbf{X}})] \quad \text{s.t: } \hat{\mathbf{X}} = \psi_{\theta_G}(\mathbf{X}), \; \bar{\mathbf{X}} = \phi_{\theta_S,\theta_G}(\mathbf{X}) \tag{6}$$

This objective can be approximated by computing its empirical estimate using samples from $P(\mathbf{X}|\hat{\mathbf{X}})$ and $P(\mathbf{Y}|\bar{\mathbf{X}})$. We generate $\hat{\mathbf{X}}, \bar{\mathbf{X}}$ samples via ancestral sampling [26] from

$$P(\hat{\mathbf{X}}|\mathbf{Z}, \mathcal{G}) P(\mathbf{Z}|\mathcal{G}, \mathbf{X}) \; \underline{P(\mathcal{G}|\mathbf{X})} \; P(\mathbf{X}), \tag{7}$$
$$P(\bar{\mathbf{X}}|\mathbf{Z} \odot \mathbf{s}, \mathcal{G}) P(\mathbf{Z} \odot \mathbf{s}|\mathcal{S}, \mathbf{Z}) \underline{P(\mathcal{S}|\mathbf{Z})} P(\mathbf{Z}|\mathcal{G}, \mathbf{X}) \underline{P(\mathcal{G}|\mathbf{X})} P(\mathbf{X}). \tag{8}$$

However, since both $P(\mathbf{X}|\hat{\mathbf{X}})$ and $P(\mathbf{Y}|\bar{\mathbf{X}})$ are unknown, we further use their variational lower bound to approximate their distributions via two additional networks. Specifically, we use $Q_{\theta_R}(\mathbf{X}|\psi_{\theta_G}(\mathbf{X}))$ to approximate $P(\mathbf{X}|\hat{\mathbf{X}})$, and $Q_{\theta_P}(\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X}))$ for $P(\mathbf{Y}|\bar{\mathbf{X}})$. This affords us the advantage of combining the four networks (ψ_{θ_G}, ϕ_{θ_S,θ_G}, Q_{θ_R}, and Q_{θ_P}) into a large single network and jointly optimize them via Stochastic Gradient Descent (SGD). The resulting formulation becomes

$$\min_{\theta_G,\theta_S,\theta_P,\theta_R} \quad \sum_{i=1}^{n} ||x_i - Q_{\theta_R}(\psi_{\theta_G}(x_i))||^2 - \lambda \sum_{i=1}^{n} p(y_i) \log(Q_{\theta_P}(y_i|(\phi_{\theta_S,\theta_G}(x_i))). \tag{9}$$

Solving Eq. (9) relies on drawing samples from $P(\mathcal{G}|\mathbf{X})$ and $P(\mathcal{S}|\mathbf{Z})$. However, since G and s are constrained to be indicators, how do we enforce the categorical constraint on the output of ψ_{θ_G} and ϕ_{θ_S,θ_G}? We clarify how adding a Gumbel-softmax layer [27] achieves this in the next section.

Gumbel-Softmax. Standard networks cannot perform backpropagation through samples. Gumbel-softmax overcome this obstacle by generating differentiable samples from a categorical distribution. Leveraging this technique, we sample a k-dimensional vector ϵ from a Gumbel distribution where its i^{th} element is sampled via $\epsilon_i = -\log(-\log u_i), u_i \sim \text{Uniform}(0,1)$. This enables us to apply the reparameterization trick [28], which consequently samples from a concrete distribution, $C \sim \text{Concrete}(\log p_1, ..., \log p_k)$, where the i^{th} element is computed with

$$C_i = \frac{\exp\{(\log p_i + \epsilon_i)/\tau\}}{\sum_{j=1}^{k} \exp\{(\log p_j + \epsilon_j)/\tau\}} \quad \text{s.t.} \lim_{\tau \to 0} P(C_i = 1) = \frac{p_i}{\sum_{j=1}^{k} p_j}. \tag{10}$$

The sharpness of the concrete distribution is controlled by τ; where as $\tau \to 0$, the concrete random variable approaches to the categorical distribution as defined in Eq. (10). Therefore, θ_G in Fig. 1 represents the combination of a network $Q_{\theta_G} : \mathbb{R}^d \mapsto \mathbb{R}^{k \times d}$ with a Gumbel-softmax layer. Since Q_{θ_G} outputs a dimension of $k \times d$, the output can be reorganized into d columns of size k vectors, where the i^{th} column represents the group membership probability $[p_1, ..., p_k]^T$ for the i^{th} feature. By passing each column into the Gumbel-softmax layer, it consequently generates a one-hot vector for each column of the G matrix, representing samples from $P(\mathcal{G}|\mathbf{X})$. Similarly, θ_S consists of $Q_{\theta_S} : \mathbb{R}^k \mapsto \mathbb{R}^k$ with a Gumbel-softmax layer. However, an m-hot vector is generated by repeating Gumbel-softmax m times. Specifically, let each trial be C^t, then s is generated by

$$C^t \sim \text{Concrete}(Q_{\theta_S}), \text{ for } t = 1, \ldots m, \quad \mathbf{s} = [\mathbf{s}_1, \ldots, \mathbf{s}_k]^T, \quad \mathbf{s}_j = \max_t C^t_j. \tag{11}$$

Discovering the Number of Groups. Instead of randomly guessing the number of groups, k, is there a theoretical guideline? We tackle this question from an information-theoretic perspective, by asking if there exists a minimum k such that all relevant information is preserved. To state the question precisely, what is the minimum m and k such that $\mathrm{I}(\mathbf{X};\mathbf{Y}) = \mathrm{I}(\phi_{\theta_S,\theta_G}(\mathbf{X});\mathbf{Y})$?

Since we compress the original features into characteristics features and then remove the least important groups, how can information retention be possible? By studying the simpler case where all groups are kept, we identified a set of conditions which this becomes possible and discovered a lower bound for k. Specifically, we simplify the problem by letting $m = k$ such that $\psi_{\theta_G} = \phi_{\theta_S,\theta_G}$, then we study if T_G^+ and T_G individually preserves information. Conceptually, since $\psi_{\theta_G} = T_G^+ \circ T_G$, information is preserved if T_G^+ and T_G both preserve information. This intuition is supported by Kraskov et al. [29]: they show that MI is invariant under diffeomorphism mappings. Therefore, we investigate if T_G and T_G^+ are diffeomorphisms and formalize these findings in the following two lemmas with their proof in App. B.

Lemma 1. *The decoder $T_G^+ : \mathbf{Z} \to \mathrm{Im}(T_G^+)$ is a Diffeomorphism map.*

Lemma 2. *If $k < d$ then the mapping $T_G : R^d \to R^k$ is not injective, thus **not** a diffeomorphism.*

Since k is always less than d when features are grouped together, our analysis proves that ψ_{θ_G} cannot be a diffeomorphism; a disappointing result. Yet, we note that while having diffeomorphism guarantees information preservation, nothing is stated about non-diffeomorphism mappings. Indeed, by digging deeper, we found that information preservation is still possible under certain non-diffeomorphism conditions. Specifically, we prove that relevant information can still be preserved if k is sufficiently large, i.e., larger than the number as defined by Eq. (63). In fact, in these cases, we proved the existence of a matrix G such that $I(\hat{\mathbf{X}};\mathbf{Y}) = I(\mathbf{X};\mathbf{Y})$. We formally state this finding in Theorem 3; the proof can be found in App. D.

Theorem 3. *Let $\mathcal{X} = \{X_1, \ldots, X_d\}$ be a random variable that consists of all features , let relevant features $\mathcal{U}$ be*

$$\mathcal{U} = \{X_j | \;\; I(X_j;\mathbf{Y}) \neq 0 \vee \exists \mathcal{A} \subseteq \mathcal{X}\, I(X_j;\mathbf{Y}|\mathcal{A}) \neq 0\}, \tag{12}$$

and let irrelevant features be $\mathcal{U}^c$, then, $\exists G \in \mathcal{G}$ such that $I(T_G(\mathbf{X});\mathbf{Y}) = I(\mathbf{X};\mathbf{Y})$ if

$$k \geq |\mathcal{U}| + \mathbb{1}(|\mathcal{U}| \neq d) \tag{13}$$

where $\mathbb{1}(|\mathcal{U}| \neq d)$ is an indicator function equal to one when $|\mathcal{U}| \neq d$ and zero when $|\mathcal{U}| = d$.

While Theorem 3 provides a theoretical bound for k, in practice, the computation of $\mathcal{U}$ assumes prior access to complex posterior distributions. Since this assumption is rarely true in practice, we provide an alternative bound that only requires the correlation coefficient between the features and the labels. We formally state this theorem and its corollaries below with their proofs in App. E.

Theorem 4. *Given ρ as the correlation measure and $\mathcal{C} = \{X_j | \rho(X_j;\mathbf{Y}) \neq 0 \,\vee\, \exists \mathcal{A}\, \rho(X_j;\mathbf{Y}|\mathcal{A}) \neq 0\}$ then $|\mathcal{C}| \leq |\mathcal{U}|$.*

Corollary 4.1. *Theorems 3 and 4 yields a lower bound for k where $|\mathcal{C}| + \mathbb{1}(|\mathcal{C}| \neq d) \leq k$.*

Corollary 4.2. *For Gaussian distributions the inequality turns into equality where $|\mathcal{C}| = |\mathcal{U}|$.*

By leveraging Corollary 4.1, a more tractable set $\mathcal{C}$ can be obtained in place of $\mathcal{U}$ to bound k.

Computational and Memory Complexities. Since our algorithm can be solved via SGD, gI has efficient memory and computational complexities of $O(kd^2)$ and $O(nkd^2)$ respectively. For a detailed derivation of these complexities, refer to App. H.

Feature Selection vs Explaining Black-Box Models. Due to the common confusion between feature selection, and black-box explanatory models (BEM), we emphasize that our focus is feature selection. Our method, gI, learns a classifier $Q_{\theta_p}(y|\phi_{\theta_S,\theta_G}(x))$ via feature grouping that approximates the true underlying posterior $P(\mathbf{Y}|\mathbf{X})$. Note that one can easily extend gI to explain black-box models by changing $P(\mathbf{Y}|\mathbf{X})$ to a complex black-box learned classifier $P_M(\mathbf{Y}|\mathbf{X})$ (e.g., neural networks [30], random forest [31]) similar to Chen et al. [11]; where, $Q_{\theta_p}(y|\phi_{\theta_S,\theta_G}(x))$ now approximates $P_M(\mathbf{Y}|\mathbf{X})$ by learning from training data with $\mathbf{Y}$ generated from the output of $P_M(\mathbf{Y}|\mathbf{X})$ for each x_i. Although gI can be easily extended to BEM, we leave this extension for future research.

3 Experiments

Datasets. We validate the theoretical claims with nine synthetic datasets constructed from a combination of three *Representation* (D_1, D_2, D_3) and three *Relevance* (R_1, R_2, R_3) redundancy patterns as shown in Table 1. Recall that X_j indicates the jth feature. For *Representation Redundancy* (D patterns), the features within the same parentheses are correlated with each other. For *Relevance Redundancy* (R patterns) the $P(\mathbf{Y} = 1|\mathbf{X})$ is directly proportional to a function of the features indicated. We generate 100000 training, 1000 validation, and 1000 test samples for each combination. For each combination, we evaluate gI's ability to correctly identify the number of groups (k), the redundancy patterns, and classification results.

We also evaluate our method on a real-world gene expression data as quantified by RNA sequencing from the COPDGene Study, an observational study to identify genomic markers associated with chronic obstructive pulmonary disease (COPD) [32]. The dataset is divided into a training and test set of 1500 and 407 patients along with the expression of 439 most relevant genes based on Gene Ontology categories [33]. We additionally test on benchmark image datasets from MNIST, and Fashion MNIST (F-MNIST) [34, 35] to evaluate our method's ability to generate visual results.

D_1	$(X_1, X_2), (X_3, X_4)$
D_2	$(X_1, X_3), (X_2, X_4)$
D_3	$(X_1, X_3, X_4), (X_2)$
R_1	$P(Y = 1\|X) \propto e^{X_1 * X_3}$
R_2	$P(Y = 1\|X) \propto e^{\sum_{i=1}^{4} X_i^2 - 4}$
R_3	$P(Y = 1\|X) \propto e^{-\sin(2X_1) + 2\|X_2\| + X_3 + \exp(-X_4 - 2.4)}$

Table 1: Synthetic data generation patterns

Model	MNIST-2	MNIST-10	F-MNIST
gI	96.7 ± 0.2	**91.6 ± 0.85**	94.6 ± 0.6
L2X	97.1 ± 0.5	80.5 ± 2.5	96.0 ± 0.6
shap	**99.24 ± 0.46**	90.8 ± 1.9	94.45 ± 2.62
INV	91.23 ± 3.48	77.94 ± 2.35	89.63 ± 3.45
Lasso	96.01 ± 0.2	86.03 ± 0.02	**96.7 ± 0.0**
Group Cluster	94.36 ± 0.05	85.0 ± 0.09	92.04 ± 0.06
OSCAR	95.56 ± 0.31	90.94 ± 0.27	95.0 ± 0.3
OWL	95.8 ± 0.25	90.92 ± 0.31	94.90 ± 0.16
LPA	95.63 ± 0.56	87.72 ± 1.23	94.83 ± 0.97

Table 2: gI $m = 1$, $k = 2$, image Classification accuracy comparison.

Experimental Settings. All experimental accuracies are reported via the mean and standard deviation of 10 runs. The experiments are implemented with Python, Numpy, Sklearn, and TensorFlow [36, 37, 38, 39] on a single NVIDIA GTX 1060Ti GPU. We use a neural network of width 100 and depth 2 to generate the probability inputs for the Gumbel-Softmax to obtain G and S; the Gumbel temperature was set to 0.1. ReLU was used as the activation function with softmax at the final layer for prediction. Adam optimizer with a learning rate of 0.001 and hyperparameters $\beta_1 = 0.9, \beta_2 = 0.999$ was used without further tuning. All datasets are centered to 0 and normalized to have a standard deviation of 1. For all data, we used two fully connected layers of width 32 and 16. All λs are identified by maximizing the objective given a validation set.

Competing Methods. We compare gI against nine related feature selection and explainable methods. For all methods, we learn from samples of the true underlying posterior $P(\mathbf{Y}|\mathbf{X})$ (i.e., ground-truth training data) to fairly compare them.

- **Global feature selection**: **Lasso** (Least Absolute Shrinkage and Selection Operator) [23] is a regression method that utilizes l_1 regularization to induce sparsity and effectively perform feature selection. **GLasso** (sparse Group Lasso) [40, 41] is a Lasso version that assumes a feature grouping structure, enforces l_1 sparsity and performs group selection with an $l_{1,2}$ regularizer.
- **Deep instance-wise feature selection**: **SHAP** (SHapley Additive exPlanations) [12] provides a unified framework for explaining models by identifying a class of additive feature importance measures for prediction. SHAP learns feature importance (Shapley values) based on a game theoretic approach. **L2X** [11] performs instance-wise feature selection for explaining black-box models by maximizing the mutual information between the selected features and the response variable. In addition, L2X uses Gumbel softmax to learn a continuous relaxation of the feature selector. **INV** (INVASE) [14] is an extension over L2X without the need to specify the number of selected features in advance and is capable of discovering subsets of features with a different size per instance. **LPA** (Learn to Pay Attention) [15] is A visual-attention based deep learning model for learning saliency maps from the original input images. We adapted the original model to a fully-connected version based on the architecture used by all methods in this paper for fair comparison. Note that these models cannot learn and do not use the feature grouping structure.

- **CAE** [42]: An end-to-end unsupervised global feature selection to reconstruct the input data, with a Gumbel softmax layer as the encoder and a standard neural network as the decoder. As an unsupervised method, we only apply CAE to the visual MNIST and F-MNIST experiments.
- **Global feature selection with group learning: OSCAR** (octagonal shrinkage and clustering algorithm for regression) [18] learns feature groups in regression by regularizing the weights with l_1 and pairwise l_∞ norm to encourage correlated predictors that have a similar effect on the response to form clusters represented by the same coefficient. **OWL-Lasso** [43] performs linear regression and group feature selection by utilizing a weighted l_1 regularization. **Group Cluster** groups the features based on hierarchical correlation clustering [44] followed by GLasso.

Results on Synthetic Data. We use synthetic datasets to answer the following questions:

- Can our model correctly identify the features that are highly dependent on each other?
- Can our model correctly identify the most relevant features in predicting $\mathbf{Y}$?
- Is k based on Theorems 3 and 4 a tight lower bound?
- How does the accuracy of our method compare to existing interpretable methods?

Given all 9 redundancy combinations of (D_i, R_j) plus six additional Gaussian noise features, Table 3 indicates that both gI($m = k$) and gI($m < k$) are capable of achieving high class accuracy while learning the latent group structure (high representation (rep) and relevant (rel) accuracies), thereby confirming Thms. 1 and 2. Moreover, since the recommended k value by Thm. 4 is a lower bound, we investigated the bound by plotting the classification accuracy at each increment of k in Fig. 2 and circle the lower bound predicted by Thm. 4. As predicted by our theorem, after the number of groups passes the lower bound calculated by Thm. 4 the preservation of the mutual information between $\mathbf{X}$ and $\mathbf{Y}$ is possible and indeed after the number of groups passes the lower bound there is no decline in the classification accuracy.

	Class Acc (**gI**($m = k$))			k (**gI**($m = k$))			Group Rep Acc (**gI**($m = k$))			Class Acc (**gI**)			m (**gI**)			Group Rel Acc (**gI**)		
	D_1	D_2	D_3	D_1	D_2	D_3	D_1	D_2	D_3	D_1	D_2	D_3	D_1	D_2	D_3	D_1	D_2	D_3
R_1	96.9 ± 1.5	100 ± 0	99.5 ± 1.5	3	2	2	99.8 ± 0.3	100 ± 0	100 ± 0	91.0 ± 3	99.6 ± 1	98.5 ± 2	1	1	1	100 ± 0	100 ± 0	100 ± 0
R_2	100 ± 0	100 ± 0	99.6 ± 0.4	3	3	3	100 ± 0	100 ± 0	99 ± 1.8	92 ± 3	95.4 ± 2	94 ± 1	2	2	2	100 ± 0	100 ± 0	100 ± 0
R_3	98.9 ± 0.8	97.6 ± 0.6	99.2 ± 0.8	3	3	3	100 ± 0	100 ± 0	99.5 ± 0.9	94.4 ± 3	94.7 ± 2	90.9 ± 5	2	2	2	100 ± 0	100 ± 0	100 ± 0

Table 3: Measuring gI's ability to identify the most relevant groups using nine redundancy patterns. Note that gI is capable of identifying the relevant groups while achieving a high classification accuracy.

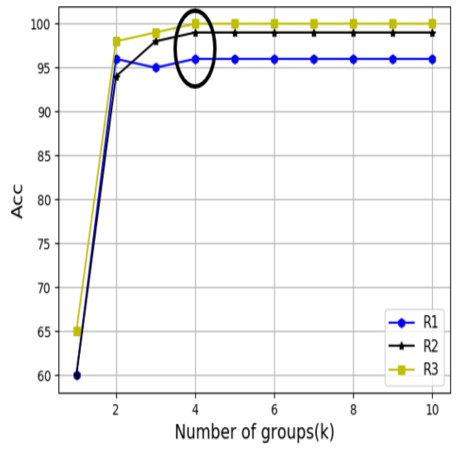

Figure 2: Accuracy versus number of groups used: We circle the number of groups predicted by Thm. 4.

	Data	D_1	D_2	D_3	$D_1 + D_2$
gI	R_1	**98.4± 1**	99.7±0.46	95± 3.8	**98.7 ± 1.3**
	R_2	**100 ± 0**	**99.5 ± 0.5**	**100 ± 0**	**100 ± 0**
	R_3	**98.8 ± 0.9**	**99.4 ± 0.6**	**99.4 ± 0.6**	**99.2 ± 0.4**
L2X	R_1	85 ± 3.5	**100 ± 0.0**	85.7 ± 6	88 ± 4
	R_2	95 ± 2	95 ± 1.4	95 ± 2.1	99.7 ± 0.5
	R_3	94 ± 2.2	95 ± 1.1	87.7 ± 1	93 ± 1.3
Shap	R_1	70.25, ± 0.83	**100 ± 0.0**	**100 ± 0.0**	89.2 ± 0.97
	R_2	88.0 ± 0.0	94 ± 0.0	82 ± 0.0	94.4 ± 0.48
	R_3	94.6 ± 0.48	95 ± 0.0	95 ± 0.0	95.4 ± 0.48
INV	R_1	87.2 ± 3	88.5 ± 3	87.2 ± 3	86 ± 2
	R_2	73 ± 3	80.9 ± 4	68 ± 3.5	75 ± 4
	R_3	74 ± 4	79 ± 2	73 ± 4	74 ± 4
Lasso	R_1	49 ± 3	**100 ± 0.0**	**100 ± 0.0**	74 ± 1
	R_2	66 ± 1	61 ± 1	67 ± 2	58 ± 2
	R_3	75 ± 2	84 ± 2	59 ± 3	81 ± 8
GLasso	R_1	49 ± 1	**100 ± 0.0**	**100 ± 0.0**	76.0 ± 0.4
	R_2	64 ± 0.08	62 ± 0.4	49 ± 0.4	56 ± 0.4
	R_3	74 ± 1	83 ± 0.5	68 ± 1.3	79 ± 2
OSCAR	R_1	49.0 ± 0.31	**100 ± 0.0**	**100 ± 0.0**	50.0 ± 0.0
	R_2	50.0 ± 0.3	50.3 ± 0.11	50.0 ± 0.3	50.1 ± 0.05
	R_3	74.7 ± 0.2	84.03 ± .15	66 ± 0.14	79.2 ± 0.13
OWL	R_1	49.0 ± 0.31	**100 ± 0.0**	**100 ± 0.0**	50.0 ± 0.0
	R_2	50.0 ± 0.3	50.3 ± 0.11	50.0 ± 0.3	50.1 ± 0.05
	R_3	74.7 ± 0.2	84.03 ± .15	66 ± 0.14	79.2 ± 0.13

Table 4: The classification prediction accuracy on synthetic datasets.

In Table 4, we compare gI against competing methods. In addition to mixing D and R redundancies together, we increase the data complexity by combining D_1 relationships with D_2 as $D_1 + D_2$, where half of the samples generated are randomly chosen to have D_1 redundancies while the other half is set to have D_2 redundancies. Since only our model performs classification based on instance-wise grouping of features, the $D_1 + D_2$ pattern is of particular interest to validate our advantage, i.e., given its correct assumption of the data, our model is expected to outperform all alternative methods. By marking the best results as bold in Table 4, we can see that gI is almost always the best performing classifier. As expected, the accuracy difference is particularly prominent with $D_1 + D_2$.

COPDGene Dataset. Given the effect of smoking on health, there is significant interest in its impact on gene expression. Specifically, how does exposure alter gene expression, and how do groups of genes exhibit *coordinated* changes given exposure? This data highlights the insufficiency of learning a single *global* group structure because smokers and non-smokers may be characterized by completely different gene groups. We emphasize the importance of identifying this variability by applying gI to the COPDGene dataset to learn the most predictive group of genes on smoking status. Instead of trying to pinpoint a *single group* of the most important genes, gI's instance-wise capability is designed to automatically identify *multiple groups*.

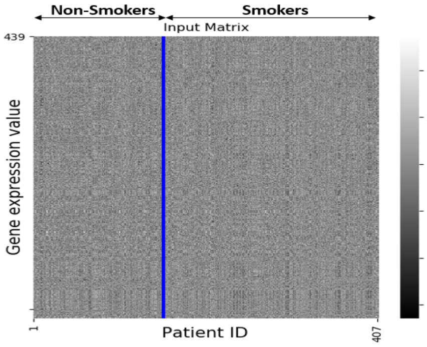

Figure 3: Gene expression (input features) of patients X^T. No pattern is visually noticeable.

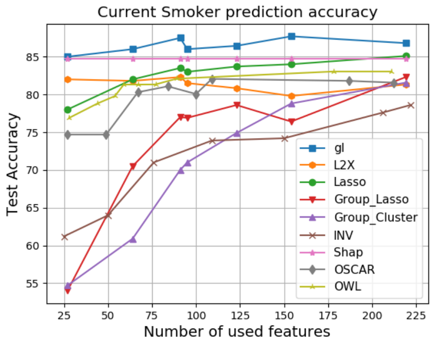

Figure 4: Prediction accuracy vs. number of features selected. gI consistently outperforms other methods.

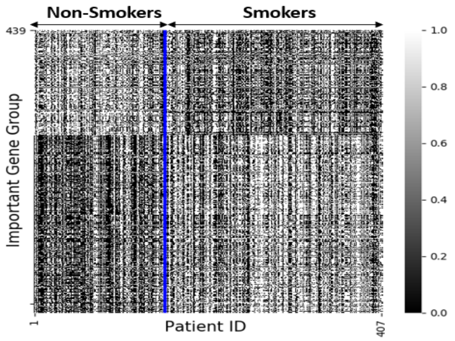

Figure 5: The important genes selected by G and $\mathbf{s}$. The selected genes (rows) are indicated by *white* pixels for each patient (column) and *black* when not selected.

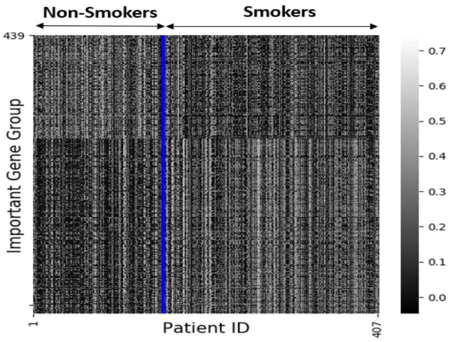

Figure 6: Each column represents the characteristic features of each patient, i.e., $\mathbf{Zs}$. Note that visually, smokers and non-smokers are clustered appropriately.

Fig. 4 compares the test accuracy between several competing methods given increasing number of features; *gI consistently achieves the highest accuracy*. Moreover, note that *Group Lasso* represents the traditional method of applying biologically predefined groups. Yet, even when domain knowledge is incorporated within *Group Lasso*, the instance-wise capability of gI identifies the gene groups that achieves much higher predictive accuracy.

We next studied the group structure produced by gI. First, notice that the original gene expression matrix in Fig. 3 lacked any visually noticeable patterns. We then plot the most relevant group of genes selected by G and s in Fig. 5, where the selected genes (rows) are indicated by the *white* pixels for each patient (column). Even with the high variance between patients, a pattern emerges; gI has identified the group of genes that are common across smokers and non-smokers respectively. As suggested by our results, there exists a visual difference in gene expression between the two groups and gI has identified the specific genes for each group. We next plot out the *characteristic features* formed by each G matrix. As predicted by Thm. 4, this compressed representation of the original input features retained the most relevant information despite the compression.

Since different genes tended to be selected in smokers compared to nonsmokers, we performed Gene Set Enrichment Analysis (GSEA) as implemented in the GenePattern Cloud instance (https://cloud.genepattern.org/) using a set of curated immunologic gene signatures (the C7 set) from the Molecular Signatures Database. Immunologic signatures is well suited for analysis of blood expression data since the majority of cells present in blood are immune cells. The analysis determines whether predefined gene sets are enriched in the extremes of the ranked list of genes, where ranking is based on each gene's likelihood of being selected among each of the two cohorts. In this analysis, using a 10 percent false discovery rate, 20 significantly enriched immunologic gene sets were identified among the most frequently selected genes for smokers, whereas no similar enrichment of immunologic signatures was observed among the genes selected the most among nonsmokers.

Competing Method Performance on Image Datasets. Table 2 reports the classification accuracy for all methods on MNIST-2, MNIST-10 (all 10 digits) and F-MNIST. Notice that while L2X and Shap performed slightly better on the simpler F-MNIST and MNIST-2 (3 vs. 8) datasets, gI performed better on the more complex MNIST-10 (all 10 digits) dataset.

In Fig. 7, we compare the visual patterns generated by gI against several best performing deep models. For each image, each method identifies and displays the most informative pixel group in *white*; the top row is the original image while the results of each method are displayed below. While L2X, LPA, and Shap are all instance-wise and can achieve high predictive accuracy, it is not clear visually from their white pixels in Fig. 7 why these pixels are important. CAE outputs a discernible shape of 8, however, its features are global, resulting in the same pixel choice across all samples. In contrast, gI discovered the pixels that are equally important, resulting in a visually compelling segmentation in the shape of the classification object. Our result suggests that capturing and identifying redundancies within the data produces visually interpretable explanations, highlighting the importance of combining group structure with instance-wise flexibility. While other methods struggle to identify the different digits and clothing, gI handled the complexity independent of the number of classes.

An even more challenging task is to also capture the style variation within the same class. We highlight this ability with 10 digits of diverse shapes in Fig. 7 under INSTANCE-WISE MNIST STYLE; a larger collection showcasing a variety of style variation results can be found in App. F. For these results, notice how the explanatory pixels follow closely to the style of the original image.

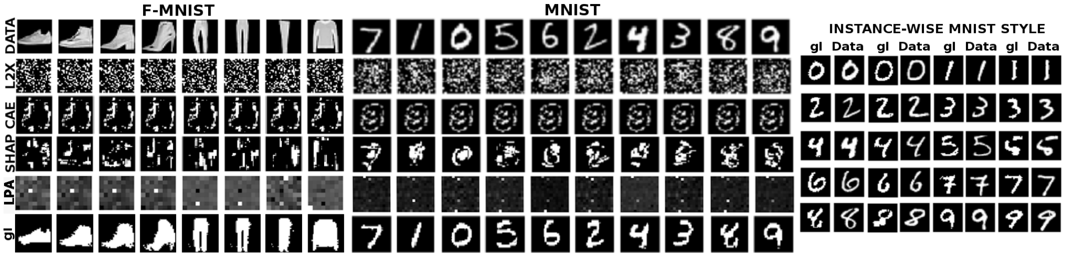

Figure 7: Comparing the most important pixels as identified by each competing algorithm.

4 Conclusion

Our theoretical contribution formally defines the concept of redundancy between features based on MI. This clarifies how features can be grouped together, and how many groups should exist while retaining the most relevant information. It further enables us to formulate an objective (gI) that captures these redundancies on an instance-wise basis. Our theories are corroborated by both synthetic and real experimental results. We have applied our instance-wise feature group discovery and selection method to lung disease gene expression data; of which we discovered gene expression patterns common to smokers and non-smokers respectively.

5 Broader Impacts

In this paper, we introduce a novel algorithm for instance-wise feature group discovery and selection. The algorithm learns mapping functions that identify the appropriate group membership of each feature along with each group's importance as an instance-wise label predictor. Namely, we have focused our paper on feature selection to model the features important for capturing the information in the underlying true posterior $P(\mathbf{Y}|\mathbf{X})$. While we have focused on feature selection, there are also other strategies to define and approach interpretability [45].

Instead of estimating the posterior distribution $P(\mathbf{Y}|\mathbf{X})$, one can apply our method to capture the information for trained black-box models $P_M(\mathbf{Y}|\mathbf{X})$, e.g., deep neural networks and random forests. Consequently, the algorithm can be used to perform instance-wise group feature selection on the black-box model, learning the features which a given black-box model perceives as important. In this approach to explainability, our method has the potential impact on making black-box models explainable in terms of knowing how the features were used during prediction. This gives rise to future research directions that can help data scientists check for bias, fairness, vulnerabilities of the models they use [46, 47].

Although this paper focuses on the machine learning aspect of our discovery, our work is also relevant from its consequential findings on the lung disease dataset. The feature selection results on the lung disease data allow us to discover the genes that interact together for predicting smoking and non-smoking. This can potentially impact our understanding of lung disease, in particular by identifying cooperative relationship between genes that can delineate important aspects of their biological functions. However, to make an impact to medical research would require further and careful investigation to confirm the findings with appropriate medical collaborators. As a warning to our ML and data analyst colleagues, we encourage applying ML to applications that is beneficial to society, such as health. But, to do so properly, one needs to work closely with domain expert collaborators to make nontrivial contributions to their fields of research.

Beyond applications to lung disease, learning important features for prediction and the features that interact together is important in genetic understanding of other diseases [48, 49]. In general, feature selection has been impactful in a variety of domains beyond medicine – for example, climate [50], law[51]. Given the potential impact it can have, including on the most pressing diseases of today, we seek to widely disseminate this research and make our source code publicly available at `https://github.com/ariahimself/Instance-wise-Feature-Grouping`.

Lastly, while our method is useful in identifying feature groups that interact together for prediction. We caution that this does not imply causation, and poses a potential misuse of our technique. Additionally, since our model is learned from a training set, its conclusions are limited by the quality and characteristics of what it was trained on. Therefore, inherent biases that pre-existed in the data will lead to biased feature groups and conclusions. As with any supervised machine learning algorithms, our method can be applied to a variety of applications (e.g., health, climate, image analysis) with potential impact to multiple sectors of society. Our intent is to build such models for societal good and we encourage others to as well.

Acknowledgements

The work described was supported in part by Award Numbers U01 HL089897, U01 HL089856, R01 HL124233, and R01 HL147326 from the National Heart, Lung, and Blood Institute, the FDA Center for Tobacco Products (CTP), and NSF IIS 1546428. The authors would also like to thank Stratis Ioannidis, Amirreza Farnoosh,Yale Chang, Davin Hill, and Zulqarnain Khan for helping to review the paper and provide fruitful discussions. Their insightful comments in conjunction with the comments of the anonymous reviewers have helped to improve this paper tremendously.

References

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

[2] Le Song, Alex Smola, Arthur Gretton, Justin Bedo, and Karsten Borgwardt. Feature selection via dependence maximization. *Journal of Machine Learning Research*, 13(May):1393–1434, 2012.

[3] Jason Weston, Sayan Mukherjee, Olivier Chapelle, Massimiliano Pontil, Tomaso Poggio, and Vladimir Vapnik. Feature selection for svms. In *Advances in neural information processing systems*, pages 668–674, 2001.

[4] Kenji Kira, Larry A Rendell, et al. The feature selection problem: Traditional methods and a new algorithm. In *Aaai*, volume 2, pages 129–134, 1992.

[5] Manoranjan Dash and Huan Liu. Feature selection for classification. *Intelligent data analysis*, 1 (3):131–156, 1997.

[6] Daphne Koller and Mehran Sahami. Toward optimal feature selection. Technical report, Stanford InfoLab, 1996.

[7] Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. In *Advances in neural information processing systems*, pages 507–514, 2006.

[8] Mahdokht Masaeli, Glenn Fung, and Jennifer G. Dy. From transformation-based dimensionality reduction to feature selection. In *Proceedings of the International Conference on Machine Learning*, pages 751–758, 2010.

[9] Mahdokht Masaeli, Yan Yan, Glenn Fung, and Jennifer G. Dy. Convex principal feature selection. In *Proceedings of the SIAM International Conference on Data Mining*, pages 619–628, 2010.

[10] Yale Chang, Yi Li, Adam Ding, and Jennifer Dy. A robust-equitable copula dependence measure for feature selection. In *Artificial Intelligence and Statistics*, pages 84–92, 2016.

[11] Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 883–892, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL `http://proceedings.mlr.press/v80/chen18j.html`.

[12] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774, 2017.

[13] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. *ICML*, abs/1704.02685, 2017.

[14] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. Invase: Instance-wise variable selection using neural networks. *ICLR*, 2018.

[15] Saumya Jetley, Nicholas A Lord, Namhoon Lee, and Philip HS Torr. Learn to pay attention. In *International Conference on Learning Representations*, 2018.

[16] 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.

[17] 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.

[18] Howard Bondell and Brian Reich. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR. *Biometrics*, 64:115–123, March 2008.

[19] Smita Chormunge and Sudarson Jena. Correlation based feature selection with clustering for high dimensional data. *Journal of Electrical Systems and Information Technology*, 5(3): 542–549, 2018.

[20] Jordan Frecon, Saverio Salzo, and Massimiliano Pontil. Bilevel learning of the group lasso structure. In *Advances in Neural Information Processing Systems*, pages 8301–8311, 2018.

[21] Xiangrong Zeng and Mário A. T. Figueiredo. Solving OSCAR regularization problems by fast approximate proximal splitting algorithms. *Digital Signal Processing*, 31:124–135, 2014.

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

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

[24] Lei Yu and Huan Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5:1205–1224.

[25] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[26] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[27] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *ICLR*, 2016.

[28] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2013.

[29] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

[30] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[31] TK Ho. Random decision forests (pdf): Proceedings of the 3rd international conference on document analysis and recognition. 1995.

[32] Aabida Saferali, Jeong H Yun, Margaret M Parker, Phuwanat Sakornsakolpat, Robert P Chase, Andrew Lamb, Brian D Hobbs, Marike H Boezen, Xiangpeng Dai, Kim de Jong, et al. Analysis of genetically driven alternative splicing identifies fbxo38 as a novel copd susceptibility gene. *PLoS genetics*, 15(7):e1008229, 2019.

[33] Adrian Alexa and Jorg Rahnenfuhrer. topgo: enrichment analysis for gene ontology. *R package version*, 2(0):2020, 2020.

[34] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

[35] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[36] Guido Van Rossum and Fred L Drake. *The python language reference manual*. Network Theory Ltd., 2011.

[37] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. URL `http://www.scipy.org/`. [Online; accessed <today>].

[38] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[39] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

[40] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A note on the group lasso and a sparse group lasso. *arXiv preprint arXiv:1001.0736*, 2010.

[41] Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of computational and graphical statistics*, 22(2):231–245, 2013.

[42] Abubakar Abid, Muhammad Fatih Balin, and James Zou. Concrete autoencoders for differentiable feature selection and reconstruction. *ICML*, 2019.

[43] Małgorzata Bogdan, Ewout Van Den Berg, Chiara Sabatti, Weijie Su, and Emmanuel J Candès. Slope—adaptive variable selection via convex optimization. *The annals of applied statistics*, 9 (3):1103, 2015.

[44] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

[45] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv:1702.08608v2*, 2017.

[46] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58: 82–115, June 2020.

[47] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Computing Surveys*, 51(5):Article 93, August 2018.

[48] Rebecka Jörnsten and Bin Yu. Simultaneous gene clustering and subset selection for sample classification via MDL. *Bioinformatics*, 19(9):1100–1109, 2003.

[49] Mee Young Park, Trevor Hastie, and Robert Tibshirani. Averaged gene expressions for regression. *Biostatistics*, 8(2):212–227, 2007.

[50] Soumyadeep Chatterjee, Karsten Steinhaeuser, Arindam Banerjee, Snigdhansu Chatterjee, and Auroop Ganguly. Sparse group lasso: Consistency and climate applications. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 47–58. SIAM, 2012.

[51] Harry Surden. Machine learning and law. *Wash. L. Rev.*, 89:87, 2014.

A Mutual information

In this section, we provide a short tutorial on Mutual Information (MI, I) and some of its known properties we use within our proofs. Given two random variables X and Y, MI measures the distributional distance between $P(X,Y)$ and $P(X)P(Y)$ via the Kullback-Leibler divergence. The equation for $I(X,Y)$ is written as

$$I(X;Y) = D_{\mathrm{KL}}(P_{(X,Y)} \| P_X \otimes P_Y) \tag{14}$$

where Kullback-Leibler divergence of two distributon P, Q is defined as

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right). \tag{15}$$

Here, $\mathcal{X}$ is the defined domain of the random variable X.

Property 1: MI is non negative, or $I(X;Y) \geq 0$.

Property 2: MI is symmetric, or $I(X;Y) = I(Y;X)$.

Property 3: MI with joint distributions has the following chain rule:

$$I(Y;X_1,\ldots,X_n) = I(Y;X_1) + I(Y;X_2|X_1) + \cdots + I(Y;X_n|X_1,\ldots,X_{n-1}). \tag{16}$$

Property 4: MI of X and Y can be increase or decrease via the conditioning of a third variable Z, i.e., it is possible that $I(X;Y|Z) \geq I(X;Y)$ or $I(X;Y|Z) \leq I(X;Y)$ when Z is introduced. As a corollary to this property, it is therefore possible that $I(X;Y) = 0 \not\to I(X;Y|Z) = 0$.

Property 5: MI between variables is conserved through diffeomorphism. We formally state this property as Lemma 3 below.

> **Lemma 3.** *Mutual information is invariant under reparametrization of the marginal variables if $X' = F(X)$ and $Y' = G(Y)$ are diffeomorphism, then:*
>
> $$I(X;Y) = I(F(X);G(Y))$$

Property 6: MI cannot be increased via any deterministic transformation of its variables. We formally state this property as Lemma 4 below.

> **Lemma 4.** *According to inequality of data processing for any deterministic transformation f we have:*
>
> $$I(f(X);Y) \leq I(X;Y).$$
>
> Therefore, as a key corollary used in our proof, we note that
> **Corollary 4.3.** *Using the encoder T_G as f will not enhance the mutual information. meaning:*
>
> $$I(T_G(X);Y) \leq I(X;Y).$$

A.1 A note to mention:

We defined $\mathcal{X} = \{X_1,\ldots,X_d\}$, and $\mathcal{A} \subseteq \mathcal{X}$, where X_j is a random variable representing feature j and $\mathcal{A}$ is a random variable that has a subset of features. But we can also define these notations as follows: $\mathcal{X} = \{(\lambda^1,\ldots,\lambda^d) | \forall j \lambda^j = X_j \vee \lambda^j = 0\}$, and $\mathcal{A} \in \mathcal{X}$. Note that $|\mathcal{X}| = 2^d$, because λ^j has two choices, to either reflect the random variable X_j or be 0. Furthermore, the every element in $\mathcal{X}$ has can be seen as a projection of the element that all λ are non zero or $(x_1,\ldots,x_d)$. Using the data processing Lemma 4. Hence we can say $I(\mathcal{X};\mathbf{Y}) = I(X;\mathbf{Y})$.

B Proof for Lemmas 1 and 2

Lemma 1. *The decoder $T_G^+ : \mathbf{Z} \to \mathrm{Im}(T_G^+)$ is a Diffeomorphism map.*

Proof. Diffeomorphism map is a map that is smooth and bijective.

Smoothness: The decoder T_G^+ in a linear map, hence smooth. We need to prove that T_G^+ is bijective.

Injectivity: First we prove it is injective, and therefore

$$\forall z_i, z_j \in \mathbb{R}^k,\, T_G^+(z_i) = T_G^+(z_j) \to z_i = z_j, \tag{17}$$

The Group matrix $G \in \mathbb{R}^{k\times d}$ can be represented as its rows which we represent as $\hat{g}_i \in \mathbb{R}^d$, $\forall i = 1, \dots, k$. because the group are not overlapping, the rows of G are always orthogonal with each other which means:

$$\forall i \neq j,\, 1 \leq i < j \leq k \quad \langle \hat{g}_i, \hat{g}_j \rangle = 0. \tag{18}$$

Let the characteristic features of the i^{th} sample be represented as $z_i \in \mathbb{R}^k = [z_{i,1}, \dots, z_{i,k}]^T$, the decoder is defined as $T_G^+(z_i) = G^T \cdot z_i$ which is equal to

$$G^T \cdot z_i = z_{i,1}\hat{g}_1 + \cdots + z_{i,k}\hat{g}_k. \tag{19}$$

Assume $T_G^+(z_i) = T_G^+(z_j)$, we will prove that $z_i = z_j$. First, because T_G^+ is linear, we can combine z_i, z_j, hence $T_G^+(z_i) = T_G^+(z_j)$ lead to $T_G^+(z_i - z_j) = 0$, using Eq. 19 we obtain

$$T_G^+(z_i - z_j) = (z_{i,1} - z_{j,1})\hat{g}_1 + \cdots + (z_{i,k} - z_{j,k})\hat{g}_k = 0. \tag{20}$$

Note that $(z_{i,\eta} - z_{j,\eta})$ is a scalar value for all η, therefore, we can multiply $\hat{g}_\eta^T$ on both side of the equality condition to obtain

$$(z_{i,1} - z_{j,1})\hat{g}_1 + \cdots + (z_{i,k} - z_{j,k})\hat{g}_k = 0 \tag{21}$$

$$(z_{i,1} - z_{j,1})\hat{g}_\eta^T\hat{g}_1 + \cdots + (z_{i,k} - z_{j,k})\hat{g}_\eta^T\hat{g}_k = \hat{g}_\eta^T(0) = 0 \tag{22}$$

Using Eq. (18), since the inner product between $\hat{g}_\eta$ with another $\hat{g}_{\neq\eta}$ is always 0, only the $\langle \hat{g}_\eta, \hat{g}_\eta \rangle$ remains in the equation as

$$(z_{i,\eta} - z_{j,\eta})\langle \hat{g}_\eta, \hat{g}_\eta \rangle = 0. \tag{23}$$

From Eq. (23), we the following two possibilities:

- $\forall\ \eta$, $(z_{i,\eta} - z_{j,\eta}) = 0$, which implies $z_i = z_j$
- Or $\langle \hat{g}_\eta, \hat{g}_\eta \rangle = 0$

The 2nd condition of $\langle \hat{g}_\eta, \hat{g}_\eta \rangle = 0$ implies that no features belong to group η. It is important to realize here that z_i came from the original features x_i where $z_i = Gx_i$. Therefore, if no features belong to group η, $z_{i,\eta}$ and $z_{j,\eta}$ must both be 0, or $z_{i,\eta} = z_{j,\eta}$ for all η. Hence , we conclude that $z_i = z_j$ and consequently T_G^+ is injective.

surjective: The decoder is surjective becuase it maps Z to $\mathrm{Im}(T_G^+)$, i.e., the entire $\mathrm{Im}(T_G^+)$ is being mapped. Since the decoder is simultaneoulsy injective and surjective, we conclude that it is also bijective. Using Lemma 3, we can conclude that mutual information is preserved where

$$I(\mathbf{Z}; \mathbf{Y}) = I(T_G^+(\mathbf{Z}); \mathbf{Y}). \tag{24}$$

□

Lemma 2. *If $k < d$ then T_G is not injective.*

Proof. Given the T_G is a linear function, we know that if T_G is injective $\iff$ dim(ker$(T_G)) = 0$, where $\ker(T_G) = \{\mathbf{v} \in R^d \,|\, T_G(\mathbf{v}) = 0\}$. Moreover, for a linear transformation T, we have the following property:

$$\text{dim}(\ker(T_G)) + \text{dim}(\mathrm{Im}(T_G) = d$$

Because dim$(\mathrm{Im}(T_G)) \leq k \to$ dim$(\ker(T_G)) \geq d - k > 0$ Thus, T_G is not injective.

□

C Proof for Theorems 1 and 2

Note on notation: We denote $H(\mathbf{X})$ as the entropy of $\mathbf{X}$.

Theorem 1. *The maximum mutual information $I(\hat{\mathbf{X}};\mathbf{X})$ is achieved if and only if its characteristic features $\mathbf{Z}$ induced by the model makes $\mathbf{X}$ representative redundant based on Def. (1), i.e.*

$$\begin{gathered}\max_G I(\hat{\mathbf{X}};\mathbf{X}) = I(\mathbf{X};\mathbf{X}) \iff \min_G I(\mathbf{X};\mathbf{X}|\mathbf{Z}) = 0, \\ \text{s.t. } G \in \{0,1\}^{k\times d}, \sum_{i=1}^{k} G_{ij} = 1, \, \mathbf{Z} = T_G(\mathbf{X}), \, \hat{\mathbf{X}} = \psi_{\theta_G}(\mathbf{X}).\end{gathered} \tag{25}$$

Proof. We first prove the condition

$$I(\hat{\mathbf{X}};\mathbf{X}) = I(\mathbf{X};\mathbf{X}) \implies I(\mathbf{X};\mathbf{X}|\mathbf{Z}) = 0. \tag{26}$$

Based on Lemma 4 in App.A, we know that $I(\hat{\mathbf{X}};\mathbf{X})$ is upper bounded by $I(\mathbf{X};\mathbf{X})$, i.e., the mutual information can never exceed the entropy of $\mathbf{X}$. Therefore, the optimal G^* that maximizes $I(\hat{\mathbf{X}};\mathbf{X})$ is found if the following condition is satisfied.

$$I(\hat{\mathbf{X}};\mathbf{X}) = I(\mathbf{X};\mathbf{X}). \tag{27}$$

This can be proven given the following three observations:

- Using Lemmas 1 and 3, MI is preserved under T_G^{+} and therefore the information between the following are the same.
$$I(\mathbf{Z};\mathbf{X}) = I(\hat{\mathbf{X}};\mathbf{X}). \tag{28}$$
Using $I(\hat{\mathbf{X}};\mathbf{X}) = I(\mathbf{X};\mathbf{X})$. and Eq. (28) implies that
$$I(\mathbf{Z};\mathbf{X}) = I(\mathbf{X};\mathbf{X}). \tag{29}$$
- We note that $\mathbf{Z} = T_G(\mathbf{X})$ and that T_G is a deterministic function. Therefore, $P(\mathbf{Z}|\mathbf{X}) = \delta(\mathbf{Z})$ where δ denotes the Kronecker's delta. Given this observation, we have
$$P(\mathbf{X},\mathbf{Z}) = P(\mathbf{X},\mathbf{Z}) \tag{30}$$
$$= P(\mathbf{Z}|\mathbf{X})P(\mathbf{X}) \tag{31}$$
$$= [\delta(T_G(\mathbf{X}))]P(\mathbf{X}) \tag{32}$$
$$P(\mathbf{X},\mathbf{Z}) = P(\mathbf{X}). \tag{33}$$
Therefore, it leads to the conclusion that
$$I(\mathbf{X},\mathbf{Z};\mathbf{X}) = I(\mathbf{X};\mathbf{X}). \tag{34}$$
- We write the chain rule for $I(\mathbf{X},\mathbf{Z};\mathbf{X})$ as
$$I(\mathbf{X},\mathbf{Z};\mathbf{X}) = I(\mathbf{Z};\mathbf{X}) + I(\mathbf{X};\mathbf{X}|\mathbf{Z}). \tag{35}$$
By using Eq. (34) and (29), the equality becomes
$$I(\mathbf{X};\mathbf{X}) = I(\mathbf{X};\mathbf{X}) + I(\mathbf{X};\mathbf{X}|\mathbf{Z}). \tag{36}$$
Therefore, $I(\mathbf{X};\mathbf{X}|\mathbf{Z})$ must equal to 0 and condition Eq. (26) is proven.

We next prove the reverse condition

$$I(\mathbf{X};\mathbf{X}|\mathbf{Z}) = 0 \implies I(\hat{\mathbf{X}};\mathbf{X}) = I(\mathbf{X};\mathbf{X}). \tag{37}$$

Start by leveraging the result from Eq. (34) to obtain

$$\begin{aligned} I(\mathbf{X};\mathbf{X}) &= I(\mathbf{X},\mathbf{Z};\mathbf{X}) && (38)\\ &= I(\mathbf{Z};\mathbf{X}) + I(\mathbf{X};\mathbf{X}|\mathbf{Z}) && \text{using the chain rule} \quad (39)\\ &= I(\mathbf{Z};\mathbf{X}) + 0 && \text{using the condition assumption} \quad (40)\\ &= I(\hat{\mathbf{X}};\mathbf{X}) && \text{using the Eq. (28)} \quad (41)\\ &= H(\mathbf{X}) && \text{using the definition of Entropy.} \quad (42)\end{aligned}$$

□

Theorem 2. *The maximum mutual information $I(\bar{\mathbf{X}}; \mathbf{X})$ is achieved if and only if its m-selected characteristic features $\mathbf{Z} \odot \mathbf{s}$ induced by the model makes $\mathbf{X}$ relevant redundant based on Def. (2), i.e.*

$$\begin{gathered}\max_G I(\bar{\mathbf{X}}; \mathbf{Y}) = I(\mathbf{X}; \mathbf{Y}) \iff \min_G I(\mathbf{X}; \mathbf{Y} | \mathbf{Z} \odot \mathbf{s}) = 0, \\ \text{s.t. } G \in \{0,1\}^{k \times d}, \sum_{i=1}^{k} G_{ij} = 1, \mathbf{Z} = T_G(\mathbf{X}),\ \bar{\mathbf{X}} = \phi_{\theta_S, \theta_G}(\mathbf{X}), \mathbf{s} \in \{0,1\}^k, |\mathbf{s}| = m.\end{gathered} \tag{43}$$

Proof. **Forward direction.**

The proof follows the similar direction as Theorem 1: First we show:

$$I(\bar{\mathbf{X}}; \mathbf{Y}) = I(\mathbf{X}; \mathbf{Y}) \implies I(\mathbf{X}; \mathbf{Y} | \mathbf{Z} \odot \mathbf{s}) = 0.$$

using the data processing lemma 4, $I(\bar{\mathbf{X}}; \mathbf{Y}) \leq I(\mathbf{X}; \mathbf{Y})$, hence the optimal solution $G^*, \mathbf{s}^*$ satisfies $I(\bar{\mathbf{X}}; \mathbf{Y}) = I(\mathbf{X}; \mathbf{Y})$. The goal is to show that given

$$I(\bar{\mathbf{X}}; \mathbf{Y}) = I(\mathbf{X}; \mathbf{Y}) \tag{44}$$

then

$$I(\mathbf{X}; \mathbf{Y} | \mathbf{Z} \odot \mathbf{s}) = 0. \tag{45}$$

As stated in Lemma 1 the mutual information is preserved under the decoder map, hence

$$I(\mathbf{Z} \odot \mathbf{s}; \mathbf{Y}) = I(T_G^{+}(\mathbf{Z} \odot \mathbf{s}); \mathbf{Y}) = I(\bar{\mathbf{X}}; \mathbf{Y}) = I(\mathbf{X}; \mathbf{Y}). \tag{46}$$

From this observation, it leads to the following derivation:

$$\begin{aligned} I(\mathbf{X}, \mathbf{Z} \odot \mathbf{s}; \mathbf{Y}) &= I(\mathbf{Z} \odot \mathbf{s}; \mathbf{Y}) + I(\mathbf{X}; \mathbf{Y} | \mathbf{Z} \odot \mathbf{s}) \quad \text{via chain rule} && (47) \\ I(\mathbf{X}; \mathbf{Y}) &= I(\mathbf{Z} \odot \mathbf{s}; \mathbf{Y}) + I(\mathbf{X}; \mathbf{Y} | \mathbf{Z} \odot \mathbf{s}) \quad \text{via Eq. (33)}, P(\mathbf{X}, \mathbf{Z}) = P(\mathbf{X}) && (48) \\ I(\mathbf{X}; \mathbf{Y}) &= I(\mathbf{X}; \mathbf{Y}) + I(\mathbf{X}; \mathbf{Y} | \mathbf{Z} \odot \mathbf{s}) \quad \text{via Eq. (46)}, I(\mathbf{Z} \odot \mathbf{s}; \mathbf{Y}) = I(\mathbf{X}; \mathbf{Y}). && (49) \end{aligned}$$

Since $I(\mathbf{X}; \mathbf{Y}) = I(\mathbf{X}; \mathbf{Y})$, then the condition $I(\mathbf{X}; \mathbf{Y} | \mathbf{Z} \odot \mathbf{s}) = 0$ must be true.

Reverse direction. We now prove

$$I(\mathbf{X}; \mathbf{Y} | \mathbf{Z} \odot \mathbf{s}) = 0 \implies I(\bar{\mathbf{X}}; \mathbf{Y}) = I(\mathbf{X}; \mathbf{Y}). \tag{50}$$

First note that

$$\begin{aligned} I(\mathbf{X}; \mathbf{Y}) &\leq I(\mathbf{X}; \mathbf{Y}) + I(\mathbf{Z} \odot \mathbf{s}; \mathbf{Y} | X) && (51) \\ &\leq I(\mathbf{X}, \mathbf{Z} \odot \mathbf{s}; \mathbf{Y}) \qquad \text{Chain Rule.} && (52) \end{aligned}$$

Second, note that $\phi_{\theta_S, \theta_G}$ is a deterministic function, and therefore a function f defined as $f(\mathbf{X}) = [\mathbf{X}, \phi_{\theta_S, \theta_G}(\mathbf{X})]^T$ is also a deterministic function. This give us the following relationship

$$\begin{aligned} I(\mathbf{X}, \mathbf{Y}) &\geq I(f(\mathbf{X}), \mathbf{Y}) \qquad \text{Apply Lemma. 4.} && (53) \\ &\geq I(\mathbf{X}, \phi_{\theta_S, \theta_G}(\mathbf{X}); \mathbf{Y}) \qquad \text{Apply function } f. && (54) \\ &\geq I(\mathbf{X}, \mathbf{Z} \odot \mathbf{s}; \mathbf{Y}) \qquad \text{Apply definition of } \phi_{\theta_S, \theta_G}(X). && (55) \end{aligned}$$

Combining Eq. (52) and Eq. (55) together, we have

$$I(\mathbf{X}, \mathbf{Z} \odot \mathbf{s}; \mathbf{Y}) \geq I(\mathbf{X}, \mathbf{Y}) \geq I(\mathbf{X}, \mathbf{Z} \odot \mathbf{s}; \mathbf{Y}), \tag{56}$$

which is only possible if

$$I(\mathbf{X}, \mathbf{Z} \odot \mathbf{s}; \mathbf{Y}) = I(\mathbf{X}, \mathbf{Y}). \tag{57}$$

Leveraging this result, we see that

$$\begin{aligned} I(\mathbf{X}; \mathbf{Y}) &= I(\mathbf{X}, \mathbf{Z} \odot \mathbf{s}; \mathbf{Y}) && (58) \\ &= I(\mathbf{Z} \odot \mathbf{s}; \mathbf{Y}) + I(\mathbf{X}; \mathbf{Y} | \mathbf{Z} \odot \mathbf{s}) \qquad \text{using the chain rule} && (59) \\ &= I(\mathbf{Z} \odot \mathbf{s}; \mathbf{Y}) + 0 \qquad \text{using the condition assumption} && (60) \\ &= I(\bar{\mathbf{X}}; \mathbf{Y}) \qquad \text{using the Eq. (29)} && (61) \end{aligned}$$

□

D Proof for Theorem 3

Theorem 3. *Let $\mathcal{X} = \{X_1, \dots, X_d\}$ be a random variable that is consists of all features , let relevant features $\mathcal{U}$ be*

$$\mathcal{U} = \{X_j |\ I(X_j; \mathbf{Y}) \neq 0 \vee \exists \mathcal{A} \subseteq \mathcal{X}\, I(X_j; \mathbf{Y}|\mathcal{A}) \neq 0\}, \tag{62}$$

and let irrelevant features be $\mathcal{U}^c$, then, $\exists G \in \mathcal{G}$ such that $I(T_G(\mathbf{X}); \mathbf{Y}) = I(\mathbf{X}; \mathbf{Y})$ if

$$k \geq |\mathcal{U}| + \mathbb{1}(|\mathcal{U}| \neq d) \tag{63}$$

where $\mathbb{1}(|\mathcal{U}| \neq d)$ is an indicator function equal to one when $|\mathcal{U}| \neq d$ and zero when $|\mathcal{U}| = d$.

Proof. Set $\mathcal{U}$ is the collection of features with the property $I(X_j; \mathbf{Y}) \neq 0$ or $I(X_j; \mathbf{Y}|\mathcal{A}) \neq 0$. In other words, the first inequality implies that a features belongs to $\mathcal{U}$ if its mutual information with respect to $\mathbf{Y}$ is not 0. Yet, just because the MI between a feature and a label is 0, sometimes, their MI is no longer 0 given another set of features. Therefore, we add the 2nd condition to includes these cases. Namely, a feature belongs to $\mathcal{U}$ if it directly provide information on $\mathbf{Y}$ or if it indirectly provide information given $\mathcal{A}$.

Note that the definition of $\mathcal{U}$ is a consequences of Def. (2). Conversely, it allows us to also define its complement $\mathcal{U}^c$ as featuers that doesn't provide any information on $\mathbf{Y}$ even when it is conditioned on a set of features $\mathcal{A}$. Formally, we define $\mathcal{U}^c$ as

$$\mathcal{U}^c = \{X_j \,|\, I(X_j; \mathbf{Y}) = 0 \text{ and }, \forall \mathcal{A}\ I(X_j; \mathbf{Y}|\mathcal{A}) = 0\}. \tag{64}$$

To prove the theorem, we first we prove the following lemma:

Lemma 5. *Set $\mathcal{U}^c$ is relevant Redundant with respect to set $\mathcal{U}$, meaning:*

$$I(\mathcal{U}^c; \mathbf{Y}|\mathcal{U}) = 0$$

Proof. Without loss of generality assume $|\mathcal{U}^c| = n$ which we present these set of features by $\hat{x}^1, \dots \hat{x}^n$:

Based on chain rule, we have the following equality for $I(\hat{x}^1, \dots \hat{x}^n, \mathcal{U}; \mathbf{Y})$:

$$I(\hat{x}^1, \dots \hat{x}^n, \mathcal{U}; \mathbf{Y}) = I(\mathcal{U}; \mathbf{Y}) + I(\hat{x}^1; \mathbf{Y}|\mathcal{U}) + I(\hat{x}^2; \mathbf{Y}|\mathcal{U}, \hat{x}^1) + \cdots + I(\hat{x}^n; \mathbf{Y}|\mathcal{U}, \hat{x}^1, \dots, \hat{x}^{n-1}) \tag{65}$$

Each $\hat{x}^i \in \mathcal{U}^c$, hence $I(\hat{x}^i; Y|\mathcal{A}) = 0$, which leads to the following:

$$I(\mathcal{U}^c, \mathcal{U}; \mathbf{Y}) = I(\hat{x}^1, \dots \hat{x}^n, \mathcal{U}; \mathbf{Y}) = I(\mathcal{U}; \mathbf{Y}) + 0 + \cdots + 0. \tag{66}$$

Eq. 66 also leads to the conclusion of this lemma:

$$I(\mathcal{U}^c, \mathcal{U}; \mathbf{Y}) = I(\mathcal{U}; \mathbf{Y}) + I(\mathcal{U}^c; \mathbf{Y}|\mathcal{U}) = I(\mathcal{U}; \mathbf{Y}) \implies I(\mathcal{U}^c; \mathbf{Y}|\mathcal{U}) = 0$$

Which means $\mathcal{U}^c$ is *relevant redundant* with respect to set $\mathcal{U}$.

□

Lemma 6. *With the definition of $\mathcal{U}$, we have the following equality:*

$$I(\mathcal{U}; \mathbf{Y}) = I(\mathcal{U}, \mathcal{U}^c; \mathbf{Y})$$

Proof. This lemma is one of the consequences of Lemma 5, using chain rules would lead us to the following results:

$$I(\mathbf{Y}; \mathcal{U}, \mathcal{U}^c) = I(\mathbf{Y}; \mathcal{U}) + I(\mathbf{Y}; \mathcal{U}^c|\mathcal{U}) = I(\mathbf{Y}; \mathcal{U}) + 0 \tag{67}$$

□

Now we prove the following Lemma, which is the final step for proofing the theorem.

Lemma 7. *The encoder, $T_G : \mathbf{X} \to \mathbf{Z}$, is a mapping induced by a $G \in \{0,1\}^{k \times d}$, $\sum_{i=1}^{k} G_{ij} = 1,$. If T_G has the property of*

$$T_G|_U := U \to \mathrm{Im}(T_G|_U) \quad \textit{is bijective} \quad \textit{and} \quad \mathrm{Im}(T_G|_U) \cap \mathrm{Im}(T_G|_{U^c}) = \{0\} \tag{68}$$

where $dim(U) = |\mathcal{U}|$, is subspace created by features in $\mathcal{U}$, and all the elements $\mathcal{U}^c$ maps to a separated axis' that is orthogonal to elements that are mapped from U, then

$$I(\mathbf{X}; \mathbf{Y}) = I(T_G(\mathbf{X}); \mathbf{Y}). \tag{69}$$

Proof. First we prove such a G exists for $k = |\mathcal{U}| + \mathbb{1}(|\mathcal{U}| \neq d)$ and it can captures the mutual information between $\mathbf{X}$ and $\mathbf{Y}$. Note the encoder $T_G : \mathbf{X} \to \mathbf{Z}$ is from $\mathbb{R}^d \to \mathbb{R}^k$. let $\hat{e}^1, \ldots, \hat{e}^k$ the basis axis in R^k, and $\hat{x}^1, \ldots, \hat{x}^k$ the values of each axis. Assume $|\mathcal{U}| = m$, T_G is bijective with respect to set U. Thus, without loss of generality, assume $e^1, \ldots, e^m$ axis in U are mapped to $\hat{e}^1, \ldots, \hat{e}^m$ in Z. And axis in U^c are mapped to $\hat{e}^{m+1}, \ldots, \hat{e}^k$. Hence T_G acts as an identity for U, because U^c sends to orthogonal subspace we can say the following: $I(T_G(\mathbf{X}); \mathbf{Y}) = I(U, \mathrm{Im}(T_G|_{U^c}); \mathbf{Y})$. based on lemma 6 we have the following:

$$I(T_G(\mathbf{X}); \mathbf{Y}) = I(T_G(U), \mathrm{Im}(T_G|_{U^c}); \mathbf{Y}) \geq I(T_G(U); \mathbf{Y}) = I(U; \mathbf{Y}) \tag{70}$$

which $I(U; \mathbf{Y})$ we know it is equal to $I(\mathbf{X}; \mathbf{Y})$ based on lemma 6, Hence $I(T_G(\mathbf{X}); \mathbf{Y}) \geq (\mathbf{X}; \mathbf{Y})$. But where $I(T_G(\mathbf{X}); \mathbf{Y}) \leq (\mathbf{X}; \mathbf{Y})$ is based on data processing Lemma 4. Hence $I(\mathbf{X}; \mathbf{Y}) = I(T_G(\mathbf{X}); \mathbf{Y}) = I(U; \mathbf{Y})$, which concludes the proof. □

Based on 7, as long as we can satisfy Eq. 68, we can guarantee the preservation of mutual information between $\mathbf{X}$ and $\mathbf{Y}$. As long as $k \geq |\mathcal{U}| + \mathbb{1}(|\mathcal{U}| \neq d)$, we can define T_G to act as identity on U and map U^c to orthogonal subspace of $\mathrm{Im}(T_G|_U)$ which doesn't need to be injective, i.e. it can map everything to one axis. If $|U| = d$ then G is the identify map. Not that G is a group matrix and has to map each input to an output, that is the reason we need at least one axis for elements in U^c.

□

E Proof for Theorem 4

Theorem 4.

Given ρ as the correlation coefficient and

$$\mathcal{C} = \{X_j | \rho(X_j; \mathbf{Y}) \neq 0 \vee \exists \mathcal{A}\, \rho(X_j; \mathbf{Y}|\mathcal{A}) \neq 0\} \tag{71}$$

then: $|\mathcal{C}| \leq |\mathcal{U}|$.

Proof. Let $\mathcal{C}_1 = \{X_j | \rho(X_j; \mathbf{Y}) \neq 0\}$ and $\mathcal{C}_2 = \{X_j | \exists \mathcal{A}, \rho(X_j; \mathbf{Y}|\mathcal{A}) \neq 0\}$, then

$$\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2. \tag{72}$$

Also we know that $\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2$ if we let $\mathcal{U}_1 = \{X_j | I(X_j; \mathbf{Y}) \neq 0\}$ and $\mathcal{U}_2 = \{X_j | \exists \mathcal{A}, I(X_j; \mathbf{Y}|\mathcal{A}) \neq 0\}$. If we can show that $\mathcal{C}_1 \subseteq \mathcal{U}_1$ and $\mathcal{C}_2 \subseteq \mathcal{U}_2$, then $|\mathcal{C}| \leq |\mathcal{U}|$ is proven.

We first note that since MI of $I(Z_1; Z_2|\mathcal{A})$ measures the KL divergence between $P(Z_1, Z_2|\mathcal{A})$ and $P(Z_1|\mathcal{A})P(Z_2|\mathcal{A})$, if $I(Z_1; Z_2|\mathcal{A}) = 0$, then the following condition must also be true:

$$P(Z_1, Z_2|\mathcal{A}) = P(Z_1|\mathcal{A})P(Z_2|\mathcal{A}). \tag{73}$$

Using the condition from Eq. (73), we can compute the conditional expectation where

$$\begin{aligned}
E_{Z_1,Z_2}[Z_1 Z_2] &= \int_{z_1} \int_{z_2} z_1 z_2 p(z_1, z_2|\mathcal{A}) dz_1 dz_2 && (74)\\
&= \int_{z_1} \int_{z_2} z_1 z_2 p(z_1|\mathcal{A}) p(z_2|\mathcal{A}) dz_1 dz_2 && (75)\\
&= \left[\int_{z_1} z_1 p(z_1|\mathcal{A}) dz_1\right] \left[\int_{z_2} z_2 p(z_2|\mathcal{A}) dz_2\right] && (76)\\
&= E_{Z_1}[Z_1] E_{Z_2}[Z_2]. && (77)
\end{aligned}$$

Since $E_{Z_1,Z_2}[Z_1Z_2] = E_{Z_1}[Z_1]E_{Z_2}[Z_2]$, it implies that the cross-covariance must also be 0 where

$$E_{Z_1,Z_2}[Z_1Z_2] - E_{Z_1}[Z_1]E_{Z_2}[Z_2] = 0. \tag{78}$$

Since ρ is the cross-covariance scaled by a constant, we see that if MI is 0 then ρ is also 0. By contrapositivity, we see that if ρ is not equal to zero then MI is not equal to 0. Therefore, if an element is in $\mathcal{C}_1$ or $\mathcal{C}_2$, it must also be included into $\mathcal{U}_1$ or $\mathcal{U}_2$. Hence, we have shown that $\mathcal{C}_1 \subseteq \mathcal{U}_1$ and $\mathcal{C}_2 \subseteq \mathcal{U}_2$

□

Corollary 4.1. *Theorems 3 and 4 yields a lower bound for k where $|\mathcal{C}| + \mathbb{1}(|\mathcal{C}| \neq d) \leq k$.*

Proof. We want to proof that $|\mathcal{C}| + \mathbb{1}(|\mathcal{C}| \neq d) \leq |\mathcal{U}| + \mathbb{1}(|\mathcal{U}| \neq d)$, and then using Theorem 3, $k \geq |\mathcal{U}| + \mathbb{1}(|\mathcal{U}| \neq d)$, we conclude the proof. Using Theorem 4, we have $\mathcal{C} \subseteq \mathcal{U}$. We have the following cases:

- **Case 1.** $|\mathcal{C}| = d$: Since $\mathcal{C} \subseteq \mathcal{U}$, thus $|\mathcal{U}| = d$. and therefore: $|\mathcal{C}| + \mathbb{1}(|\mathcal{C}| \neq d) = |\mathcal{U}| + \mathbb{1}(|\mathcal{U}| \neq d) \leq k$.
- **Case 2.** $|\mathcal{C}| < d$: This means $\mathbb{1}(|\mathcal{C}| \neq d) = 1$ We have two sub cases:
  - $\mathbb{1}(|\mathcal{U}| \neq d) = 1$: using the fact that $|\mathcal{C}| \leq |\mathcal{U}|$, we conclude: $|\mathcal{C}| + \mathbb{1}(|\mathcal{C}| \neq d) \leq |\mathcal{U}| + \mathbb{1}(|\mathcal{U}| \neq d)$, since both indicator function are equal to one.
  - $\mathbb{1}(|\mathcal{U}| \neq d) = 0$: This means that $|\mathcal{U}| = d$ or the whole space, since $|\mathcal{C}| < d$, therefore, there exist atleast one element in $\mathcal{U}$ that is not in $\mathcal{C}$, thus: $|\mathcal{C}| < |\mathcal{U}|$, which means: $\mathbb{1}(|\mathcal{C}| \neq d) \leq |\mathcal{U}| + \mathbb{1}(|\mathcal{U}| \neq d)$.

□

Corollary 4.2. For Gaussian distributions the inequality turns into equality where $|\mathcal{C}| = |\mathcal{U}|$.

Proof. If $\mathbf{X}$ and $\mathbf{Y}$ are independent, then they are also uncorrelated. However, if $\mathbf{X}$ and $\mathbf{Y}$ are uncorrelated, then they could be dependent. General case when lack of correlation implies independence is when the joint distribution of $\mathbf{X}$ and $\mathbf{Y}$ is Gaussian, which means $\mathcal{C} = \mathcal{U}$. Note that in classification case, $P(X, Y)$ cannot be Gaussian due to the discreteness of the classification problem.

□

F Complete Collection of *Style Adaptation* Results

In figure 8, we see the complete collection of *Style Adaptation* Results. Our method, gI, is capable of capturing the most relevant pixels in spite of having multiple classes with variations among each class.

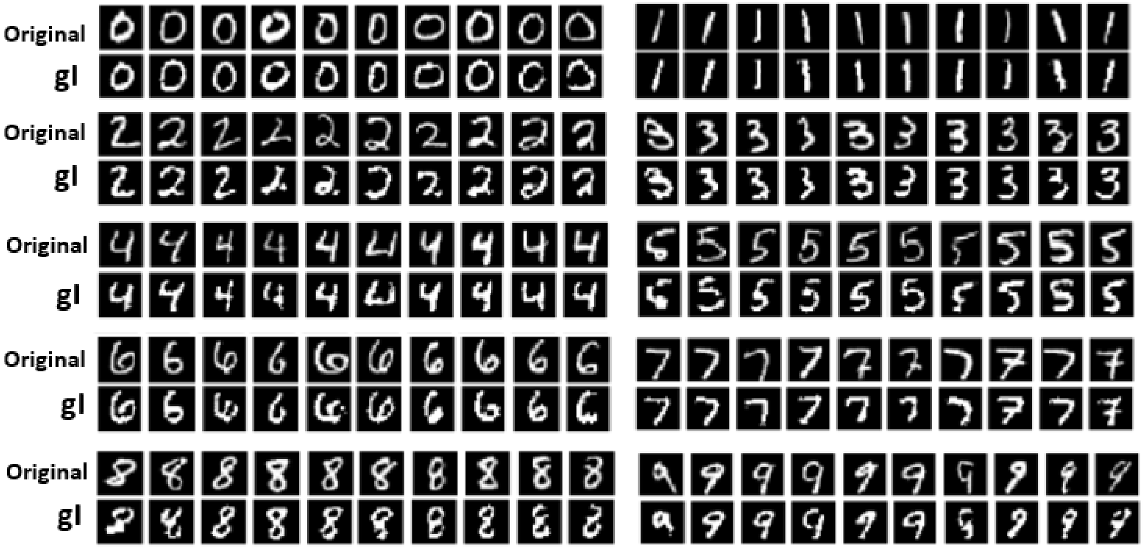

Figure 8: gI is capable to handling the added complexity of style variation among multiple classes.

MNIST result for 3 groups

In Fig 9, we investigate gI model on $k = 3$ and $m = 1$ number of groups we showed each group with a different color red, blue, green. As we can see increasing the number of groups kept the shape of each digits but it separates one of the groups into two which is the digit patterns

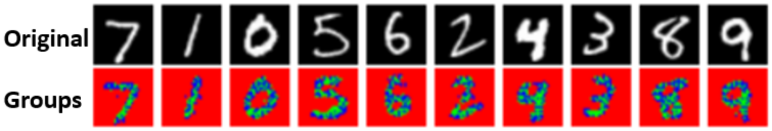

Figure 9: MNIST result for $k = 3$ groups and $m = 1$ number of most important groups.

MNIST alphabet

Im Fig 10, we tried our model for classification of 25 alphabet and the result of grouping is as follows for some of the instances.

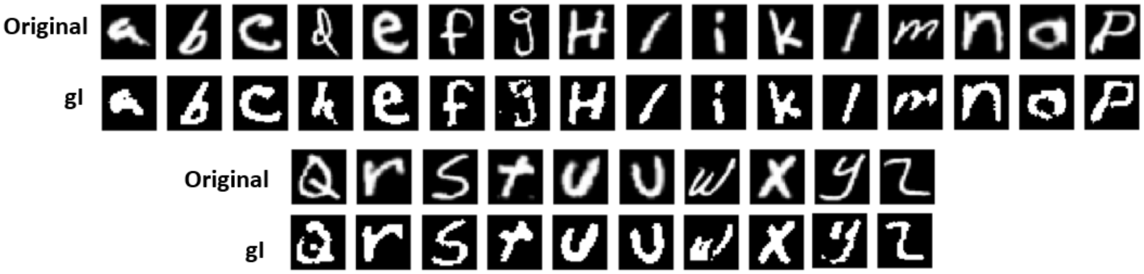

Figure 10: MNIST Alphabet

COPD result for 3 Groups.

In Fig 11 we tried COPD gene data set with $k = 3$ number of groups and we are showing the important group with white pixels similar to the paper. The result are indicator of similar network as we have for 2 groups in the paper but increasing number of groups seems to make the important group sparser or less number of features in the most important group.

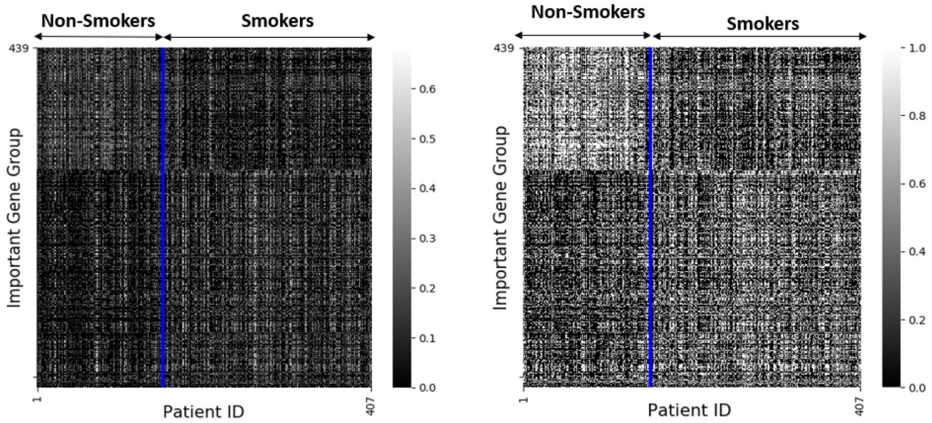

Figure 11: COPD gene expression and gene most important groups for $k = 3, m = 1$.

Implementation details for experiments.

Since an overly expressive network of Q_{θ_R} for reconstruction confounds the interpretability of the group structure. It is preferable to define Q_{θ_R} as a simpler function. We have found two functions that work well experimentally. First, for *balanced* dataset, Q_{θ_R} can simply be an identify function. However, for *unbalanced* datasets, we found a well behaved function to be one that converts each characteristic feature to the average value of features in that group. This can be seen as the following function: $f_{\theta_R} : \mathbb{R}^d \to \mathbb{R}^b$ where given I_i to be the indexes of feature belong to group i. And let $j \in I_j$ It send $\hat{x}_j$ to $f_{\theta_R}(\hat{x}_j) = \frac{\hat{x}_j}{|I_i|}$. which means the average value of each group is a good reconstruction of each feature in that group.

G Variational lower bound

From Eq. (7), Eq. (8) we can derive the graphical model which is shown on Fig. 12.

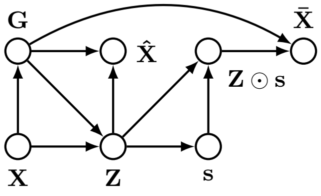

Figure 12: Graphical model for group learning representation

Given our network ϕ_{θ_S,θ_G} that is parameterized by θ_G, θ_S. We wish to solve the problem

$$\underset{\theta_G,\theta_S}{\arg\max}\ \underbrace{I(\psi_{\theta_G}(\mathbf{X}),\mathbf{X})}_{\Xi_1}+\underbrace{\lambda I(\phi_{\theta_S,\theta_G}(\mathbf{X});\mathbf{Y})}_{\Xi_2}. \tag{79}$$

MI, however, is difficult to compute due to its requirement of both the joint and marginal distributions. Chen et al. [11] circumvented this problem by calculating the variational lower bound. Since the lower bound is tractable, it can be maximized as a surrogate to Eq. (**??**). We start the derivation by only looking at Ξ_2 following the definition of MI as

$$\arg\max_{\theta_G,\theta_S} \mathbf{MI}(\phi_{\theta_S,\theta_G}(\mathbf{X});\mathbf{Y}) = \arg\max_{\theta_G,\theta_S} \int_{x\in\mathcal{X}}\int_{y\in\mathcal{Y}} p(\phi_{\theta_S,\theta_G}(x),y)\log\left(\frac{p(\phi_{\theta_S,\theta_G}(x),y)}{p(\phi_{\theta_S,\theta_G}(x))p(y)}\right)dxdy. \tag{80}$$

Since $p(\phi_{\theta_S,\theta_G}(x),y) = p(y|\phi_{\theta_S,\theta_G}(x))p(\phi_{\theta_S,\theta_G}(x))$, we replace the numerator term inside the log and cancel out $p(\phi_{\theta_S,\theta_G}(x))$, the objective then becomes

$$\arg\max_{\theta_G,\theta_S} \mathbf{MI}(\phi_{\theta_S,\theta_G}(\mathbf{X});\mathbf{Y}) = \arg\max_{\theta_G,\theta_S} \int_{x\in\mathcal{X}}\int_{y\in\mathcal{Y}} p(\phi_{\theta_S,\theta_G}(x),y)\log\left(\frac{p(y|\phi_{\theta_S,\theta_G}(x))}{p(y)}\right)dxdy. \tag{81}$$

The integrals can be rewritten into

$$\begin{aligned}
&= \arg\max_{\theta_G,\theta_S} \int_{x\in\mathcal{X}}\int_{y\in\mathcal{Y}} p(\phi_{\theta_S,\theta_G}(x),y)\log\left[p(y|\phi_{\theta_S,\theta_G}(x))\right] - p(\phi_{\theta_S,\theta_G}(x),y)\log\left[p(y)\right]dxdy,\\
&= \arg\max_{\theta_G,\theta_S} \int_{x\in\mathcal{X}}\int_{y\in\mathcal{Y}} p(\phi_{\theta_S,\theta_G}(x),y)\log\left[p(y|\phi_{\theta_S,\theta_G}(x))\right]dxdy - \int_{x\in\mathcal{X}}\int_{y\in\mathcal{Y}} p(\phi_{\theta_S,\theta_G}(x),y)\log\left[p(y)\right]dxdy,\\
&= \arg\max_{\theta_G,\theta_S} \int_{x\in\mathcal{X}}\int_{y\in\mathcal{Y}} p(\phi_{\theta_S,\theta_G}(x),y)\log\left[p(y|\phi_{\theta_S,\theta_G}(x))\right]dxdy - \int_{y\in\mathcal{Y}}\left[\int_{x\in\mathcal{X}} p(\phi_{\theta_S,\theta_G}(x),y)dx\right]\log\left[p(y)\right]dy,\\
&= \arg\max_{\theta_G,\theta_S} \int_{x\in\mathcal{X}}\int_{y\in\mathcal{Y}} p(\phi_{\theta_S,\theta_G}(x),y)\log\left[p(y|\phi_{\theta_S,\theta_G}(x))\right]dxdy - \int_{y\in\mathcal{Y}} p(y)\log\left[p(y)\right]dy.
\end{aligned}$$

Since $\int_{y\in\mathcal{Y}} p(y)\log\left[p(y)\right]dy$ no longer has a θ_G,θ_S term, the maximization over θ_G,θ_S can be treated as a constant, i.e., this term can be removed from the optimization object which leads us to

$$\arg\max_{\theta_G,\theta_S} \int_{x\in\mathcal{X}}\int_{y\in\mathcal{Y}} p(\phi_{\theta_S,\theta_G}(x),y)\log\left[p(y|\phi_{\theta_S,\theta_G}(x))\right]dxdy, \tag{82}$$

$$\arg\max_{\theta_G,\theta_S} \int_{x\in\mathcal{X}}\int_{y\in\mathcal{Y}} p(y|\phi_{\theta_S,\theta_G}(x))p(\phi_{\theta_S,\theta_G}(x))\log\left[p(y|\phi_{\theta_S,\theta_G}(x))\right]dxdy, \tag{83}$$

$$\arg\max_{\theta_G,\theta_S} \int_{x\in\mathcal{X}} p(\phi_{\theta_S,\theta_G}(x))\left[\int_{y\in\mathcal{Y}} p(y|\phi_{\theta_S,\theta_G}(x))\log\left[p(y|\phi_{\theta_S,\theta_G}(x))\right]dy\right]dx. \tag{84}$$

$$\arg\max_{\theta_G,\theta_S} E_{\phi_{\theta_S,\theta_G}(\mathbf{X})}\left[\int_{y\in\mathcal{Y}} p(y|\phi_{\theta_S,\theta_G}(x))\log\left[p(y|\phi_{\theta_S,\theta_G}(x))\right]dy\right], \tag{85}$$

$$\begin{aligned}\arg\max_{\theta_G,\theta_S} E_{\phi_{\theta_S,\theta_G}(\mathbf{X})}E_{\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X})}\left[\log(p(\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X})))\right] &=\\ \arg\max_{\theta_G,\theta_S} E_{\mathbf{Y},\phi_{\theta_S,\theta_G}(\mathbf{X})}\left[\log(p(\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X})))\right].\end{aligned} \tag{86}$$

If we look closer at the inner expectation, $E_{\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X})}\left[log(p(\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X})))\right]$, we do not assume to have $p(\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X}))$. Instead, we wish to approximate the distribution via another distribution $Q_{\theta_P}(\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X}))$ that is parameterized by θ_P. The approximation can be done by making sure that the KL divergence between p and q is minimized. When writing out the KL divergence, we get

$$\begin{aligned}
KL(p||q) &= \int_{y\in\mathcal{Y}} p(y|\phi_{\theta_S,\theta_G}(x))\log\left[\frac{p(y|\phi_{\theta_S,\theta_G}(x))}{Q_{\theta_P}(y|\phi_{\theta_S,\theta_G}(x))}\right]dy,\\
&= \int_{y\in\mathcal{Y}} p(y|\phi_{\theta_S,\theta_G}(x))\log(p(y|\phi_{\theta_S,\theta_G}(x))) - p(y|\phi_{\theta_S,\theta_G}(x))\log(Q_{\theta_P}(y|\phi_{\theta_S,\theta_G}(x)))dy,\\
&= E_{\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X})}[\log p(y|\phi_{\theta_S,\theta_G}(x)))] - E_{\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X})}[\log Q_{\theta_P}(y|\phi_{\theta_S,\theta_G}(x)))].
\end{aligned}$$

Since KL divergence is always 0 or greater, we get the inequality relation

$$E_{\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X})}[\log p(y|\phi_{\theta_S,\theta_G}(x)))] - E_{\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X})}[\log Q_{\theta_P}(y|\phi_{\theta_S,\theta_G}(x)))] \geq 0 \tag{87}$$

$$E_{\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X})}[\log p(y|\phi_{\theta_S,\theta_G}(x)))] \geq E_{\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X})}[\log Q_{\theta_P}(y|\phi_{\theta_S,\theta_G}(x)))]. \tag{88}$$

The inequality suggests that $E_{\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X})}[\log(Q_{\theta_P}(\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X})))]$ is a lower bound of $E_{\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X})}[\log(p(\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X})))]$, and they are equal only when $p = q$. Therefore, by finding the θ that maximizes $E_{\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X})}[\log(Q_{\theta_P}(\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X})))]$ is equivalent to finding the best approximation of $E_{\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X})}[\log(p(\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X})))]$. Next, we take Inequality (88) and rewrite each term back in terms of its integration, we obtain

$$\begin{aligned}&\int_{y\in\mathcal{Y}} p(y|\phi_{\theta_S,\theta_G}(x))\text{log}p(y|\phi_{\theta_S,\theta_G}(x)))dy \geq \\ &\qquad\qquad \int_{y\in\mathcal{Y}} p(y|\phi_{\theta_S,\theta_G}(x))\text{log}Q_{\theta_P}(y|\phi_{\theta_S,\theta_G}(x)))dy.\end{aligned} \tag{89}$$

The key realization of this inequality is that given any $\phi_{\theta_S,\theta_G}(\mathbf{X})$, the inequality will still hold. Therefore, if we additionally integrate both terms over any set of $\mathcal{X}$, the inequality will still hold. Following this logic, we can add an additional integration and maintain the inequality.

$$\begin{aligned}&\int_{x\in\mathcal{X}} p(\phi_{\theta_S,\theta_G}(x)) \int_{y\in\mathcal{Y}} p(y|\phi_{\theta_S,\theta_G}(x))\text{log}p(y|\phi_{\theta_S,\theta_G}(x)))dydx \geq \\ &\qquad\qquad \int_{x\in\mathcal{X}} p(\phi_{\theta_S,\theta_G}(x)) \int_{y\in\mathcal{Y}} p(y|\phi_{\theta_S,\theta_G}(x))\text{log}Q_{\theta_P}(y|\phi_{\theta_S,\theta_G}(x)))dydx\end{aligned} \tag{90}$$

$$\begin{aligned}&\int_{x\in\mathcal{X}}\int_{y\in\mathcal{Y}} p(y,\phi_{\theta_S,\theta_G}(x))\text{log}p(y|\phi_{\theta_S,\theta_G}(x)))dydx \geq \\ &\qquad \int_{x\in\mathcal{X}}\int_{y\in\mathcal{Y}} p(y,\phi_{\theta_S,\theta_G}(x))\text{log}Q_{\theta_P}(y|\phi_{\theta_S,\theta_G}(x)))dydx.\end{aligned} \tag{91}$$

$$E_{\mathbf{Y},\phi_{\theta_S,\theta_G}(\mathbf{X})}\left[\log(p(\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X})))\right] \geq E_{\mathbf{Y},\phi_{\theta_S,\theta_G}(\mathbf{X})}\left[\log(Q_{\theta_P}(\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X})))\right] \tag{92}$$

By looking at the relationship between Eq. (86) and (92), notice that if we simultaneously maximize θ and θ_G, θ_S using Q_{θ_P}, the θ_P term would help us find the closest approximation of p while the θ_G, θ_S term would help us maximize the MI objective. Therefore, to maximize Eq. (**??**), we can use Q_{θ_P} as a surrogate and instead maximize

$$\max_{\theta_P,\theta_G,\theta_S} E_{\mathbf{Y},\phi_{\theta_S,\theta_G}(\mathbf{X})}\left[\log(Q_{\theta_P}(\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X})))\right] \tag{93}$$

We can estimate Eq. 93 by ancestral sampling from $\mathbf{X}$ and $\mathbf{Y}$ based on the graphical model in Fig. 12 to compute the expectation empirically in the new objective below as

$$\max_{\theta_P,\theta_G,\theta_S} \frac{1}{n}\sum_{i=1} \log(Q_{\theta_P}(y_i|\phi_{\theta_S,\theta_G}(x_i))). \tag{94}$$

Here, we are performing maximum likelihood. Note that since $\mathbf{Y}$ is the label, the probability of y_i equaling its label is 1, and the probability of it being another label is 0. Therefore, the Eq. (94) can be equivalently written as minimizing the Cross-Entropy loss where

$$\min_{\theta_P,\theta_G,\theta_S} -\sum_{i=1} p(y_i)\log(Q_{\theta_P}(y_i|\phi_{\theta_S,\theta_G}(x_i))). \tag{95}$$

Therefore, given $\psi_{\theta_G,\theta_S}(\mathbf{X}) = \bar{\mathbf{X}}$, objective (95) can be used in place of the Ξ_2 term of our objective Eq. (79).

Following the same derivation, we can replace the Ξ_1 with its lower bound as well. Here we use Q_{θ_R} as the neural network to approximate the true distribution.

$$\max_{\theta_R,\theta_G} E_{\mathbf{X},\psi_{\theta_G}(\mathbf{X})}\left[\log(Q_{\theta_R}(\mathbf{X}|\psi_{\theta_G}(\mathbf{X})))\right]. \tag{96}$$

Consequently, instead of maximizing Eq. (79) directly, we can maximized its variational lower bound

$$\max_{\theta_R,\theta_G,\theta_S} E_{\mathbf{X},\psi_{\theta_G}(\mathbf{X})}\left[\log(Q_{\theta_R}(\mathbf{X}|\psi_{\theta_G}(\mathbf{X})))\right] + \lambda E_{\mathbf{Y},\phi_{\theta_S,\theta_G}(\mathbf{X})}\left[\log(Q_{\theta_P}(\mathbf{Y}|\phi_{\theta_S,\theta_G}(\mathbf{X})))\right], \tag{97}$$

or

$$\max_{\theta_P,\theta_R,\theta_G,\theta_S} E_{\mathbf{X},\hat{\mathbf{X}}}\left[\log(Q_{\theta_R}(\mathbf{X}|\hat{\mathbf{X}}))\right] + \lambda E_{\mathbf{Y},\bar{\mathbf{X}}}\left[\log(Q_{\theta_P}(\mathbf{Y}|\bar{\mathbf{X}}))\right]. \tag{98}$$

Different Variations of this Objective. It is not always necessary to approximate $p(\mathbf{X}|\hat{\mathbf{X}})$ with Q_{θ_R}. Depending on prior information on the data, we can simply assume $p(\mathbf{X}|\hat{\mathbf{X}})$ to have a certain distribution. For example, we can simply set it to the Gaussian distribution if $\mathbf{X}$ can take any value. Here, if we let Q_{θ_P} be a Gaussian distribution of some constant σ, then given $\psi_{\theta_G}(x_i)$ as the mean Eq. (96) becomes

$$\max_{\theta_G,\theta_S} \frac{1}{n} \sum_{i=1} \log \left(e^{-\frac{||x_i - \psi_{\theta_G}(x_i)||^2}{2\sigma^2}} \right). \tag{99}$$

By building σ directly into λ, we can ignore σ. By applying the log term to the exponential term, the objective becomes

$$\min_{\theta_G,\theta_S} \sum_{i=1} ||x_i - \psi_{\theta_G}(x_i)||^2. \tag{100}$$

From this objective, we see that by assuming that Q_{θ_R} is a Gaussian Distribution, we can instead optimize MSE as a variational lower bound for the mutual information, i.e., we no longer need to pass $\psi_{\theta_G}(x_i)$ through Q_{θ_R}. Therefore, as a surrogate, we can solve Eq. (3) with

$$\min_{\theta_P,\theta_R,\theta_G,\theta_S} \sum_{i=1} ||x_i - \psi_{\theta_G}(x_i)||^2 - \lambda \sum_{i=1} p(x_i)\log(Q_{\theta_P}(x_i|\phi_{\theta_S,\theta_G}(x_i))). \tag{101}$$

H Computational and Memory Complexity Analysis

We derive the complexity for a general k, d, but in most cases, we assume the number of group are much smaller than number of features meaning: $k << d$. For our purposes, we are using stochastic gradient descent. So the complexity is proportion to the number of samples N into the number of parameters involved in the neural net. For each sample, the input size is d, then we have an neural net which the output is the Group matrix G, hence the number of parameters for this part is kd^2. Since G determines the auto-encoder $\psi_{\theta_G}(x_i) = G^T G x_i$ for a sample x_i. thus that is just matrix multiplication. So for auto-encoder $\psi_{\theta_G}(x_i)$ the complexity is $O(nkd^2)$. For group selection part, we have another neural net for learning the projection map through selector $\mathbf{s}$, which the complexity is k^2 which lead to $\mathbf{Z} \odot \mathbf{s}$. Having $\mathbf{Z} \odot \mathbf{s}$, lead to $\bar{\mathbf{X}}$ by matrix multiplication. We used the last neural net from $\bar{\mathbf{X}}$ to predict the class lables. Assumming we have C classes, lead to the complexity of Cd. Hence the overall complexity for **gI** is $O(n(kd^2 + k^2 + Cd))$, assuming $C << d$ and $k << d$, complexity of **gI** is $O(nkd^2)$. For the memory complexity of a stochastic gradient descent, we only need to save the information of weights for each sample at a time, hence it is $O(kd^2)$. If we are doing mini-batch this number linearly increases by the size of the mini-batches.