Interpreting Black-Box Models via Instance-Wise Feature Grouping

Anonymous Authors[1]

Abstract

Many machine learning methods are black-box models, and our ability to interpret them is still an ongoing challenge. In domains where feature interactions are complex and can differ per sample, we learn which features interact together as a group and which groups are important for the classifier. The groups are established via optimizing a variational lower bound of Mutual Information to capture redundancies between the features, which we formally define as representation redundancy and relevant redundancy. Theoretically, we leverage these redundancies to design a formulation for instance-wise feature group discovery and reveal a guideline to help discover the appropriate number of groups. Our experimental results[1] further corroborate with our theoretical work and provide visually compelling evidence of interpretability on MNIST and Fashion MNIST.

1. Introduction

Many powerful machine learning algorithms are black-box models. Due to the growing usage of machine learning in medicine, marketing, and criminal justice [1], it is becoming increasingly important to provide explanations to justify classification decisions. Being told by an algorithm that a person has cancer or is guilty of a crime is no longer sufficient: the algorithm must tell us why. This paper focuses on reverse-engineering the decisions of a classifier and provides an instance-wise explanation of how each feature affects the decision-making process.

A common strategy to justify a classification label is to identify the importance of each feature via feature weighting. Techniques such as Linear Discriminant Analysis and Interpretable Kernel Dimension Reduction [2; 3], are a few examples of this approach. Instead of weighing each feature, a different approach is to focus on *Feature Selection* techniques that directly identify the most predictive features [4; 5; 6; 7; 8; 9; 10; 11; 12].

While knowing the most important features are useful, the same features may not be equally important across the entire population. This complexity is particularly evident with images; while one set of pixels may help us identify a person, a vastly different set of pixels would be required to identify a cat. 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 by Chen et al. [13], Lundberg and Lee [14], and Yoon et al. [15].

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. 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.

Due to the advantage of learning the feature groups, there exist methods like Group Lasso [16] and Bilevel Learning [17] that selects which feature groups are important given a predefined grouping. Yet, in many applications the feature groups are unknown; e.g., Chormunge and Jena [18] first learns the feature groups based on k-means clustering. While these methods perform group feature selection, the groups are global and not instance-wise. Moreover, the classifiers they work with are shallow, thereby limiting their classification accuracy. In contrast to these approaches, this paper introduces *instance-wise* methods that can learn the feature group structure and identify its importance on *black-box models*. We refer to this approach as *Instance-wise Feature Grouping*.

Our Contribution.

- We introduce two methods capable of identifying the unique group structure for every sample: *Group classifier* (gC) and *Group interpreter* (gI). The *Group classifier* automatically learns which features interact together as a group for predicting the output class. The *Group Interpreter* not only automatically learns which features are grouped together but also which group is

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. **AUTHORERR: Missing \icmlcorrespondingauthor.**

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

[1]All experiments are easily reproducible and publicly available on https://github.com/anonymous.

important for a black-box classifier.

- We formally define two types of information redundancies via Mutual Information: *Representation Redundancy* captures the dependency between features while *Relevant Redundancy* captures the dependency between features and its corresponding labels.
- We then draw on the formal definitions of the redundancies to design a formulation for *Instance-wise Feature Grouping*. Leveraging information theory, we theoretically prove how our formulation discovers groups that satisfy the two redundancies, and reveal a guideline to help discover the appropriate number of groups necessary to preserve information.
- Our experimental results on synthetic data corroborate with our theoretical results, and our empirical results on two image data sets (MNIST and Fashion MNIST) show visually compelling evidence of interpretability. All experiments are easily reproducible and publicly available on https://github.com/anonymous.

2. Ingredients of Group Learning

Given $\mathbf{X} \in \mathbb{R}^{n \times d}$ as a dataset 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 summarize these groups by compressing them into k characteristic features (to be defined in the next paragraph). We indicate each feature's group membership with a random matrix $G \in \{0,1\}^{k \times d}$ where $G_{i,j} = 1$ if the j^{th} feature of a sample belongs to the i^{th} group. Consequently, each column becomes an indicator vector denoting its corresponding feature's group membership.

To learn G, we propose to train a non-traditional *autoencoder* ψ that encodes 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 $\psi(\mathbf{X}) = \hat{\mathbf{X}}$. 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 $z_i = T_G(x_i) = Gx_i$, $\hat{x}_i = T_G^+(z_i) = G^T z_i$, and $\hat{x}_i = \psi(x_i) = T_G^+ \circ T_G(x_i) = G^T G x_i$. Based on these definitions, ψ deviates from traditional autoencoders in that G completely describes the entire autoencoder. In addition, since each feature of z_i is a linear function of only features of the same group, each feature encapsulates the characteristics of its corresponding group; accordingly, we refer to them as *characteristic features*.

We learn G, and consequently ψ based on two types of information redundancy. Namely, (1) Features can be redundant if the occurrence of a feature outcome is highly dependent on another set of features; we refer to this redundancy as *Representation Redundancy*. (2) Features can also be redundant if they provide similar label predicting information, i.e., given the occurrence of a feature, additional features may not provide any extra label-predicting information; we refer to this type of redundancy as *Relevant Redundancy*. Formally, given $\mathcal{X}_j$ as a random variable representing the j^{th} feature, and let $\mathcal{X} = \{X_1, \ldots, X_d\}$ be a set of all features with its cardinality denoted as $|\mathcal{X}|$, then the two redundancies are defined below utilizing mutual information (MI, I) as a measure of dependence.

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).

The Group Matrix G. In our problem, each sample corresponds to a unique and individualized G matrix. However, instead of finding a separate G matrices for every sample, we directly learn $P(\mathbf{G}|\mathcal{X})$ to obtain a distribution of $\mathbf{G}$ given any input from $\mathcal{X}$. Note that while $\mathbf{G}$ is a random variable, G is the most likely outcome of $\mathbf{G}$, i.e., once $P(\mathbf{G}|\mathcal{X})$ is learned, we set $G = \arg\max_{\mathbf{G}} P(\mathbf{G}|\mathcal{X})$.

Group Learning Methods. We provide two methods that can identify the feature groups, *Group Classifier* and *Group Interpreter*. The *Group classifier* (gC) automatically learns which features interact together as a group for predicting the output class on a per sample basis. The *Group Interpreter* (gI) not only automatically learns which features are grouped together but also which group is important for each sample instance. Each method comes with its own trade-offs in terms of its interpretability, complexity, and classification accuracy. gC is less complex to learn than gI and as such often leads to higher classification accuracies than gI. However, gI offers higher interpretability than gC as it also provides the importance of each feature group.

The *Group Classifier* optimizes for the following objective:

$$\max_{\psi} \quad I(\hat{\mathbf{X}}; \mathbf{X}) + \lambda I(\hat{\mathbf{X}}; \mathbf{Y}) \quad \text{s.t:} \quad \hat{\mathbf{X}} = \psi(\mathbf{X}). \tag{3}$$

The ψ function that maximizes $I(\hat{\mathbf{X}}; \mathbf{X})$ generates groups that capture *Representation Redundancy*. Simultaneously, by maximizing $I(\hat{\mathbf{X}}; \mathbf{Y})$, it also captures *Relevant Redundancy*. The control parameter λ balances the importance of the two redundancy criteria. We show that the choice of maximizing Eq. (3) satisfies both Defs. 1 and 2 in the following two theorems; the proof is included in App. C.

Theorem 1. *The characteristic features induced by the optimal maximization of $I(\hat{\mathbf{X}};\mathbf{X})$ satisfies Def. (1) such that*

$$\arg\max_G I(\hat{\mathbf{X}};\mathbf{X}) = I(\mathbf{X};\mathbf{X}) \rightarrow I(\mathbf{X};\mathbf{X}|\mathbf{Z}) = 0. \quad (4)$$

Conversely, the characteristic features that satisfies Def. (1) maximizes $I(\hat{\mathbf{X}};\mathbf{X})$ such that

$$\arg\min_G I(\mathbf{X};\mathbf{X}|\mathbf{Z}) = 0 \rightarrow I(\hat{\mathbf{X}};\mathbf{X}) = I(\mathbf{X};\mathbf{X}). \quad (5)$$

Theorem 2. *The characteristic features induced by the optimal maximization of $I(\hat{\mathbf{X}};\mathbf{Y})$ satisfies Def. (2) such that*

$$\arg\max_G I(\hat{\mathbf{X}};\mathbf{Y}) = I(\mathbf{X};\mathbf{Y}) \rightarrow I(\mathbf{X};\mathbf{Y}|\mathbf{Z}) = 0. \quad (6)$$

Conversely, the characteristic features that satisfies Def. (2) maximizes $I(\hat{\mathbf{X}};\mathbf{Y})$ such that

$$\arg\min_G I(\mathbf{X};\mathbf{Y}|\mathbf{Z}) = 0 \rightarrow I(\hat{\mathbf{X}};\mathbf{Y}) = I(\mathbf{X};\mathbf{Y}). \quad (7)$$

As we demonstrate later in the experimental section, gC can learn the feature groups and achieve high accuracies. However, it does not inform us on which feature group is important. Hence, we introduce our second method, the *Group Interpreter*. In emphasizing interpretability, for each sample, we wish to identify the m most relevant groups that are used to label the data. In so doing, we identify the most *Relevant Redundant* groups while also identifying the *Representation Redundant* groups.

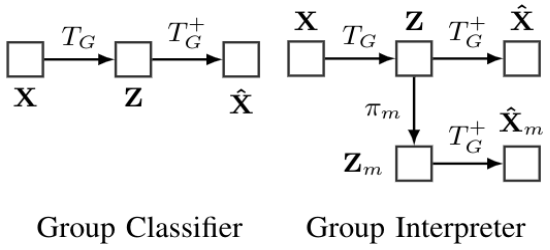

Figure 1. Functional flow of the *Group Classifier* and *Interpreter*.

To discover the m most important groups, we introduce an m-hot masking vector $S \in \mathbb{R}^k$, where m elements are ones while the rest are zeros. Similar to our approach in finding G, we do not find S directly for each sample, instead, we learn a distribution $P(\mathbf{S}|\mathbf{Z})$ where $\mathbf{S}$ is a random variable, and we set $S = \arg\max_{\mathbf{S}} P(\mathbf{S}|\mathbf{Z})$, i.e., S is the most likely outcome of $P(\mathbf{S}|\mathbf{Z})$. Once the masking vector S is obtained, a Hadamard product between the features of $\mathbf{Z}$ and S will mask out the least important groups. We define the function $\pi_m(\mathbf{Z})$ as the Hadamard product between $\mathbf{Z}$ and S where $\pi_m(\mathbf{Z}) = \mathbf{Z} \odot S$ and denote the output of $\pi_m(\mathbf{Z})$ as $\mathbf{Z}_m$. Given $\mathbf{Z}_m$ as the new bottleneck, the output of the autoencoder is denoted as $\hat{\mathbf{X}}_m$. Consequently, we denote ϕ as the autoencoder where $\phi(\mathbf{X}) = \hat{\mathbf{X}}_m$.

The *Group Interpreter* optimizes for the following objective:

$$\begin{aligned} &\max_{\psi,\phi} I(\hat{\mathbf{X}};\mathbf{X}) + \lambda I(\hat{\mathbf{X}}_m;\mathbf{Y}), \\ &\text{s.t:} \quad \hat{\mathbf{X}} = \psi(\mathbf{X}), \quad \hat{\mathbf{X}}_m = \phi(\mathbf{X}). \end{aligned} \quad (8)$$

To further clarify the difference between our two models, we provide their functional diagram in Fig. 1. We also note that while the *Group Interpreter* provides greater interpretability, as we observed experimentally, it does so at the cost of increased complexity and a slight reduction of classification accuracy.

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 k such that $\mathrm{I}(\mathbf{X};\mathbf{Y}) = \mathrm{I}(\psi(\mathbf{X});\mathbf{Y})$?

Since the mapping function ψ is a composition of T_G and T_G^+, if information is preserved through both functions, then information is also preserved for ψ. Kraskov et al. [19] has shown that MI is invariant under diffeomorphism mappings. Therefore, we investigate if both functions are diffeomorphisms and formalize these findings in the following two lemmas with their proof in App. B.

Lemma 1. *T_G^+ is a diffeomorphism.*

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

Since k is always less than d when features are grouped together, our analysis proves that ψ 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. (10). In fact, in these cases, we proved the existence of a matrix G such that $I(\hat{X};Y) = I(X;Y)$. We formally state this finding in Theorem 3; the proof can be found in App. D.

Theorem 3. *Given $\mathcal{A} \subseteq \mathcal{X}$, let relevant features $\mathcal{U}$ be*

$$\mathcal{U} = \{X_j | I(X_j;\mathbf{Y}) \neq 0 \vee \exists \mathcal{A}\ I(X_j;\mathbf{Y}|\mathcal{A}) \neq 0\}, \quad (9)$$

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

$$k \geq |\mathcal{U}| + \mathbb{1}(|\mathcal{U}| \neq d) \quad (10)$$

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 coefficient and*

$$\mathcal{C} = \{X_j | \rho(X_j; \mathbf{Y}) \neq 0 \ \vee \ \exists \mathcal{A} \, \rho(X_j; \mathbf{Y} | \mathcal{A}) \neq 0\} \quad (11)$$

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.

Expected Code-Length Perspective. Optimizing Eqs. (3) and (8), allows us to discover the hidden group structure based on *representation* and *relevance*; this is done by learning an autoencoder that maximally retain these information. An alternative view is to interpret our autoencoder as trying to minimize the expected code length for $P(\mathbf{Y}|\mathbf{X})$ using $P(\mathbf{Y}|T_G(\mathbf{X}))$, i.e., the group structures we discover is the minimum coding length representation of the original data.

In Chen et al. [13], they show that the most important instance-wise features while masking out the rest optimizes for the shortest coding length representation. In this paper, we extend their work and show that instance-wise group feature selection using our formulation also optimizes for the shortest code-length representation. We formally state this expected code-length perspective below with its proof in App. F.

Theorem 5. *Let $f(x) = P(\mathbf{G}|x)$ the global optimum of*

$$f^*(x) = \arg\min_G E\left[\log(\frac{1}{P(\mathbf{Y}|z)})|x\right]. \quad (12)$$

is the global optimum solution to $\max_f I(\mathbf{Z}; \mathbf{Y})$, Furthermore every optimal solution of $\max_f I(\mathbf{Z}; \mathbf{Y})$ is degenerates to f^* almost surely.

3. Method Implementation

Approximating Mutual Information. Since the various distributions required to compute MI are difficult to obtain, we instead maximize MI's variational lower bound as a surrogate. We provide here a summary of the key conclusions while leaving the detail derivations to App. M.

By noting that $\psi(\mathbf{X}) = \hat{\mathbf{X}}$, we extend the work by Chen et al. [13] to include group discovery and show that the objective of $\max_\psi I(\mathbf{Y}; \hat{\mathbf{X}})$ is equivalent to solving

$$\max_\psi E_{\mathbf{Y}, \psi(\mathbf{X})}[\log(P(\mathbf{Y}|\psi(\mathbf{X}))]. \quad (13)$$

Since $P(\mathbf{Y}|\psi(\mathbf{X}))$ is unknown, it can be approximated with another distribution Q_θ parameterized by θ. By using this approximation, it can be shown via KL divergence that Eq. (13) is always greater than and lower bounded by

$$\max_{\psi,\theta} E_{Y, \psi(\mathbf{X})}[\log(Q_\theta(\mathbf{Y}|\psi(\mathbf{X}))]. \quad (14)$$

As a lower bound, finding the θ that maximizes Eq. (14) draws Q_θ closer to $P(\mathbf{Y}|\psi(\mathbf{X}))$. Therefore, $I(\mathbf{Y}; \hat{\mathbf{X}})$ can be alternatively maximized by simultaneously finding the θ and ψ that maximizes the empirical measure of Eq. (14),

$$\max_{\psi,\theta} \frac{1}{n} \sum_{i=1}^{n} \log(q_\theta(y_i|\psi(x_i))). \quad (15)$$

We note that the exemplary steps used for $I(\mathbf{Y}; \hat{\mathbf{X}})$ can also be applied to approximate $I(\hat{\mathbf{X}}_m; \mathbf{Y})$ and $I(\mathbf{X}; \hat{\mathbf{X}})$. For completeness, their step-by-step derivations can be found in App. N.

Sample Generation. To obtain the empirical estimate via averaging in Eq. (15), we need to generate samples from $P(\mathbf{Y}, \hat{\mathbf{X}}_m)$ and $P(\mathbf{X}, \hat{\mathbf{X}})$ and enable gradient descent [20] to optimize this average. Our generative model for $\hat{\mathbf{X}}$ and $\hat{\mathbf{X}}_m$ is shown in Fig. 2.

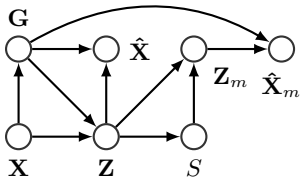

Figure 2. Graphical model for group learning representation

Following the graphical model, samples from $\hat{\mathbf{X}}$ and $\hat{\mathbf{X}}_m$ can be drawn through ancestral sampling [21] given the joint distributions of

$$P(\hat{\mathbf{X}}|\mathbf{Z}, \mathbf{G}) P(\mathbf{Z}|\mathbf{G}, \mathbf{X}) \underline{P(\mathbf{G}|\mathbf{X})} P(\mathbf{X}), \quad (16)$$

$$P(\hat{\mathbf{X}}_m|\mathbf{Z}_m, \mathbf{G}) P(\mathbf{Z}_m|\mathbf{S}, \mathbf{Z}) \underline{P(\mathbf{S}|\mathbf{Z})} P(\mathbf{Z}|\mathbf{G}, \mathbf{X}) \underline{P(\mathbf{G}|\mathbf{X})} P(\mathbf{X}). \quad (17)$$

Note that within Eqs. (16) and (17), only $\underline{\mathbf{P}(\mathbf{G}|\mathbf{X})}$ and $\underline{\mathbf{P}(\mathbf{S}|\mathbf{Z})}$ (underlined) are stochastic, while the rest are deterministic functions within T_G and T_G^+. Therefore, the ability to draw samples from $\mathbf{P}(\mathbf{G}|\mathbf{X})$ and $\mathbf{P}(\mathbf{S}|\mathbf{Z})$ is the crucial step. For these, we use the Gumbel-Softmax [22] as a reparameterization trick to generate samples from both $\mathbf{P}(\mathbf{G}|\mathbf{X})$ and $\mathbf{P}(\mathbf{S}|\mathbf{Z})$. Given a categorical distribution with

probabilities $p_1, ..., p_d$, Gumbel-Softmax is capable of generating differentiable samples of m-hot vectors. Therefore, by setting m to the number of important groups, $\mathbf{P}(\mathbf{S}|\mathbf{Z})$ can be sampled. Alternatively, we can also set $m = 1$ and sample a d-dimensional one-hot vector for each column of G, which in effect produces samples for G.

In practice, we use multiple neural networks to separately learn the distributions for each column of $\mathbf{P}(\mathbf{G}|\mathbf{X})$ and $\mathbf{P}(\mathbf{S}|\mathbf{Z})$. These networks use the softmax function to produce a categorical distribution which is then fed into the Gumbel-Softmax to generate their corresponding samples of G and S. Given G and S, each sample of $\hat{\mathbf{X}}$ is computed by $G^T G x_i$, and each sample of $\hat{\mathbf{X}}_m$ is computed by $G^T(Gx_i \odot S)$. Once $\mathbf{X}$ and $\hat{\mathbf{X}}_m$ is obtained, we let Q_θ be another neural network and jointly optimizes over θ. For a tutorial and additional implementation details on the Gumbel-Softmax and Q_θ refer to App. H.

Computational and Memory Complexities. Since our algorithm can be solved via gradient descent backpropagation, both gI and gC have efficient memory and computational complexities of $O(kd^2)$ and $O(nkd^2)$ respectively, where n is the number of samples and d is the original number of features. For a detailed derivation of these complexities, refer to App. O.

4. Experiments

Datasets. Although interpretability may be a highly subjective concept to prove experimentally, there are certain visual patterns that are unmistakable. We highlight our method's ability to produce these visual patterns with MNIST, and Fashion MNIST (F-MNIST) [23; 24]. Given these images, we ask our *Group Interpreter* to report the features (pixels in this case) that led it to conclude its classification, e.g., which pixels in an image of 7 lead the method to classify it as 7?

Additionally, we use nine synthetic datasets to evaluate the various theoretical claims of this paper. Since features can be redundant simultaneously through *Representation* and *Relevance*, we generate synthetic data based on different combinations of these redundancy patterns. Namely, we simulate three *Representation* redundancy patterns (D_1, D_2, D_3) and three *Relevance* redundancy patterns (R_1, R_2, R_3) to achieve nine datasets based on the combinations of these patterns. We then generate ten features $(x^1, ..., x^{10})$ with features $(x^5, ..., x^{10})$ as IID Gaussian noise $(N(0, I))$ and $(x^1, ..., x^4)$ as features that contain nine potential redundant patterns of (D_i, R_j). For each combination, we evaluate if the features organize together into the correct number of groups (k) and the correct redundancy patterns; we also ask if additional grouping of features yield highly accurate classification results. For additional details on the dataset statistics as well as how exactly the synthetic data were generated, refer to App. I.

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 [25; 26; 27; 28] 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 of our results rely on fully connected layers without any usage of CNN structure. All λs are identified by maximizing gI or gC given a validation set.

Competing Methods. We compare gC and gI against seven other state-of-the-art methods: L2X, DeepLift (DL), INVASE (INV), GLasso, Lasso, Shap, and Concrete Autoencoder (CAE) [13; 29; 15; 16; 30; 31; 14; 32]. These methods are mostly classification models that provide a certain level of interpretability. However, due to their vast difference in capabilities, we classify them in Table 2 based on seven interpretability properties with their descriptions listed in Table 1. For additional details on the competing methods, refer to App. J.

Interpretability Property	
Instance Wise	Provide different interpretation for each sample.
Model-Agnostic	Capable of interpreting results from any classifier.
Group Learning	The method separates the features into groups.
Importance of Groups	The method learns the most important groups.
Number of Groups	The method learns the number of groups.
Importance of Features	The method learns the most important features.
Classifier	The method performs classification.

Table 1. We evaluate existing interpretable models based on the definition of these seven interpretability properties.

	gI	gC	L2X	DL	INV	gLasso	Lasso	CAE	Shap
Instance Wise	**yes**	**yes**	**yes**	**yes**	**yes**	no	no	no	**yes**
Model-Agnostic	**yes**	**yes**	**yes**	**yes**	**yes**	no	no	no	**yes**
Group Learning	**yes**	**yes**	no	no	no	no	no	no	no
Importance of Groups	**yes**	no	no	no	no	no	no	no	no
Number of Groups	**yes**	**yes**	no	no	no	no	no	no	no
Importance of Features	**yes**	**yes**	**yes**	**yes**	**yes**	**yes**	**yes**	**yes**	**yes**
Classifier	**yes**	**yes**	**yes**	**yes**	**yes**	**yes**	**yes**	no	**yes**

Table 2. We evaluate the seven competing interpretable methods based on the property definitions in Table 1.

Results on 3 vs. 8 Digits and on a Four-Class F-MNIST. In Fig. 3, we compare the visual patterns generated by our gI model against the three most recent and advanced interpretable models, L2X, Shap and CAE. For each image, we ask each model to provide the most informative pixels in *white*; the top row is the original image while the results of each method are displayed below.

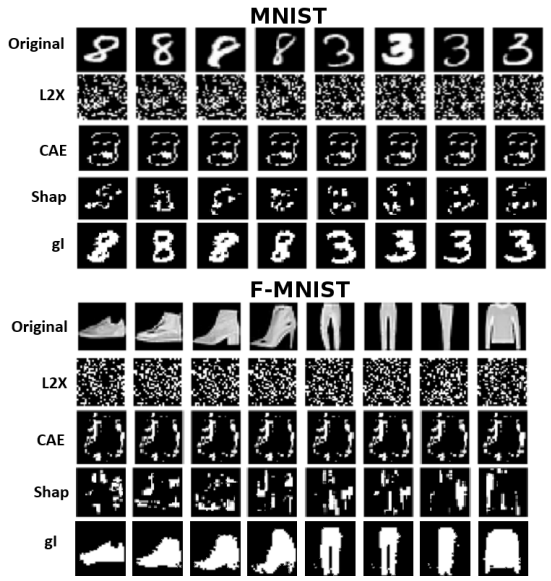

Figure 3. We ask each method to identify the most informative pixels in white. Notice that our method, gI, supplies the pixels almost identical to the shape of the object.

Although L2X and Shap both identify the most important pixels instance-wise, Fig.3 demonstrates the importance of also knowing pixels' relationship to each other. Consequently, even though both L2X and Shap perform instance-wise interpretation, the white pixels provide little visual evidence that links the important pixels to its classification.

CAE's selection of the most important pixels is global. Therefore, they are incapable of distinguishing the nuances between different classes, resulting in the same pixel choice across all samples. Because the number 3 and 8 are highly similar in shape, CAE is able to find an outline that resembles both 3 and 8. However, as we demonstrate later in Fig. 4, CAE no longer provides explanatory feedback as more complex shapes are added into the training set.

Contrary to L2X and CAE, gI supplied a visually indisputable explanation to its decisions. When asked to explain why an image of 8 and 3 are classified as the number 8 and 3, gI responded with the pixels in the exact shape of 8 and 3. Similar to the MNIST results, the pixels in the shape of a shoe, pants and shirt are all unmistakable explanation for F-MNIST when compared to its original image.

Results on Training on All 10 Digits. Our method is able to provide separate explanations for each class even if we increase the complexity of the training set. Instead of using 2 digits of similar shapes, we now use all 10 digits of vastly different shapes and display their results in Fig. 4. We note that since CAE is not instance-wise, we evaluated an additional setting, sCAE. Instead of giving sCAE all 10 digits, sCAE only trains on the same digit for each class, thereby significantly simplifies the data complexity under the sCAE setting.

Figure 4. We evaluate each method's ability to provide interpretable pixels given the shape variation of 10 digits.

Similar to Fig. 3, the most explanatory pixels supplied by L2X lack any visual connection to the original image. Shap performed slightly better with some discernible results. Since CAE can only supply one global result, it displayed the shape of 8 for all 10 digits. Therefore, none of the competing methods truly capture the shape diversity when more digits are added. Even when only one digit is trained under the sCAE settings, the explanatory pixels' link to the original images are not visually obvious. Yet, in contrast to these results, gI is capable of capturing the complexity of all shapes independent of the number of classes.

The Ability to Capture Style Variations. In addition to identify the different classes, an even more challenging task is to also capture the style variation within the same class. We evaluate our method's ability to capture the style difference given all 10 digits and present our results in Fig. 5; a larger collection of showcasing a variety of style variation results can be found in App. G. For these results, notice how the explanatory pixels follow closely to the style of the original image.

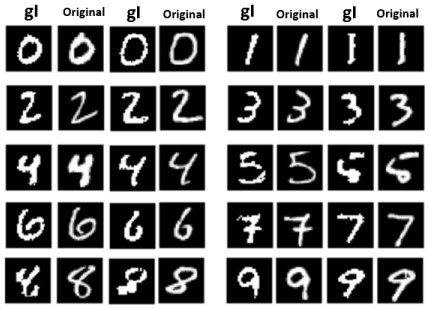

Figure 5. gI captures the style variations within the same class given 10 digits.

Classification Accuracy Results. Since L2X, Shap, and gI are also classifiers, we compared their classification accuracies given 2 and 10 digits in Table 3. Notice that while L2X and Shap performed slightly better on the simpler F-MNIST

and MNIST-2 (3 vs. 8) dataset, gI performed better on the more complex MNIST-10 (all 10 digits) dataset. Given the additional interpretation advantage of gI, the slight accuracy degradation on simpler datasets is a reasonable compromise.

Model	MNIST-2	MNIST-10	F-MNIST
gI	96.7 ± 0.2	$\mathbf{91.6 \pm 0.85}$	94.6 ± 0.6
L2X	97.1 ± 0.5	80.5 ± 2.5	$\mathbf{96.0 \pm 0.6}$
shap	$\mathbf{99.24 \pm 0.46}$	90.8 ± 1.9	94.45 ± 2.62

Table 3. Classification accuracies for gI, L2X and Shap.

Understanding Cases when a Classifier is Wrong. In addition to providing explanatory pixels that justifies the correct classifications, the same pixels can be used to understand the wrong predictions. We show in Fig. 6 two images of 5 and 9 that were mistakenly labeled as 6 and 5. Since gI is capable of capturing the style, it can be seen from the gI results why the two numbers were mislabeled.

Figure 6. Explanatory pixels can help us understand why the numbers 5 and 9 were mistakenly labeled as 6 and 5.

We have thus far asked gI to identify the the most important group of features from two groups. This is a reasonable request since the pixels are only black and white. However, when we request gI to identify the most important group of features from three groups, we discovered some interesting results; we report these additional results in App. K.

Experiments on Synthetic Data. We use synthetic datasets with known *Representation* and *Relevance Redundancies* to corroborate the various theoretical claims of this work. Namely, we answer the following questions.

Theorem 1: Can our model correctly identify the features that are highly dependent on each other?
Theorem 2: Can our model correctly identify the most relevant features in predicting $\mathbf{Y}$?
Noise Identification: Can our model correctly identify the noisy features?
Theorems 3 and 4: Is k based on Theorems 3 and 4 a tight lower bound?
Accuracy Comparison: How does the accuracy of our method compare to existing interpretable methods?

Types of Redundancy. We list the different redundancy patterns in Table 4 that we generated for our synthetic data. Recall that X_j indicates the jth feature. For *Representation Redundancy*, the D patterns, the features within the same parentheses are correlated with each other. For *Relevance Redundancy*, R patterns, the $P(Y = 1|X)$ is directly proportional to a function of the first four features, i.e., the features used within the function are also features that are highly correlated to the label.

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 4. The synthetic data are designed with these *Representation* and *Relevance* redundancy patterns.

Identifying *Representation* and Noisy Groups. We generate three datasets with the patterns $(D_1, R_2), (D_2, R_2)$, and (D_3, R_2). As shown in Fig. 7, by running gI on these datasets, it is able to perfectly identify the groups to match the D patterns in Table 4. Note that the black segments identify the most important groups based on *Representation Redundancy*, while the grey segments denote the noisy groups. Not only have we fully recovered the *Representation Redundant* features, we also placed the noisy features into the same group, labeling them as *Irrelevant* features (meaning that we have also correctly identified the *Relevant* features).

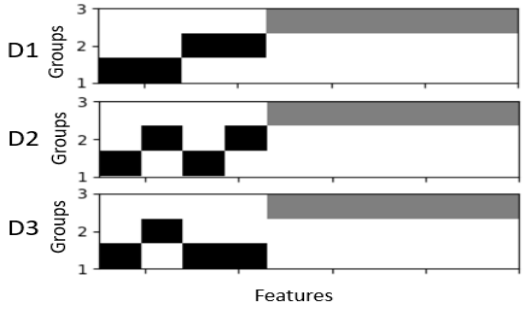

Figure 7. Identification of the *Representation* and noisy features.

Identifying the Number of Relevant Groups, k. We generate three alternative patterns of $(D_2, R_1), (D_2, R_2)$, and (D_2, R_3) to help confirm the tightness of our lower bound as recommended by Theorems 3 and 4. We plot the classification accuracy at each increment of k in Fig. 8 and circle the lower bound for each dataset. Ideally, given a tight lower bound, the classification accuracy should approach 100% after the circled values.

As predicted by our theorems, the classification accuracy monotonically increases as the number of group k increases.

	Classification Acc (**gC**)			k (**gC**)			Group Relevance Acc (**gC**)			Classification Acc (**gI**)			m (**gI**)			Group Importance 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 ± 0.0	99.5 ± 1.5	3	2	2	99.8 ± 0.3	100 ± 0.0	100 ± 0.0	91.0 ± 3.4	99.6 ± 1.2	98.5 ± 2.2	1	1	1	100 ± 0.0	100 ± 0.0	100 ± 0.0
R_2	100 ± 0.0	100 ± 0.0	99.6 ± 0.4	3	3	3	100 ± 0.00	100 ± 0.00	99 ± 1.8	92 ± 3	95.4 ± 2.2	94 ± 1	2	2	2	100 ± 0	100 ± 0.0	100 ± 0.0
R_3	98.9 ± 0.8	97.6 ± 0.6	99.2 ± 0.8	3	3	3	100 ± 0.00	100 ± 0.00	99.5 ± 0.9	94.4 ± 3	94.7 ± 1.6	90.9 ± 5	2	2	2	100 ± 0.0	100 ± 0.0	100 ± 0.0

Table 5. Measuring gC and gI's ability to identify the most relevant groups using nine redundancy patterns. Note that both gC and gI are capable of perfectly identifying the relevant groups while achieving a high classification accuracy.

Crucially, notice how the accuracy approaches 100% immediately after the lower bound. Although, not proven analytically, we have here shown empirically the quality and tightness of our bound.

Learning the k Most Relevant Groups. Given k, does our model also correctly predict the most relevant groups? To answer this question, we generated nine datasets of every combinations of D and R redundancies. For each feature of each dataset, we compare its importance and relevance designation as predicted by our model against the ground truth. We then report the percentage of the accurate predictions for both gI and gC in Table 5.

By using the suggested lower bound as a guideline for k, Table 5 shows that for gC, we are able to identify the relevant group almost perfectly. Similarly for gI, we achieved perfect accuracy on determining the most important groups. In addition, we are able to achieve high classification accuracy on both gC and gI while finding the most relevant groups, thereby confirming Theorem 2.

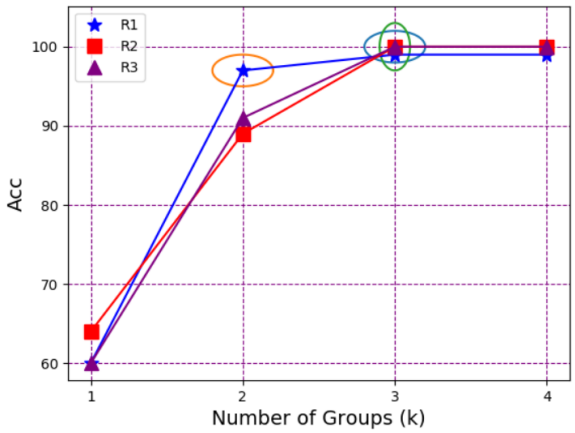

Figure 8. The classification prediction accuracy (reflecting the mutual information between the class label and the groups of features) with respect to different numbers of groups of features k. The k values for the circled points are computed from Theorem 3.

Accuracy Comparisons. In Table 6, we lastly compare our *Group Classifier* model purely as a classifier against other interpretable classification models. 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. This implies that 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 definitively outperform all alternative methods.

	Data	D_1	D_2	D_3	$D_1 + D_2$
gC	R_1	**98.4± 1**	99.7±0.46	95± 3.8	**98.7 ± 1.3**
	R_2	**100 ± 0.0**	**99.5 ± 0.5**	**100 ± 0.0**	**100 ± 0.0**
	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
DL	R_1	75.7 ± 10.6	99.4 ±1.2	86.9 ± 11.0	78.1 ± 9.9
	R_2	93.8 ± 4	94.8 ± 3	95.4 ± 2.2	89.5 ± 3
	R_3	68.5 ± 5	73.8 ± 5	69.2 ± 8.1	96.3 ± 1.7
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	50 ± 0.0	50 ± 0.0	48 ± 0.4
	R_2	50 ± 0.08	48 ± 0.4	49 ± 0.4	5 ± 0.4
	R_3	74 ± 1	83 ± 0.5	60 ± 13	77 ± 2

Table 6. A comparison of classification accuracy (in percentage) against various competing methods. Notice that gC almost always achieves the highest accuracy. This difference is especially prominent in the case of $D_1 + D_2$.

By marking the best results of each dataset as bold in Table 6, we can clearly see that gC is almost always the best performing classifier. As expected, the accuracy difference is particularly prominent with $D_1 + D_2$ since gC accurately models the hidden assumptions of the data.

5. Conclusion

Our model discovers the hidden group structure with two types of information redundancies: *Representation Redundancy* and *Relevance Redundancy*. Our theoretical contribution demonstrates how, by learning a mapping through an autoencoder, the ideal group structure can be found. In addition, we have proposed a theoretical lower bound to determine number of groups. Our experimental results are

easily reproducible, visually compelling, and achieve high classification accuracies. In summary, we have introduced and developed a novel interpretable method to explain black-box models by learning the hidden feature group interaction structure and which groups are important for each sample instance.

References

[1] Zachary C Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

[2] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[3] Chieh Wu, Jared Miller, Yale Chang, Mario Sznaier, and Jennifer Dy. Solving interpretable kernel dimension reduction. *arXiv preprint arXiv:1909.03093*, 2019.

[4] 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.

[5] Hadi Zarkoob. Feature selection for gene expression data based on hilbert-schmidt independence criterion. Master's thesis, University of Waterloo, 2010.

[6] 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.

[7] 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.

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

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

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

[11] Mahdokht Masaeli, Glenn Fung, and Jennifer G Dy. From transformation-based dimensionality reduction to feature selection. 2010.

[12] 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.

[13] Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan. Learning to explain: An information-theoretic perspective on model interpretation. *arXiv preprint arXiv:1802.07814*, 2018.

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

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

[20] Dinh Nho Hao and D Lesnic. The cauchy problem for laplace's equation via the conjugate gradient method. *IMA Journal of Applied Mathematics*, 65(2):199–217, 2000.

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

[22] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

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

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

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

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

[27] 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.

[28] 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.

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

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

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

[32] Abubakar Abid, Muhammad Fatih Balin, and James Zou. Concrete autoencoders for differentiable feature selection and reconstruction. *arXiv preprint arXiv:1901.09346*, 2019.

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

$$I(X;Y) = D_{\mathrm{KL}}(P_{(X,Y)} \| P_X \otimes P_Y) \tag{18}$$

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

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right). \tag{19}$$

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:

$$I(Y; X_1, \ldots, X_n) = I(Y; X_1) + I(Y; X_2|X_1) + \cdots + I(Y; X_n|X_1, \ldots, X_{n-1}). \tag{20}$$

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 $I(X;Y) = 0 \not\Rightarrow I(X;Y|Z) = 0$.

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 5.1.** *Using the encoder T_G as f will not enhance the mutual information. meaning:*
>
> $$I(T_G(X); Y) \leq I(X;Y).$$

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

$$\forall z_i, z_j \in \mathbb{R}^k,\ T_G^+(z_i) = T_G^+(z_j) \to z_i = z_j, \tag{21}$$

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

$$\forall i \neq j,\ 1 \leq i < j \leq k \quad \langle \hat{g}_i, \hat{g}_j \rangle = 0. \tag{22}$$

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

$$G^T \cdot z_i = z_{i,1}\hat{g}_1 + \cdots + z_{i,k}\hat{g}_k. \tag{23}$$

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. 23 we obtain

$$T_G^+(z_i - z_j) = (z_{i,1} - z_{j,1})\hat{g}_1 + \cdots + (z_{i,k} - z_{j,k})\hat{g}_k = 0. \tag{24}$$

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

$$(z_{i,1} - z_{j,1})\hat{g}_1 + \cdots + (z_{i,k} - z_{j,k})\hat{g}_k = 0 \tag{25}$$

$$(z_{i,1} - z_{j,1})\hat{g}_\eta^T\hat{g}_1 + \cdots + (z_{i,k} - z_{j,k})\hat{g}_\eta^T\hat{g}_k = \hat{g}_\eta^T(0) = 0 \tag{26}$$

Using Eq. (22), 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

$$(z_{i,\eta} - z_{j,\eta})\langle \hat{g}_\eta, \hat{g}_\eta \rangle = 0. \tag{27}$$

From Eq. (27), 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 $\text{Im}(T_G^+)$, i.e., the entire $\text{Im}(T_G^+)$ 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

$$I(Z;Y) = I(T_G^+(Z);Y). \tag{28}$$

□

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:

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

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

□

C. Proof for Theorems 1 and 2

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

Theorem 1. *The characteristic features induced by the optimal maximization of $I(\hat{\mathbf{X}}; \mathbf{X})$ satisfies Def. (1) such that*

$$\arg\max_{G} I(\hat{\mathbf{X}}; \mathbf{X}) = I(\mathbf{X}; \mathbf{X}) \rightarrow I(\mathbf{X}; \mathbf{X}|\mathbf{Z}) = 0 \tag{29}$$

Conversely, the characteristic features that satisfies Def. (1) maximizes $I(\hat{\mathbf{X}}; \mathbf{X})$ such that

$$\arg\min_{G} I(\mathbf{X}; \mathbf{X}|\mathbf{Z}) = 0 \rightarrow I(\hat{\mathbf{X}}; \mathbf{X}) = I(\mathbf{X}; \mathbf{X}) \tag{30}$$

Proof.

Proof for Condition. (29): Based on Lemma 4, we know that $I(\hat{X}; X)$ is upper bounded by $I(X; X)$, i.e., the mutual information can never exceed the entropy of X. Therefore, given the optimal G^*, $I(\hat{X}; X) = I(X; X)$. Thus, condition Eq. (29) can be proven by showing that if

$$I(\hat{X}; X) = I(X; X) \tag{31}$$

then

$$I(X; X|Z = \tilde{z}) = 0. \tag{32}$$

This can be proven given the following three observations:

- Using Lemmas 1 and 3, MI is preserved under T_G^+ and therefore $I(Z = \tilde{z}; X) = I(\hat{X}; X)$. By combining this result to Eq. (31), we see that
$$I(Z = \tilde{z}; X) = I(\hat{X}; X) = I(X; X). \tag{33}$$
- We note that $Z = T_G(X)$ and that T_G is a deterministic function. Therefore, $P(T_G(X) = \tilde{z}|X = x) = 1$ for a specific $\tilde{z}$ given x and $P(T_G(X) \neq \tilde{z}|X = x) = 0$. Given this observation we see that
$$\begin{aligned} P(X, Z = \tilde{z}) &= P(X, T_G(X) = \tilde{z}) && (34)\\ &= P(T_G(X) = \tilde{z}|X)P(X) && (35)\\ &= [1]P(X) && (36)\\ P(X, Z = \tilde{z}) &= P(X). && (37)\end{aligned}$$
Therefore, it leads to the conclusion that
$$I(X, Z = \tilde{z}; X) = I(X; X). \tag{38}$$
- We lastly write the chain rule for $I(X, Z = \tilde{z}; X)$ as
$$I(X, Z = \tilde{z}; X) = I(Z = \tilde{z}; X) + I(X; X|Z = \tilde{z}). \tag{39}$$
By using Eq. (38) and (33), the equality becomes
$$I(X; X) = I(X; X) + I(X; X|Z = \tilde{z}). \tag{40}$$
Therefore, $I(X; X|Z = \tilde{z})$ must equal to 0 and condition Eq. (32) is proven.

Proof for Condition. (30): Start by leveraging the result from Eq. (38) to obtain

$$\begin{aligned} I(X; X) &= I(X, Z; X) && && (41)\\ &= I(Z; X) + I(X; X|Z) && \text{using the chain rule} && (42)\\ &= I(Z; X) + 0 && \text{using the condition assumption} && (43)\\ &= I(\hat{X}; X) && \text{using the Eq. (33)} && (44)\\ &= H(X) && \text{using the definition of Entropy.} && (45)\end{aligned}$$

□

Theorem 2. *The characteristic features induced by the optimal maximization of $I(\hat{\mathbf{X}};\mathbf{Y})$ satisfies Def. (2) such that*

$$\arg\max_G I(\hat{\mathbf{X}};\mathbf{Y}) = I(\mathbf{X};\mathbf{Y}) \rightarrow I(\mathbf{X};\mathbf{Y}|\mathbf{Z}) = 0. \tag{46}$$

Conversely, the characteristic features that satisfies Def. (2) maximizes $I(\hat{\mathbf{X}};\mathbf{Y})$ such that

$$\arg\min_G I(\mathbf{X};\mathbf{Y}|\mathbf{Z}) = 0 \rightarrow I(\hat{\mathbf{X}};\mathbf{Y}) = I(\mathbf{X};\mathbf{Y}). \tag{47}$$

Proof.
Proof for Eq. 46: The proof follows the similar direction as Theorem 1: using the data processing lemma 4, $I(\hat{X};Y) \leq I(X;Y)$, hence the optimal solution G^* should satisfies, $I(\hat{X};Y) = I(X;Y)$. The goal is to show that given

$$I(\hat{X};Y) = I(X;Y) \tag{48}$$

then

$$I(X;Y|Z) = 0. \tag{49}$$

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

$$I(Z;Y) = I(T_G^+(Z);Y) = I(\hat{X};Y) = I(X;Y). \tag{50}$$

From this observation, it leads to the following derivation:

$$I(X,Z;Y) = I(Z;Y) + I(X;Y|Z) \quad \text{via MI chain rule} \tag{51}$$
$$I(X;Y) = I(Z;Y) + I(X;Y|Z) \quad \text{via condition Eq. (37)}, P(X,Z) = P(X) \tag{52}$$
$$I(X;Y) = I(X;Y) + I(X;Y|Z) \quad \text{via condition Eq. (50)}, I(Z;Y) = I(X;Y). \tag{53}$$

Since $I(X:Y) = I(X;Y)$, then the condition $I(X;Y|Z) = 0$ must be true. □

D. Proof for Theorem 3

Theorem 3. *Given $\mathcal{A} \subseteq \mathcal{X}$, define relevant features $\mathcal{U}$ as $\mathcal{U} = \{X_j | I(X_j;\mathbf{Y}) \neq 0 \vee \exists \mathcal{A}\ I(X_j;\mathbf{Y}|\mathcal{A}) \neq 0\}$, then, $\exists G$ such that $I(T_G(\mathbf{X});\mathbf{Y}) = I(\mathbf{X};\mathbf{Y})$ if*

$$k \geq |\mathcal{U}| + \mathbb{1}(|\mathcal{U}^c| \neq 0) \tag{54}$$

where $\mathbb{1}(|\mathcal{U}^c| \neq 0)$ is an indicator function with one when $|\mathcal{U}^c| \neq 0$ and zero when $|\mathcal{U}^c| = 0$.

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

$$\mathcal{U}^c = \{X_j \,|\, I(X_j;\mathbf{Y}) = 0 \text{ and }, \forall \mathcal{A}\ I(X_j;\mathbf{Y}|\mathcal{A}) = 0\}. \tag{55}$$

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. Assume $|\mathcal{U}^c| = n$ which we denote by $\hat{x}^1, \dots \hat{x}^n$:

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

$$I(\hat{x}^1, \dots \hat{x}^n, \mathcal{U}; \mathbf{Y}) = I(\mathcal{U}; \mathbf{Y}) + I(\hat{x}^1; \mathbf{Y}|\mathcal{U}) + I(\hat{x}^2; \mathbf{Y}|\mathcal{U}, \hat{x}^1) + \cdots + I(\hat{x}^n; \mathbf{Y}|\mathcal{U}, \hat{x}^1, \dots, \hat{x}^{n-1}) \tag{56}$$

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

$$I(\mathcal{U}^c, \mathcal{U}; \mathbf{Y}) = I(\hat{x}^1, \dots \hat{x}^n, \mathcal{U}; \mathbf{Y}) = I(\mathcal{U}; \mathbf{Y}) + 0 + \cdots + 0. \tag{57}$$

Eq. 57 also leads to the conclusion of this lemma:

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

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:*

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

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

$$I(Y; \mathcal{U}, \mathcal{U}^c) = I(Y; \mathcal{U}) + I(Y; \mathcal{U}^c|\mathcal{U}) = I(Y; \mathcal{U}) + 0 \tag{58}$$

□

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

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

$$T_G|_U := U \to \mathrm{Im}(U) \quad \textit{is bijective} \quad \textit{and} \quad \mathrm{Im}(U) \cap \mathrm{Im}(U^c) = \{0\} \tag{59}$$

where $dim(U) = \mathbb{R}^{|\mathcal{U}|}$, 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

$$I(X; Y) = I(T_G(X); Y). \tag{60}$$

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 X and Y. Note the encoder $T_G : X \to Z$ is from $\mathbb{R}^d \to \mathbb{R}^k$. let $\hat{e}^1, \dots, \hat{e}^k$ the basis axis in R^k, and $\hat{x}^1, \dots, \hat{x}^k$ the values of each axis. Assume $|\mathcal{U}| = m$, T_G is bijective with respect to set U. Thus, without loss of generality, assume $e^1, \dots, e^m$ axis in U are mapped to $\hat{e}^1, \dots, \hat{e}^m$ in Z. And axis in U^c are mapped to $\hat{e}^{m+1}, \dots, \hat{e}^k$. Hence T_G acts as an identity for U, because U^c sends to orthogonal subspace we can say the following: $I(T_G(X); Y) = I(U, \mathrm{Im}(U^c); Y)$, which we now it is equal to $I(X; Y)$ based on lemma 6. Furthermore, based on Lemma 3 for U , we can derive that:

$$I(U; Y) = I(x^1, \dots, x^m; Y) = I(\hat{x}^1, \dots, \hat{x}^m; Y) = I(T_G(U); Y) \leq I(\hat{x}^1, \dots, \hat{x}^k; Y) \leq I(X; Y) \tag{61}$$

where $I(\hat{x}^1, \dots, \hat{x}^k; Y) \leq I(X; Y)$ is based on data processing Lemma 4. Hence $I(X; Y) \leq I(T_G(X); Y) = I(\hat{x}^1, \dots, \hat{x}^k; Y) = I(X; Y)$, which concludes the proof.

□

Based on 7, as long as we can satisfy Eq. 59, we can guarantee the preservation of mutual information between X and 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 $\mathrm{Im}(U)$. 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

$$\mathcal{C} = \{X_j | \rho(X_j; Y) \neq 0 \vee \exists \mathcal{A}\, \rho(X_j; Y|\mathcal{A}) \neq 0\} \tag{62}$$

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

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

$$\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2. \tag{63}$$

Also we know that $\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2$ if we let $\mathcal{U}_1 = \{X_j | I(X_j; Y) \neq 0\}$ and $\mathcal{U}_2 = \{X_j | \exists \mathcal{A}, I(X_j; Y|\mathcal{A}) \neq 0\}$. 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:

$$P(Z_1, Z_2|\mathcal{A}) = P(Z_1|\mathcal{A})P(Z_2|\mathcal{A}). \tag{64}$$

Using the condition from Eq. (64), we can compute the conditional expectation where

$$\begin{aligned} E_{Z_1,Z_2}[Z_1 Z_2] &= \int_{z_1}\int_{z_2} z_1 z_2 p(z_1, z_2|\mathcal{A}) dz_1 dz_2 && (65)\\ &= \int_{z_1}\int_{z_2} z_1 z_2 p(z_1|\mathcal{A}) p(z_2|\mathcal{A}) dz_1 dz_2 && (66)\\ &= \left[\int_{z_1} z_1 p(z_1|\mathcal{A}) dz_1\right]\left[\int_{z_2} z_2 p(z_2|\mathcal{A}) dz_2\right] && (67)\\ &= E_{Z_1}[Z_1] E_{Z_2}[Z_2]. && (68) \end{aligned}$$

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

$$E_{Z_1,Z_2}[Z_1 Z_2] - E_{Z_1}[Z_1]E_{Z_2}[Z_2] = 0. \tag{69}$$

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: $\mathbb{1}(|\mathcal{C}| \neq d) \leq |\mathcal{U}| + \mathbb{1}(|\mathcal{U}| \neq d)$.

□

F. Proof for Theorem 5

Theorem 5. *Let $f(x) = P(\mathbf{G}|x)$ the global optimum of*

$$f^*(x) = \arg\min_f E[\log(\frac{1}{P_m(Y|z)})|x]. \tag{70}$$

is the global optimum solution to $\arg\max_f I(Z;Y)$*, Furthermore every optimal solution of* $\arg\max_f I(Z;Y)$ *is degenerates to f^* almost surely.*

Proof. First we need to prove the following properties, on conditional independency of our model.

Property 7: Given the graphical model as shown in Fig. 9, then

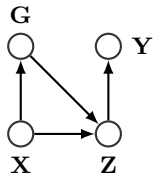

Figure 9. Graphical model for group learning representation

$$\int_y\int_z\int_G\int_x p(y,z,G,x)log[p(y|z)]dydzdGdx = \int_y\int_z p(y,z)log[p(y|z)]dydz. \tag{71}$$

Proof. Note that since y is conditionally independent of G, x given z, we have $p(y,z,G,x) = p(y|z)p(G,x|z)p(z)$ or

$$\int_y\int_z\int_G\int_x p(y,z,G,x)log[p(y|z)]dydzdGdx = \int_y\int_z\int_G\int_x p(y|z)p(x,G|z)p(z)log[p(y|z)]dydzdGdx. \tag{72}$$

We can now separate the integral of x, G as

$$\int_y\int_z p(y|z)\left[\int_G\int_x p(x,G|z)dGdx\right]p(z)log[p(y|z)]dydz. \tag{73}$$

Regardless of the z given to the inner integral, it will always sum up to 1, and therefore, it can be cancelled out to get

$$\int_y\int_z p(y|z)p(z)log[p(y|z)]dydz = \int_y\int_z p(y,z)log[p(y|z)]dydz \tag{74}$$

□

Given Property 7, the proof for Theorem 5 follows a similar approach as Theorem 1 by Chen et al. [13]. The optimal solution G^* induced from f^* in the optimal minimum of Eq. 70 Hence:

$$E_{Y|X}[\log(\frac{1}{P(Y|Z)})|X] \geq E_{Y|X}[\log(\frac{1}{P(Y|T_{G^*}(X))})|X]. \tag{75}$$

By flipping the log we can change the direction of inequality:

$$\begin{aligned} &E_{Y|X}[\log(\frac{1}{P(Y|Z)})|X] \geq E_{Y|X}[\log(\frac{1}{P(Y|T_{G^*}(X))})|X]. \\ \xrightarrow{\text{flipping the log}} &E_{Y|X}[\log(P(Y|Z))|X] \leq E_{Y|X}[\log(P(Y|T_{G^*}(X)))|X]. \end{aligned} \tag{76}$$

Taking the expectation of X from both side and subtracting $E_{Y|X}(\log(P(Y))$ leads to the following results (more details of this part in **Details of this computation**):

$$I(Z;Y) \leq I(Z^*;Y) \tag{77}$$

Which proves that G^* is the optimal groups.

Conversely: assume there is an optimal solution $P(G|x)$ for some x that is not part of solution of $P(G^*|x)$. Define set $\mathcal{E}$ to be set of x with such a property, meaning $\mathcal{E} = \{x|P(G \notin G^*|X = x) > 0\}$, Thus,

For any $x \in \mathcal{E}$, we have the following:

$$E_{Y|X}[\log(\frac{1}{P(Y|Z)})|X = x] > E_{Y|X}[\log(\frac{1}{P(Y|T_G^*(x))})|X = x]. \tag{78}$$

For $x \notin \mathcal{E}$, the optimal solution is in G^*, meaning:

$$E_{Y|X}[\log(\frac{1}{P(Y|Z)})|X = x] \geq E_{Y|X}[\log(\frac{1}{P(Y|T_G^*(x))})|X = x]. \tag{79}$$

Similar as forward proof, taking expectation over X for all the elements and subtracting $E_{Y|X}(\log(P(Y))$ leads to the following results:

$$I(Z;Y) < I(Z^*;Y)$$

which is a contradiction to the fact that $I(Z;Y)$ is the maximum mutual information.

Details of this computation: In here we prove the taking expectation of X from both side of Eq. 76 subtracting $E_{Y|X}(\log(P(Y))$ leads to the definition of mutual information for our graphical model:

$$I(Z;Y) = \int_{z\in\mathcal{Z}}\int_{y\in\mathcal{Y}} p(z,y)\log\left(\frac{p(z,y)}{p(z),p(y)}\right)dzdy \tag{80}$$

$$= \int_{z\in\mathcal{Z}}\int_{y\in\mathcal{Y}} p(z,y)\log\left(p(y|z)\right)dzdy + \text{const} \tag{81}$$

where the constant term is $E_{Y|X}(\log(P(Y))$, which we are going to to subtract from Eq. 76, hence dropping this term would leads to the following: Using Property 7, Eq. (81) becomes:

$$= \int_{x\in\mathcal{X}}\int_{G\in\mathcal{G}}\int_{z\in\mathcal{Z}}\int_{y\in\mathcal{Y}} p(z,y,G,x)\log\left[p(y|z)\right]dzdydxdG \qquad \text{Using Property 7} \tag{82}$$

$$= \int_{x\in\mathcal{X}}\int_{G\in\mathcal{G}}\int_{z\in\mathcal{Z}}\int_{y\in\mathcal{Y}} p(z|y,G,x)p(y,G,x)\log\left[p(y|z)\right]dzdydxdG \qquad z \text{ is known given } G,x \tag{83}$$

$$= \int_{x\in\mathcal{X}}\int_{G\in\mathcal{G}}\int_{y\in\mathcal{Y}}\left[\int_{z\in\mathcal{Z}} p(z|y,G,x)\log\left[p(y|z)\right]dz\right]p(y,G,x)dydxdG \qquad \text{rearrange integral} \tag{84}$$

Given G and x, z becomes a known value $\tilde{z}$ and $p(z|y,G,x)$ becomes a Dirac delta function with 1 at $\tilde{z}$ and 0 everywhere else. The inner integral therefore becomes

$$= \int_{x\in\mathcal{X}}\int_{G\in\mathcal{G}}\int_{y\in\mathcal{Y}} p(y,G,x)log[p(y|\tilde{z})]dydxdG \tag{85}$$

$$= \int_{x\in\mathcal{X}}\int_{y\in\mathcal{Y}} p(y|x)p(x)log[p(y|\tilde{z})]dydx \qquad \text{marginalize } G \tag{86}$$

$$= \int_{x\in\mathcal{X}} p(x)\left[\int_{y\in\mathcal{Y}} p(y|x)log[p(y|\tilde{z})]dy\right]dx \qquad \text{rearrange integral} \tag{87}$$

$$= E_X E_{Y|x}\left[log[p(Y|\tilde{z})]|X = x\right] \tag{88}$$

$$= E_X E_{Y|x}\left[log\frac{1}{p(Y|T_G(x))}|X = x\right]. \tag{89}$$

$$\tag{90}$$

□

Theorem 5 is specifically talks about the encoder objective function. For the entire autoencoder, a similar theorem can also be stated as an extension.

Theorem 5 Auto-encoder extension.

Let $f(x) = P(\mathbf{G}|x)$ *the global optimum of*

$$f^*(x) = \arg\min_f E_{Y|x}[\log(\frac{1}{P(Y|\hat{x})})|x]. \tag{91}$$

is the global optimum solution to $\max_f I(\hat{X};Y)$*, Furthermore every optimal solution of* $\max_f I(\hat{X};Y)$ *is degenerates to* f^* *almost surely.*

Proof. Similar property as **Property 7** can be applied to the relationship of $\hat{X}$ and Y.

Property 8: Given the graphical model as shown in Fig. 2, then

$$\int_y\int_z\int_G\int_x\int_{\hat{x}} p(y,z,\hat{x},G,x)log[p(y|\hat{x})]dydzdGdxd\hat{x} = \int_y\int_z p(y,\hat{x})log[p(y|\hat{x})]dyd\hat{x}. \tag{92}$$

Proof. Note that since y is conditionally independent of z, G, x given $\hat{x}$, we have $p(y,z,G,x,\hat{x}) = p(y|\hat{x})p(G,x,z|\hat{x})p(\hat{x})$, i.e., we get

$$\int_{\hat{x}}\int_y\int_z\int_G\int_x p(y|\hat{x})p(x,G,z|\hat{x})p(\hat{x})log[p(y|\hat{x})]dydzdGdxd\hat{x}. \tag{93}$$

We can now separate the integral of x, G, z as

$$\int_y\int_{\hat{x}} p(y|\hat{x})p(\hat{x})\left[\int_G\int_x\int_z p(x,G,z|\hat{x})dGdxdz\right]log[p(y|\hat{x})]dyd\hat{x}. \tag{94}$$

Regardless of the $\hat{x}$ given to the inner integral, it will always sum up to 1, and therefore, it can be cancelled out to get

$$\int_y\int_{\hat{x}} p(y,\hat{x})log[p(y|\hat{x})]dyd\hat{x}. \tag{95}$$

□

The optimal solution G^* induced from f^* in the optimal minimum of Eq. 91 Hence:

$$E_{Y|X}[\log(\frac{1}{P(Y|\hat{X})})|X] \geq E_{Y|X}[\log(\frac{1}{P(Y|\hat{X}^*)})|X]. \tag{96}$$

Because the inequality holds for any G, it holds for the expectation over $P(G|X)$ as well, meaning:

$$\begin{aligned} &E_{G|X}E_{Y|X}[\log(\frac{1}{P_m(Y|\hat{X})})|X] \geq E_{Y|X}[\log(\frac{1}{P(Y|\hat{X}^*)})|X]. \\ \xrightarrow{\text{flipping the log}} &E_{G|X}E_m[\log(P_m(Y|\hat{X}))|X] \leq E_{Y|Z}[\log(P(Y|\hat{X}^*))|X]. \end{aligned} \tag{97}$$

Taking the expectation of X from both side and subtracting $E_{Y|X}(\log(P(Y))$ leads to the following results (more details of this part in **Details of this computation**):

$$I(\hat{X};Y) \leq I(\hat{X}^*;Y) \tag{98}$$

Which proves that G^* is the optimal groups.

Conversely: assume there is an optimal solution $P(G|x)$ for some x that is not part of solution of $P(G^*|x)$. Define set $\mathcal{E}$ to be set of x with such a property, meaning $\mathcal{E} = \{x|P(G \notin G^*|X = x) > 0\}$, Thus,

For any $x \in \mathcal{E}$, we have the following:

$$E_{G|X} E_m[\log(\frac{1}{P_m(Y|Z)})|X = x] > E_m[\log(\frac{1}{P_m(Y|\hat{X}^*)})|X = x]. \tag{99}$$

For $x \notin \mathcal{E}$, the optimal solution is in G^*, meaning:

$$E_{G|X} E_{Y|x}[\log(\frac{1}{P(Y|\hat{X})})|X = x] \geq E_{Y|x}[\log(\frac{1}{P(Y|\hat{X}^*)})|X = x]. \tag{100}$$

Similar as forward proof, taking expectation over X for all the elements and subtracting $E_{Y|X}(\log(P(Y))$ leads to the following results:

$$I(\hat{X}; Y) < I(\hat{X}^*; Y)$$

which is a contradiction to the fact that $I(\hat{X}; Y)$ is the maximum mutual information.

Details of computations This proof starts from $I(\hat{X}; Y)$.

$$I(\hat{X}; Y) = \int_{\hat{x}} \int_y p(\hat{x}, y) log[p(y|\hat{x})] d\hat{x} dy + \text{const.} \tag{101}$$

Using Property 8, and ignoring the the constant value, this integral can be rewritten in

$$= \int_z \int_G \int_x \int_{\hat{x}} \int_y p(\hat{x}, y, z, G, x) log[p(y|\hat{x})] d\hat{x} dy dx dz dG \tag{102}$$

$$= \int_z \int_G \int_x \int_{\hat{x}} \int_y p(\hat{x}|y, z, G, x) p(y, z, G, x) log[p(y|\hat{x})] d\hat{x} dy dx dz dG \tag{103}$$

$$= \int_z \int_G \int_x \int_y \left[\int_{\hat{x}} p(\hat{x}|y, z, G, x) log[p(y|\hat{x})] d\hat{x} \right] p(y, z, G, x) dy dx dz dG. \tag{104}$$

Since $p(\hat{x}|y, z, G, x)$ obtains a known value $\hat{\mathbf{x}}$, $p(\hat{x}|y, z, G, x)$ becomes a Dirac function with 1 at $\hat{\mathbf{x}}$ and 0 elsewhere, the integral becomes

$$= \int_z \int_G \int_x \int_y p(y, z, G, x) log[p(y|\hat{\mathbf{x}})] dy dx dz dG \tag{105}$$

$$= \int_x \int_y p(y, x) log[p(y|\hat{\mathbf{x}})] dy dx \qquad \text{marginalize } z \text{ and } G \tag{106}$$

$$= \int_x \int_y p(y|x) p(x) log[p(y|\hat{\mathbf{x}})] dy dx \qquad \text{marginalize } z \text{ and } G \tag{107}$$

$$= \int_x p(x) \int_y [p(y|x) log[p(y|\hat{\mathbf{x}})] dy] \, dx \qquad \text{Separate out } p(x) \tag{108}$$

$$= E_{p(x)} E_{p(y|x)} \left[log[\frac{1}{p(y|\hat{\mathbf{x}})}] | X = x \right] \tag{109}$$

□

G. Complete Collection of *Style Adaptation* Results

In Figure 10, we see the complete collection of *Style Adaptation* Results.

H. Using the Gumbel-Softmax for the *Group Interpreter*

One can use Gumbel-Softmax to sample a continuous approximation of a m hot vector. Assume that we are given a categorical distribution with probabilities $p_1, \ldots, p_k$. If we wish to generate a single sample of a one-hot vector based on this

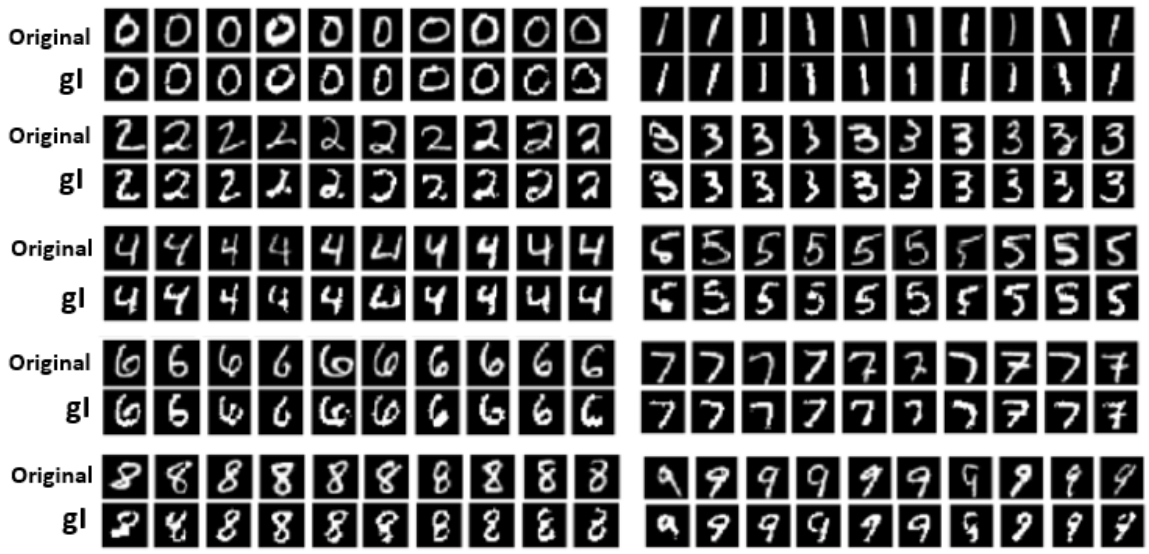

Figure 10.

categorical distribution, then the first step is to generate k perturbations from a uniform distribution, i.e., each perturbation is generated with

$$\epsilon_i = -\log(-\log u_i), u_i \sim \text{Uniform}(0, 1). \tag{110}$$

The output of these k perturbations is $[\epsilon_1, ..., \epsilon_k]$. The i^{th} element of the one-hot vector is then computed with

$$C_i = \frac{\exp(\log p_i + \epsilon_i)/\tau}{\sum_{j=1}^{D} \exp(\log p_j + \epsilon_j)/\tau} \tag{111}$$

where τ is a parameter that control the sharpness of the one-hot vectors. Note that since the Gumbe-Softmax method results in a differentiable one-hot vectors, the values in $\mathbb{1}$ are not exactly 1s and 0s, but an approximation. The τ value controls the sharpness of these approximations where a small τ of approximately 0 yield a very sharp of nearly 1 and 0.

By repeating Eq. (111) for each of the k groups, a one-hot vector can be generated for feature j can be denoted as $\mathbf{C}_j$ where

$$\mathbf{C}_j = [C_1, C_2, ..., C_k]^T. \tag{112}$$

We again repeated Eq. (112) for each of the d features to generate the columns of G as

$$G = [\mathbf{C_1}, \mathbf{C_2}, ..., \mathbf{C_d}]. \tag{113}$$

To generate samples for S where m elements are 1s, we repeat Gumbel-softmax m times to create a matrix V where

$$V = [\mathbf{C_1}, ..., \mathbf{C_m}]. \tag{114}$$

By setting the k^{th} row of V as $\bar{V}_k$, the k^{th} element of S can be set to the maximum value in $\bar{V}_k$ where

$$S_k = \max \bar{V}_k. \tag{115}$$

Note that in the event where the newly generated $\mathbf{C_j}$ is a repeat of a previous column, it is discarded, and a new $\mathbf{C_j}$ is generated in its place.

In practice, instead of sampling from $P(\mathbf{G}|\mathbf{X})$, we are actually sampling one column at a time using Gumbel-Softmax. We use multiple neural networks with a softmax output to represent the distribution of each column $P(\mathbf{g}_j|\mathbf{X})$. The output of the softmax becomes $p_1, ..., p_k$ that is used by the Gumbel-Softmax to generate the samples. By repeating this process d times and merging the results, it is equivalent to generating a sample of G. Once G is obtained, then we can accordingly obtain

samples from $\hat{\mathbf{X}}$ and $\hat{\mathbf{X}}_m$ to compute the objective, Eq. (15). From the objective, backpropagation is possible to obtain the weights of the networks.

We define Q_θ to be any family of function that maps the data to a simplex. For this, we use neural networks parameterized by θ to represent this distribution. Given the flexibility of the neural network, it is flexible enough to model $P(Y|\hat{X})$ and $P(X|\hat{X}_m)$ to approximate the mutual information.

I. Addition Information on Data Statistics and Synthetic Data Generation

- A complete description of the data collection process, including sample size:

  Synthetic Dataset: For tuning the parameter λ we use a validations dataset once and set the parameter for the rest of 12 Synthetic dataset $\{(D_i, R_j)\}$. After tuning the parameter we separate the data into training and test with 100000 and 1000 sample size respectively.

  Real Dataset: For Real dataset, we did the similar procedure of tuning the parameter on validation and then fix the parameter for all real dataset without having validation and separate the data into training and test based on default spilt in keras which is 50,000 for training and 10,000 for the test.

- An explanation of any data that were excluded, description of any pare-processing step:

  Synthetic Dataset: There were no prepossessing step involved for this dataset.

  Real Dataset: We normalized the colors between zero and one by dividing everything by 255.

- The exact number of evaluation runs: For our model we run 2 epochs for the training through all the dataset. And we change the seed of the randomness, and compute the mean and std.

J. Additional Details on Competing Methods

We compare our model with the following models:

- Lasso: In statistics and machine learning, lasso (least absolute shrinkage and selection operator; also Lasso or LASSO) is a regression analysis method that can do variable selection and regularization in order to increase the prediction accuracy and interpretability of the statistical model it produces.
- Group Lasso: Is a version of Lasso that has a grouping structure of the features and perform regression on top of that. In this experiment, we will provide the grouping of features for the model. The only problem with GLasso is when the groups are not global, and we will give one of the grouping structure to the model.
- L2X, Shap, DeepLift: These models are deep neural networks model, which can perform instance-wise feature selection. These model cannot learn and do not use the grouping structure. But they can be used as models for interpretability and variable selection. We will compare our performance with these models.
- INVASE: This model is an extension over L2X, which can perform instance-wise feature selection without the need to specify the number of selected features in advance.

K. Results of MNIST When $m = 3$

Although we do not have any theoretical reason to explain the results, we noticed that when we requested gI to identify the 3 most important groups in white pixels, the three groups identified by gI were edges, interior points, and exterior points. Listing the 3 groups as G1, G2, and G3, the results can be seen in Fig. 11.

L. Notations

L.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: $\mathcal{X} = \{(\lambda^1, \ldots, \lambda^d) | \forall j \lambda^j = x^j \vee \lambda^j = 0\}$,

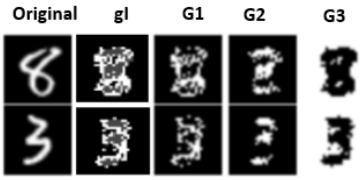

Figure 11. Learning 3 important groups. G1 are pixels of edges, G2 are pixels of the interior and G3 are the pixels of the background.

and $\mathcal{A} \subseteq \mathcal{X}$. 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 $I(\mathcal{X}; Y) = I(X; Y)$.

M. Variational lower bound

Given a function ψ_γ that is parameterized by γ, we wish to solve the problem

$$\arg\max_\gamma \text{MI}(\psi_\gamma(X); Y). \tag{116}$$

MI, however, is difficult to compute due to its requirement of both the joint and marginal distributions. Chen et al. [13] circumvented this problem by calculating the variational lower bound. Since the lower bound is tractable, it can be maximized as a surrogate to Eq. (116). We start the derivation by following the definition of MI as

$$\arg\max_\gamma \text{MI}(\psi_\gamma(X); Y) = \arg\max_\gamma \int_{x\in\mathcal{X}} \int_{y\in\mathcal{Y}} p(\psi_\gamma(x), y) \log \left(\frac{p(\psi_\gamma(x), y)}{p(\psi_\gamma(x)), p(y)} \right) dxdy. \tag{117}$$

Since $p(\psi_\gamma(x), y) = p(y|\psi_\gamma(x))p(\psi_\gamma(x))$, we replace the numerator term inside the log and cancel out $p(\psi_\gamma(x))$, the objective then becomes

$$\arg\max_\gamma \text{MI}(\psi_\gamma(X); Y) = \arg\max_\gamma \int_{x\in\mathcal{X}} \int_{y\in\mathcal{Y}} p(\psi_\gamma(x), y) \log \left(\frac{p(y|\psi_\gamma(x))}{p(y)} \right) dxdy. \tag{118}$$

The integrals can be rewritten into

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

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

$$\arg\max_\gamma \int_{x\in\mathcal{X}} \int_{y\in\mathcal{Y}} p(\psi_\gamma(x), y) \log \left[p(y|\psi_\gamma(x)) \right] dxdy, \tag{119}$$

$$\arg\max_\gamma \int_{x\in\mathcal{X}} \int_{y\in\mathcal{Y}} p(y|\psi_\gamma(x)) p(\psi_\gamma(x)) \log \left[p(y|\psi_\gamma(x)) \right] dxdy, \tag{120}$$

$$\arg\max_\gamma \int_{x\in\mathcal{X}} p(\psi_\gamma(x)) \left[\int_{y\in\mathcal{Y}} p(y|\psi_\gamma(x)) \log \left[p(y|\psi_\gamma(x)) \right] dy \right] dx. \tag{121}$$

$$\arg\max_{\gamma} E_{\psi_\gamma(X)}\left[\int_{y\in\mathcal{Y}} p(y|\psi_\gamma(x))\log\left[p(y|\psi_\gamma(x))\right]dy\right], \tag{122}$$

$$\arg\max_{\gamma} E_{\psi_\gamma(X)} E_{Y|\psi_\gamma(X)}\left[\log(p(Y|\psi_\gamma(X)))\right] = \arg\max_{\gamma} E_{Y,\psi_\gamma(X)}\left[\log(p(Y|\psi_\gamma(X)))\right]. \tag{123}$$

If we look closer at the inner expectation, $E_{Y|\psi_\gamma(X)}\left[log(p(Y|\psi_\gamma(X)))\right]$, we do not assume to have $p(Y|\psi_\gamma(X))$. Instead, we wish to approximate the distribution via another distribution $q_\theta(Y|\psi_\gamma(X))$ that is parameterized by θ. 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

$$\begin{aligned} KL(p||q) &= \int_{y\in\mathcal{Y}} p(y|\psi_\gamma(x))\log\left[\frac{p(y|\psi_\gamma(x))}{q_\theta(y|\psi_\gamma(x))}\right]dy, \\ &= \int_{y\in\mathcal{Y}} p(y|\psi_\gamma(x))\log(p(y|\psi_\gamma(x))) - p(y|\psi_\gamma(x))\log(q_\theta(y|\psi_\gamma(x)))dy, \\ &= E_{Y|\psi_\gamma(X)}[\log p(y|\psi_\gamma(x)))] - E_{Y|\psi_\gamma(X)}[\log q_\theta(y|\psi_\gamma(x)))]. \end{aligned}$$

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

$$E_{Y|\psi_\gamma(X)}[\log p(y|\psi_\gamma(x)))] - E_{Y|\psi_\gamma(X)}[\log q_\theta(y|\psi_\gamma(x)))] \geq 0 \tag{124}$$

$$E_{Y|\psi_\gamma(X)}[\log p(y|\psi_\gamma(x)))] \geq E_{Y|\psi_\gamma(X)}[\log q_\theta(y|\psi_\gamma(x)))]. \tag{125}$$

The inequality suggests that $E_{Y|\psi_\gamma(X)}[\log(q_\theta(Y|\psi_\gamma(X)))]$ is a lower bound of $E_{Y|\psi_\gamma(X)}[\log(p(Y|\psi_\gamma(X)))]$, and they are equal only when $p = q$. Therefore, by finding the θ that maximizes $E_{Y|\psi_\gamma(X)}[\log(q_\theta(Y|\psi_\gamma(X)))]$ is equivalent to finding the best approximation of $E_{Y|\psi_\gamma(X)}[\log(p(Y|\psi_\gamma(X)))]$. Next, we take Inequality (125) and rewrite each term back in terms of its integration, we obtain

$$\int_{y\in\mathcal{Y}} p(y|\psi_\gamma(x))\log p(y|\psi_\gamma(x)))dy \geq \int_{y\in\mathcal{Y}} p(y|\psi_\gamma(x))\log q_\theta(y|\psi_\gamma(x)))dy. \tag{126}$$

The key realization of this inequality is that given any $\psi_\gamma(X)$, 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.

$$\int_{x\in\mathcal{X}} p(\psi_\gamma(x))\int_{y\in\mathcal{Y}} p(y|\psi_\gamma(x))\log p(y|\psi_\gamma(x)))dydx \geq \int_{x\in\mathcal{X}} p(\psi_\gamma(x))\int_{y\in\mathcal{Y}} p(y|\psi_\gamma(x))\log q_\theta(y|\psi_\gamma(x)))dydx, \tag{127}$$

$$\int_{x\in\mathcal{X}}\int_{y\in\mathcal{Y}} p(y,\psi_\gamma(x))\log p(y|\psi_\gamma(x)))dydx \geq \int_{x\in\mathcal{X}}\int_{y\in\mathcal{Y}} p(y,\psi_\gamma(x))\log q_\theta(y|\psi_\gamma(x)))dydx, \tag{128}$$

$$E_{Y,\psi_\gamma(X)}\left[\log(p(Y|\psi_\gamma(X)))\right] \geq E_{Y,\psi_\gamma(X)}\left[\log(q_\theta(Y|\psi_\gamma(X)))\right] \tag{129}$$

By looking at the relationship between Eq. (123) and (129), notice that if we simultaneously maximize θ and γ using q_θ, the θ term would help us find the closest approximation of p while the γ term would help us maximize the MI objective. Therefore, to maximize Eq. (116), we can use q as a surrogate and instead maximize

$$\max_{\theta,\gamma} E_{Y,\psi_\gamma(X)}\left[\log(q_\theta(Y|\psi_\gamma(X)))\right] \tag{130}$$

If we notice that the samples from our data is coming from $p(X, Y)$, therefore, the samples of the data can be used to compute the expectation empirically in the new objective below as

$$\max_{\theta,\gamma} \frac{1}{n}\sum_{i=1} \log(q_\theta(y_i|\psi_\gamma(x_i))). \tag{131}$$

Note that for the case where Y denotes the label, the probability of y_i equaling its label is 1. Therefore, the Eq. (131) can be equivalently written as the Cross-Entropy objective where

$$\max_{\theta,\gamma} \frac{1}{n}\sum_{i=1} p(y_i)\log(q_\theta(y_i|\psi_\gamma(x_i))). \tag{132}$$

N. Lower bound for MI$(\hat{X}_m; Y)$ and MI$(\hat{X}; X)$

Assuming that $\psi_\gamma(X) = \hat{X}_m$, we can apply the derivation from Appendix M to find the lower bound for MI$(\hat{X}_m; Y)$ where

$$\arg\max_{\gamma} \text{MI}(\hat{X}_m; Y) \geq \arg\max_{\gamma,\theta} E_{Y,\psi_\gamma(X)}[log q_\theta(Y|\psi_\gamma(X))]. \tag{133}$$

By applying the graphical model shown in Fig. 12, $\hat{X}_m$ can be sampled via ancestral sampling given the joint distribution of

$$p(\hat{X}_m|Z_m, G)p(Z_m|S, Z)p(S|Z)p(Z|G, X)p(G|X)p(X). \tag{134}$$

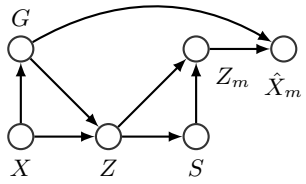

Figure 12. Graphical model for group learning representation

Assuming that we have chosens some q_θ and ψ_γ, by initializing θ and γ with some random values, we are able to take pairs (x_i, y_i) from the data to generate samples coming from $p(Y, \hat{X}_m)$ and compute the empirical expectation in Eq. (133) with

$$\max_{\theta,\gamma} \frac{1}{n} \sum_{i=1} \log[q_\theta(y_i|\hat{x}_i)]. \tag{135}$$

We can use the same process to derive the lower bound for MI$(\hat{X}; X)$. Similarly, we only need to focus on generating samples from $\hat{X}$ using the graphical model in Fig. 13. Again, we can use ancestral sampling with Eq. (136) to generate samples of $\hat{X}$.

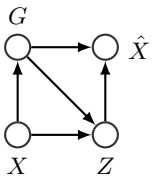

Figure 13. Graphical model for group learning representation

$$P(\hat{\mathbf{X}}|\mathbf{Z}, \mathbf{G})P(\mathbf{Z}|\mathbf{G}, \mathbf{X})\underline{P(\mathbf{G}|\mathbf{X})}P(\mathbf{X}). \tag{136}$$

O. 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. Knowing G determines the auto-ecnoder $\psi(x_i) = G^T G x_i$ for a sample x_i. thus that is just matrix multiplication. So for **gI** the complexity is $O(Nkd^2)$. For **gC**, we have another neural net for learning the projection map π_m, which the complexity is k^2 which lead to $\mathbf{Z}_m$. Having $\mathbf{Z}_m$, lead to $\hat{\mathbf{X}}_m$ by matrix multiplication. We used the last neural net from $\hat{\mathbf{X}}_m$ to predict the class lables. Assumming we have C classes, lead to the complexity of Cd. Hence the overall complexity for **gC** is $O(N(kd^2 + k^2 + Cd))$, assuming $C << d$ and $k << d$, complexity of **gC** 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.