INTERPRETABILITY WITH FULL COMPLEXITY BY CONSTRAINING FEATURE INFORMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Interpretability is a pressing issue for machine learning. Common approaches to interpretable machine learning constrain interactions between features of the input, rendering comprehensible the effects of features on a model's output at the expense of model complexity. We approach interpretability from a different angle: constrain the information about the features utilized by the model, without any restrictions on the complexity of the model. Borrowing from information theory, we use the Distributed Information Bottleneck to find optimal compressions of each feature that maximally preserve information about the output. The learned information allocation, by feature and by feature value, is a rich source of interpretability well-suited to problems with many features and complex feature interactions. The central object of analysis is not a single trained model, but rather a spectrum of models serving as approximations that leverage variable amounts of information and range from the uninformative to the most performant model obtainable. Information is allocated to features by their relevance to the output, tackling the problem of feature selection with inclusion/exclusion existing on a learned continuum. The optimal compression of each feature—at every stage of approximation—allows fine-grained inspection of how feature values are similar or distinct with regards to the prediction. We develop a framework for extracting insight from the spectrum of approximate models and demonstrate its utility on a range of tabular datasets.

1 INTRODUCTION

Interpretability is a pressing issue for machine learning (ML) (Doshi-Velez & Kim, 2017; Fan et al., 2021; Rudin et al., 2022). Models are growing more complex, and machine learning is increasingly integrated into fields where flawed decisions can have serious ramifications (Caruana et al., 2015; Rudin et al., 2018; Rudin, 2019).

Interpretability is not a binary property that machine learning methods have or do not: it is the degree to which a learning system can be probed and comprehended (Doshi-Velez & Kim, 2017). Various constraints on the learning system are incorporated, commonly on features and their interactions, that bring about a degree of comprehensibility (Molnar et al., 2020; Molnar, 2022). As opposed to explainable AI that happens post-hoc after a black-box model is trained, interpretable methods engineer the constraints into the model from the outset (Rudin et al., 2022). A common route to interpretability enforces that the effects of features combine in a simple (e.g. linear) manner, greatly restricting the space of possible models in exchange for comprehensibility.

In this work, we place no restrictions on the model complexity and instead constrain the amount of information available to the model. The model identifies optimal information from the features, granting a measure of salience of each feature of the input, a spectrum of models utilizing different amounts of optimal information about the input, and a learned compression of each feature highlighting in fine-grained detail from where the information is coming. The central object of interpretation is no longer a single optimized model, but rather a spectrum containing a rich signal about what information flows from inputs to outputs at all levels of fidelity.

The information constraint we incorporate is a variant of the Information Bottleneck (IB) (Tishby et al., 2000; Asoodeh & Calmon, 2020). The IB is a framework for extracting the relevant information in a relationship, though it lacks interpretability because the extracted information is free be any

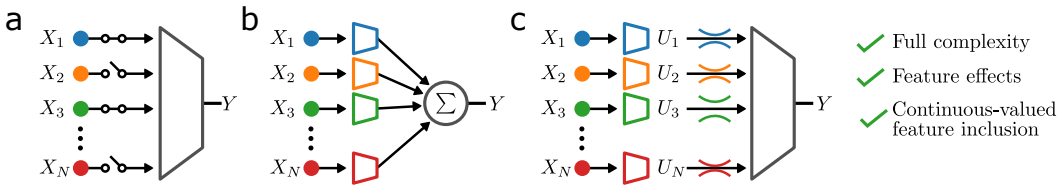

Figure 1: **Restricting information flow from features grants interpretability with full complexity.** **(a)** Feature selection methods optimize a binary mask on features to uncover the most informative ones. **(b)** Interpretable ML methods sacrifice model complexity in exchange for comprehensible feature interactions (shown is a GAM with a neural network to transform each feature). **(c)** Our method places an information bottleneck penalty on each feature, with no constraints on complexity before or after the bottlenecks. The features X_i are compressed to optimal representations U_i, which communicate the most relevant information to the rest of the model. Compressing the features in this way allows feature inclusion/exclusion to exist on a continuum. The learned compression scheme for each feature grants insight into feature effects on the model output.

black-box function of the input. Instead, by siloing the information from each feature and using a sum-rate constraint in the form of the Distributed IB (Estella Aguerri & Zaidi, 2018; Murphy & Bassett, 2022), we introduce a constraint at an appropriate level in the model's processing: after every feature is processed and before any interaction effects between the features can be incorporated. Before and after the bottlenecks, arbitrarily complex processing can occur, allowing the Distributed IB to be used in scenarios with many features and complex feature interactions.

By restricting the flow of information into a model, we acquire a rich and novel signal about the relationship in the data and the inner workings of the model. We develop a framework for extracting insight from the signal of interpretability acquired by the Distributed IB, and demonstrate the following strengths:

1. **Full complexity of feature processing and feature interactions.** The method is compatible with arbitrary neural network architectures before and after the bottlenecks.
2. **Gradual addition of detail.** By way of a spectrum of models serving as approximations to the relationship, a relationship is decomposed into comprehensible steps.
3. **Interpretability of broad strokes and fine details.** The information allocation by feature allows a high-level analysis
4. **Intuitive to implement.** The information penalty is the same KL divergence term used in β-VAE (Higgins et al., 2017); the method is simply a probabilistic encoder for each feature.

We compare the Distributed IB to other approaches in interpretable ML on a variety of datasets.

2 Related Work

An information theoretic analysis of a relationship, whether in data itself or in the form of a trained machine learning model, and specifically focused on the features of an input, has relevance to several large areas of research.

Feature selection and variable importance. Feature selection methods isolate the most important features in a relationship (Fig. 1a), both for interpretability and to reduce computational burden (Chandrashekar & Sahin, 2014; Cai et al., 2018). When higher order feature interactions are present, the combinatorially large search space presents an NP-hard optimization problem (Chandrashekar & Sahin, 2014). Modern feature selection methods navigate the search space with feature inclusion weights, or masks, that can be trained as part of the full stack with the machine learning model (Fong & Vedaldi, 2017; Balın et al., 2019; Lemhadri et al., 2021; Kolek et al., 2022).

Many classic methods of statistical analysis work to the same end: to identify the most important features in data. Canonical correlation analysis (CCA) (Hotelling, 1936) relates sets of random variables by finding transformations that maximize linear correlation. Modern extensions broaden the

class of transformations to include kernel methods (Hardoon et al., 2004) and deep learning (Andrew et al., 2013; Wang et al., 2016). Another line of research generalizes the linear correlation to use mutual information instead (Painsky et al., 2018). Analysis of variance (ANOVA) finds a decomposition of variance by features (Fisher, 1936; Stone, 1994); extensions incorporate ML-based transforms (Märtens & Yau, 2020). Variable importance (VI) methods quantify the effect of different features on an outcome (Breiman, 2001; Fisher et al., 2019).

Comprehensible machine learning systems. To make machine learning systems more comprehensible, constraints are incorporated to reduce the complexity of the system. Generalized Additive Models (GAMs) require feature effects to combine linearly (Fig. 1b), though each effect can be an arbitrary function of a feature (Chang et al., 2021; Molnar, 2022). By using neural networks to learn the transformations from raw feature values to effects, Neural Additive Models (NAMs) maintain the interpretability of GAMs with more expressive effects (Agarwal et al., 2021). Pairwise interactions between effects can be included for more complex models (Yang et al., 2021; Chang et al., 2022), though complications arise involving degeneracy and the number of effect terms to inspect grows rapidly. Regularization can be used to encourage sparsity in the ML model so that fewer components—weights in the network (Ng, 2004) or even features (Lemhadri et al., 2021)—conspire to produce the final prediction.

Information theory. The proposed method can also be seen as a regularization to encourage sparsity of information. It is based upon the Information Bottleneck, a broad framework that extracts the most relevant information in a relationship (Tishby et al., 2000) and has proven useful in representation learning (Alemi et al., 2016; Wang et al., 2019; Asoodeh & Calmon, 2020; Bang et al., 2021). However, the standard IB has one bottleneck for the entire input, preventing an understanding of where the relevant information originated and hindering interpretability.

Distributing bottlenecks to multiple sources of data has been of interest in information theory as a solution to multiterminal source coding, or the "CEO problem" (Berger et al., 1996; Estella Aguerri & Zaidi, 2018; Aguerri & Zaidi, 2021; Steiner & Kuehn, 2021). Only recently has the Distributed IB been utilized for interpretability (Murphy & Bassett, 2022), whereby features of data serve as the distributed sources. With this formulation, the extracted information from each feature can be used to understand the role of the features in the relationship of interest.

3 METHODS

At a high level, we introduce a penalty on the amount of information used by a model about each feature of the input. The penalty has the effect of sorting the information about every feature, such that information that is least helpful to the prediction task is discarded. Varying the magnitude of the penalty yields a prioritization of information about the features and a spectrum of models, or approximations, that utilize different amounts of information when making a prediction. The methodology shares strengths with feature selection, which surfaces informative features, and interpretable machine learning methods, where the feature effects can be comprehended.

Let $X, Y \sim p(x, y)$ be random variables representing the input and output of a relationship to be modeled from data. The mutual information between two random variables is a general measure of their statistical dependence and is defined as the expected reduction of entropy of one variable after learning the value of the other:

$$I(X;Y) = H(Y) - H(Y|X), \tag{1}$$

with $H(X) = \mathbb{E}_{x\sim p(x)}[-\log p(x)]$ Shannon's entropy (Shannon, 1948).

The Information Bottleneck (IB) (Tishby et al., 2000) is a learning objective to extract the most relevant information from X about Y. It optimizes a constrained channel U that conveys information about X, realized as a stochastic representation $U \sim p(u|x)$. The representation is found by minimizing a loss consisting of two competing mutual information terms balanced by a scalar parameter β,

$$\mathcal{L}_{\text{IB}} = \beta I(U;X) - I(U;Y). \tag{2}$$

The first term is the bottleneck—acting as a penalty on information passing into U—with β determining the strength of the bottleneck. In the limit where $\beta \to 0$, the bottleneck is fully open and all information from X may be freely conveyed into U in order for it to be maximally informative about

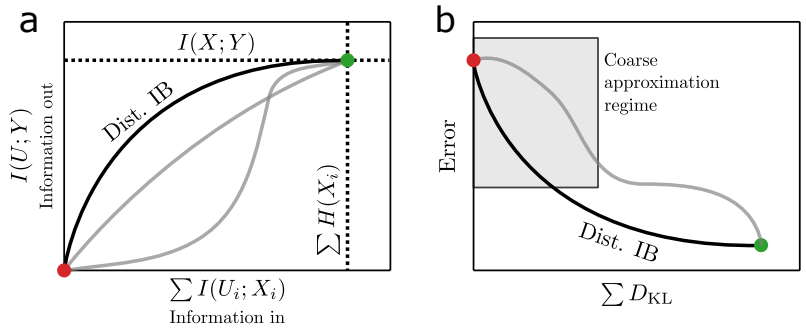

Figure 2: **Visualizing the tradeoff between information into the model and predictive power out. (a)** The distributed information plane displays the sum total information from all features, encoded independently of one another. While there are many possible paths, the Distributed IB objective finds the optimal trajectory in terms of the most information out for the least in. Increasing β traverses the optimal trajectory from the green dot—all bottlenecks fully open, all information is available for prediction—to the red dot—a model that is independent of the input. **(b)** A pragmatic alternative displays the sum of the KL-divergences against the particular error metric being optimized, though the horizontal is now only a proxy for the information into the model. Trajectories are a helpful means to display the spectrum of models found through optimization. In the large $\sum D_{\mathrm{KL}}$ regime, the effect of the bottlenecks are negligible and the models are essentially unconstrained ML. A Distributed IB optimization gradually increases β, the strength of the bottlenecks, shedding information. We focus on the low-information regime where the coarse approximations are more comprehensible.

Y. As β increases, only the most relevant information in X about Y becomes worth conveying into U, until eventually it is optimal for U to be uninformative random noise.

Because mutual information is generally difficult to estimate from data (Saxe et al., 2019; McAllester & Stratos, 2020), variants of IB replace the mutual information terms of Eqn. 2 with bounds amenable to deep learning (Alemi et al., 2016; Achille & Soatto, 2018; Kolchinsky et al., 2019). We follow the Variational Information Bottleneck (VIB) (Alemi et al., 2016), which learns representations U in a framework resembling that of β-VAEs (Higgins et al., 2017).

A sample x is encoded as a distribution in representation space $p(u|x) = f(x, \phi, \epsilon)$ with a neural network parameterized by weights ϕ and a source of noise $\epsilon \sim \mathcal{N}(0, 1)$ to facilitate the "reparameterization trick" (Kingma & Welling, 2014). The bottleneck restricting the information from X takes the form of the Kullback-Leibler (KL) divergence—$D_{\mathrm{KL}}(w(x)||z(x)) = \mathbb{E}_{x\sim w(x)}[-\log\,(z(x)/w(x))]$—between the encoded distribution $p(u|x)$ and a prior distribution $r(u) = \mathcal{N}(0, 1)$. As the KL divergence tends to zero, all representations become indistinguishable from the prior and from each other, and therefore uninformative. A representation u is sampled from $p(u|x)$ and then decoded to a distribution over the output Y with a second neural network parameterized by weights ψ, $q(y|u) = g(u, \psi)$. The second term of Eqn. 2, measuring the predictability of Y from the representation U, is replaced with the expected cross entropy between the predicted distribution and the ground truth. In place of Eqn. 2, the following loss may be minimized with standard gradient descent methods:

$$\mathcal{L}_{\mathrm{VIB}} = \beta D_{\mathrm{KL}}(p(u|x)||r(u)) - \mathbb{E}[\log q(y|u)]. \tag{3}$$

The Distributed IB (Estella Aguerri & Zaidi, 2018) concerns the optimal scheme of compressing multiple sources of information as they relate to a single relevance variable Y. Let $\{X_i\}$ be the features of X; the Distributed IB encodes each feature X_i independently of the rest, and then integrates the information from all features to predict Y. A bottleneck is installed after each feature by way of a compressed representation U_i (Fig. 1c), and the full set of representations $U_X = \{U_i\}$ is used as input to a model—wholly unconstrained—to predict Y. The same mutual information bounds of the Variational IB (Alemi et al., 2016) can be applied in the Distributed IB setting (Aguerri & Zaidi, 2021),

$$\mathcal{L}_{\mathrm{DIB}} = \beta \sum_i D_{\mathrm{KL}}(p(u_i|x_i)||r(u_i)) - \mathbb{E}[\log q(y|u_X)]. \tag{4}$$

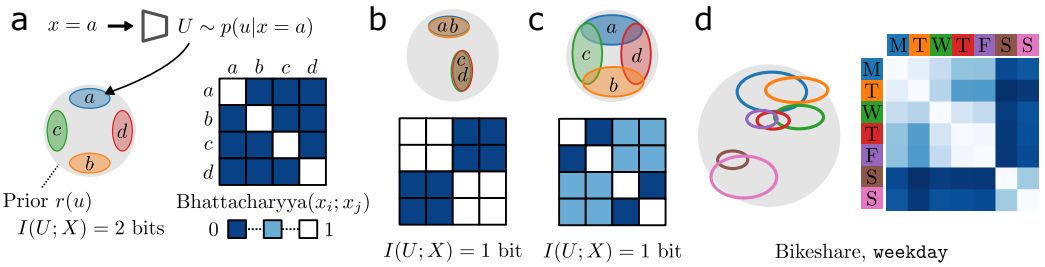

Figure 3: **Soft clustering and distinguishability matrices**. **(a)** An input value is encoded to a distribution in u. The Bhattacharyya coefficient quantifies how distinguishable the distributions are for different encodings, which we express as a matrix. The diagonal (white) values are indistinguishable and everything else (blue) is distinguishable, yielding 2 bits of information. **(b)** 1 bit of information allows a hard clustering of the values, such that the ab cluster is distinguishable from the cd cluster, but no more than that. **(c)** 1 bit also allows a soft clustering, where some pairs of values are partially distinguishable. **(d)** When trained with a two-dimensional embedding space per feature on the Bikeshare dataset, we recover a soft clustering of the day of the week. Weekend is distinguishable from weekday, with intermediate values of distinguishability otherwise.

Utilizing arbitrary error metrics. Instead of quantifying predictive power through expected cross entropy, another error metric can be used in combination with the constraint on the information about the features. For instance, mean squared error (MSE) may be used instead

$$\mathcal{L} = ||y - \hat{y}||^2 + \beta \sum D_{\text{KL}}, \tag{5}$$

with y and $\hat{y}$ the scalar prediction and ground truth values, respectively. The connection to the Information Bottleneck is diminished, as β is no longer a unitless ratio of information out to information in: it becomes a ratio of the error metric to information in, and the optimal range must be found anew for each problem. Optimization will nevertheless yield a spectrum of approximations prioritizing the most relevant information about the input features, showing that the heart of this framework is regularization on the information about input features inspired by the Distributed Information Bottleneck rather than a strict application of it.

The information plane. Tishby & Zaslavsky (2015) popularized the visualization of learning dynamics in the "information plane", a means to display how the amount of information into a particular transformation changes the amount of information that can be extracted. We can instead use a distributed information plane (Fig. 2a) to compare the total information about the features with the information about the output Y. A trajectory represents a spectrum of relationships between $\{X_i\}$ and Y of varying fidelity. Trajectories have two universal endpoints: the origin (no information in, no information out; red point in Fig. 2a) and at $(\sum H(X_i), I(X, Y))$ (all information in, mutual information between X and Y out; green point in Fig. 2a).

A pragmatic alternative to the distributed information plane replaces $\sum I(U_i; X_i)$ with its upper bound used during optimization, the sum of KL divergences across the features. Additionally, we replace the vertical axis $I(U; Y)$ with the particular error used for training. Switching the vertical axis from mutual information to a measure of error has the effect of inverting the trajectories: the uninformative point is now the *maximum* error with zero $\sum D_{\text{KL}}$ (red point in Fig. 2b).

In the course of finding the relevant information in each X_i about Y, we optimize Eqn. 4 with β increased logarithmically over the course of training and obtain a trajectory in the distributed information plane. Training begins with negligible information restrictions ($\beta \approx 0$) so that the relationship may be found without obstruction (Wu et al., 2020; Lemhadri et al., 2021), and ends once all information is gone ($\beta \approx 1$). As β increases, the total KL divergence decreases; the KL divergence of each features' representation tracks the allocation of information to features of X.

Confusion matrices and soft clustering As each feature value x_i is encoded to a probability distribution $p(u_i|x_i)$ in channel space, the compression scheme can be visualized with a confusion matrix showing the similarity of feature values. We employ a symmetric measure of the similarity between probability distributions, the Bhattacharyya coefficient (Kailath, 1967), which is straightforward to

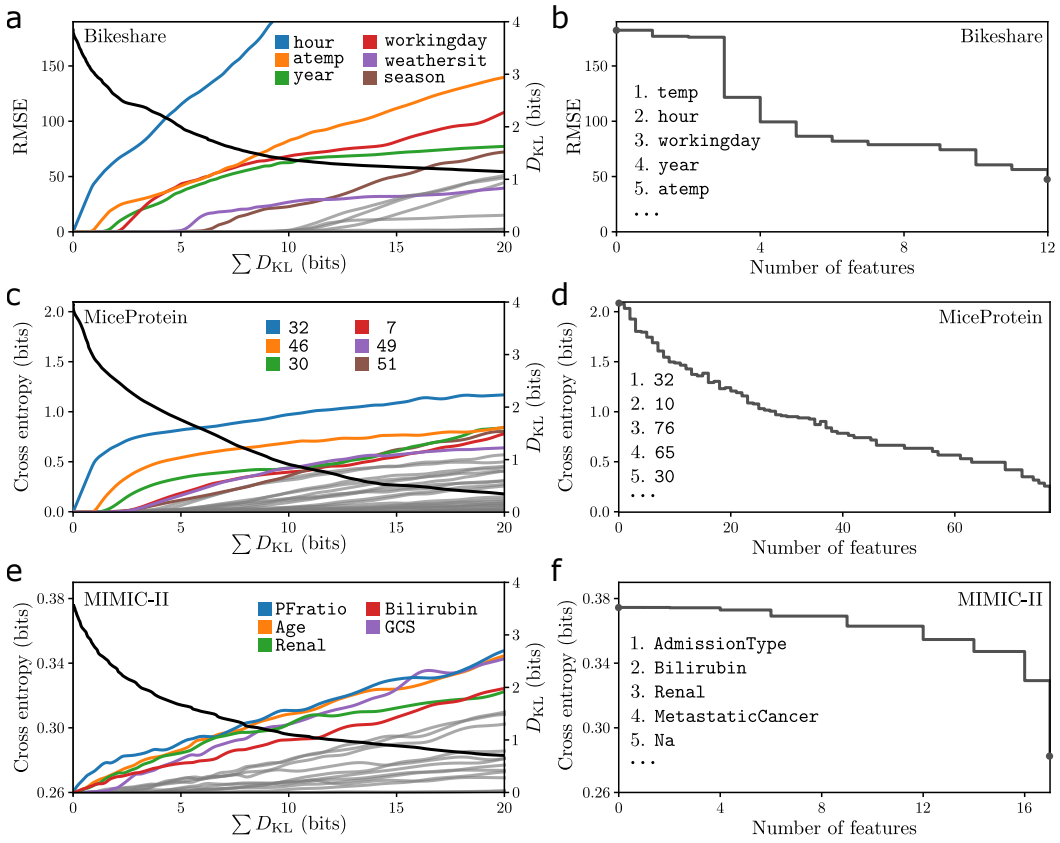

Figure 4: **Quasi-information plane trajectories and information allocation by feature. (a)** For the Bikeshare dataset, the optimal RMSE error versus $\sum D_{\mathrm{KL}}$ (black) decays as more information is used by the model about the features. The shifting allocation of information to the features tells of their relevance to the prediction at each level of approximation. Only a few bits about a handful of features is enough for most of the decrease in error. We label and color the first features to contribute non-negligibly to the approximations. **(b)** The LassoNet trajectory corresponding to **(a)** yields a much sparser feature selection signal: the error per number of input features, and an order in which the features were added. Note the LassoNet trajectory extends to the point of all features included, whereas the Distributed IB is truncated at 20 bits of $\sum D_{\mathrm{KL}}$. **(c, d)** Same as **(a, b)** for the MiceProtein dataset, with expression levels of 77 proteins measured to classify mice in one of eight categories related to Down syndrome. **(e, f)** Same as **(a, b)** for the MIMIC-II dataset, a binary classification task related to ICU mortality, with 17 input features.

compute when both are multivariate normal distributions. The Bhattacharyya coefficient is 0 when the distributions are perfectly disjoint (i.e., zero probability is assigned by $p(x)$ everywhere that $q(x)$ has nonzero probability, and vice versa), and 1 when the distributions are identical. In other words, if the Bhattacharyya coefficient between $p(u|x_1 = a)$ and $p(u|x_1 = b)$ is 1, the feature values are indistinguishable to the rest of the network and the output is independent of whether the actual feature value was a or b. On the other extreme, if the Bhattacharyya coefficient is 0, it is worth the information penalty to discern between the feature values.

The learned compression of a feature is therefore a clustering of the feature values in regard to their similarity in predicting the output. Without restricting the complexity of the machine learning model, we cannot know exactly how the feature values contribute to a prediction. Assessing the similarity of feature values with respect to the rest of the network is the best that we can do. The spectrum of learned models may be seen as ranging from all inputs perfectly distinguishable ($\beta \to 0$, unconstrained machine learning) to all inputs perfectly indistinguishable (the model utilizes zero information about the input).

Table 1: Performance on various tabular datasets. NAM and NODE-GAM allow no feature interactions; NODE-GA^{2}M contain pairwise feature interactions; the Neural Oblivious Decision Ensemble (NODE) LassoNet, and Distributed IB methods are full complexity models (allowing arbitrary feature interactions). ‡Results from Chang et al. (2022).

	Bikeshare (RMSE ↓)	Wine (RMSE ↓)	MIMIC-II (AUC % ↑)	MIMIC-III (AUC % ↑)	Microsoft (MSE ↓)
NAM (Agarwal et al., 2021)	99.9$\pm$1.3	0.71$\pm$0.02	83.0$\pm$0.8	81.7$\pm$0.2	0.5908$\pm$0.0004
NODE-GAM (Chang et al., 2022)	100.7$\pm$1.6	0.71$\pm$0.03	83.2$\pm$1.1	81.4$\pm$0.5	0.5821$\pm$0.0004
NODE-GA^{2}M (Chang et al., 2022)	49.8$\pm$0.8	0.67$\pm$0.02	84.6$\pm$1.1	82.2$\pm$0.7	0.5618$\pm$0.0003
LassoNet (Lemhadri et al., 2021)	46.0$\pm$2.0	0.68$\pm$0.02	83.4$\pm$0.8	79.4$\pm$0.8	0.5667$\pm$0.0012
NODE‡ (Popov et al., 2019)	**36.2**$\pm$1.9	0.64$\pm$0.01	84.3$\pm$1.1	**82.8**$\pm$0.7	**0.5570**$\pm$0.0002
Distributed IB (Ours)	38.0$\pm$1.1	**0.62**$\pm$0.01	**84.9**$\pm$1.2	80.2$\pm$0.4	0.5790$\pm$0.0003

We show in Fig. 3 example confusion matrices corresponding to the compression of four values with different amounts of information. The compression can be a hard clustering of the feature's values into two equally-probable sets where x_a and x_b are indistinguishable, and similarly for x_c and x_d (Fig. 3). Another possible compression with one bit of information is shown in Fig. 3c, where each feature value is 'confused' with half of the other feature values (a soft clustering). We find the latter to be the case found by the Distributed IB more often than not, meaning the learned compressions cannot be represented by a simple decision tree; rather, the confusion matrices are the fundamental distillations of the information preserved about each feature.

Boosting expressivity for low-dimensional features. Similarly with the authors of Agarwal et al. (2021), we found the feature encoders struggled with expressivity due to the low dimensionality of the features. We *positionally encode* feature values (Tancik et al., 2020) before they pass to the neural network, taking each value z to $[\sin(\omega_1 z), \sin(\omega_2 z), ...]$ where ω_k are the frequencies for the encoding.

4 Experiments

We analyze a variety of tabular datasets with the Distributed IB and compare the interpretability signal to what is obtainable through other methods. Our emphasis is on the unique capabilities of the Distributed IB to produce insight about the relationship in the data modelled by the neural network. We include performance results in Tab. 1 for completeness. Note that the Distributed IB places no constraints on the machine learning models used and in the $\beta \to 0$ regime could hypothetically be as good as any finely tuned black-box model. The freedom from complexity constraints is particularly clear in the smaller regression datasets, whereas performance is only middling on several other datasets (refer to App. B for implementation details).

4.1 Increasing sparsity to extract informative features

We first analyze the Distributed IB's capacity to extract relevant features for three datasets in Fig. 4. Bikeshare (Dua & Graff, 2017) is a regression dataset to predict the number of bikes rented given 16 descriptors of the date, time, and weather; MiceProtein (Higuera et al., 2015) is a classification dataset relating the expression of 77 genes to normal and trisomic mice exposed to various experimental conditions; MIMIC-II (Johnson et al., 2016) is a binary classification dataset to predict hospital ICU mortality given 17 patient measurements. We compare to LassoNet (Lemhadri et al., 2021), a feature selection method that attaches an L1 regularization penalty to each feature and enforces a constraint on weights in the first layer. By gradually increasing the strength of the overall L1 regularization, LassoNet finds a spectrum of models that utilize fewer features while maximizing predictive accuracy about the output variable, providing a signal about what features and how many are necessary for different levels of accuracy.

The Distributed IB also gradually enforces sparsity to probe the tradeoff between information about the features and predictive information, though with important differences. LassoNet can only vary the information used for prediction through a discrete-valued proxy: the number of active features.

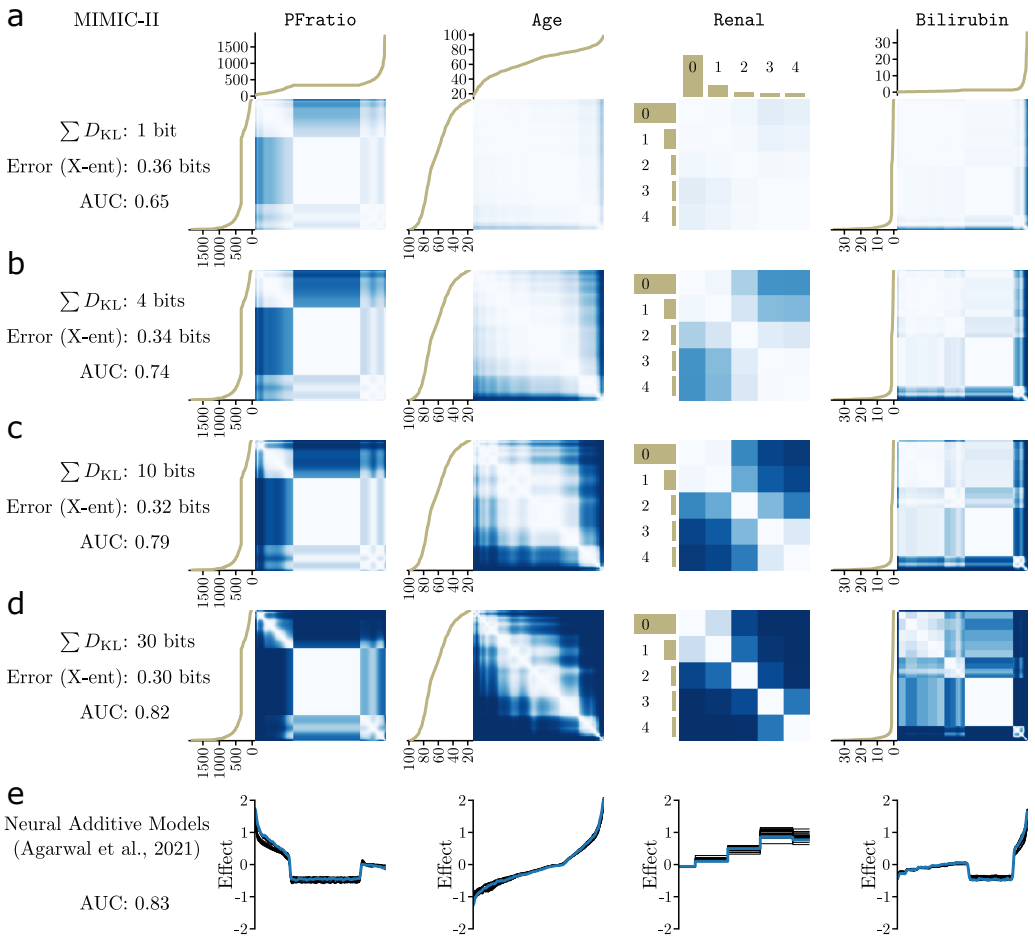

Figure 5: **Feature effects via inspection of the optimal compression schemes. (a-d)** For the MIMIC-II dataset, we display distinguishability matrices at different levels of approximation for four features. To the left (top) of the distinguishability matrices are the feature values corresponding to the row (column), which have been sorted. The features (`PFratio`, `Age`, `Renal`, and `Bilirubin`) were the first four to contribute to the approximations in the low $\sum D_{\mathrm{KL}}$ regime of Fig. 4. For the categorical variable `Renal` the barchart displays the frequency of the particular values. As $\sum D_{\mathrm{KL}}$ increases, the cross entropy error decreases and the ROC AUC (area under curve) increases. **(e)** The effects of these four features can be directly visualized with NAM; the horizontal coordinates match those of the matrices in **(a-d)**.

Binary inclusion/exclusion of features is a common limitation for feature selection methods (Bang et al., 2021; Kolek et al., 2022)). By contrast, the Distributed IB compresses each feature along a continuum, allowing for rich signal of interpretability.

In Fig. 4a we find that only a few bits of information (loosely defined through $\sum D_{\mathrm{KL}}$), mostly coming from the `hour` feature, is enough to get most of the performance of the full-information model. Compared to the discrete spectrum of LassoNet (Fig. 4b), the Distributed IB quantifies how much information is required—total, and from each feature—for a given level of predictability. Interestingly, while many of the same features are prioritized by both methods, the performance for LassoNet only improves appreciably with three features, suggesting a higher order interaction. The Distributed IB finds approximate models that are predictive almost immediately, suggesting only a little information about each of the features is necessary to account for the interaction.

In contrast to Bikeshare, where decent approximations required only a handful of features, the Mice-Protein and MIMIC-II datasets were found to require information from many features simultaneously.

The information allocation is dynamic, with some features saturating quickly, and others growing in information usage at different rates. The signal obtained by the Distributed IB can help decompose through datasets with a large number of features and complex interaction effects.

4.2 Fine-grained inspection of feature effects

We now analyze the signal from a Distributed IB optimization at the level of particular feature values. One of the primary strengths of a GAM is the effects of the feature values are straightforwardly obtained, though at the expense of interactions between the features. By contrast, the Distributed IB permits arbitrarily complex feature interactions, and still maintains a window into the role specific feature values play toward the ultimate prediction. In Fig. 5 we compare the confusion matrices found by the Distributed IB to the feature effects obtainable with NAM (Agarwal et al., 2021).

It is important to note that the Distributed IB obtains this level of granularity for the entire spectrum of learned models, allowing one to build up toward the maximally predictive model by a series of approximations that gradually incorporate more information from the input features. As an example, we display in Fig. 5 a series of approximations to the relationship between patient descriptors and eventual outcome in the MIMIC-II dataset. To a first approximation (Fig. 5a), with only about one bit of information about all features, the scant distinguishing power is allocated primarily to discerning low from high values of `PFratio`, and the largest values of `Bilirubin`. We are unable to say how this particular information about these features is used for prediction—the price of allowing the model to be more complex—but we can begin to infer from where the discerning power is allocated. At four bits, old age begins to be discerned from everyone under around 80 years old, and the `Renal` categories are partitioned. At ten and thirty bits of information about the inputs, certain distinctions have become hard, and others (such as in the `Age` feature) retain an elaborate structure of similarity.

The feature effects are directly retrievable with GAM-based methods such as NAM (Fig. 5e). The feature effects found by the additive model largely corroborate the distinguishability found by the Distributed IB: where the effect is similar between feature values, distinguishing between the feature values is less beneficial to the loss than when the feature effect is different. Such distinctions show up at higher values of $\sum D_{\mathrm{KL}}$.

5 Discussion

The Distributed IB provides a rich signal about where information resides in a relationship, with regards to features of the input. It provides a novel source of interpretability, particularly useful in scenarios where obtaining a foothold is difficult due to large numbers of features and complex interaction effects. The capacity for successive approximation, allowing detail to be incorporated gradually so that humans' limited cognitive capacity (Cowan, 2010; Rudin et al., 2022) does not prevent comprehension of the full relationship, is one of the primary factors in the Distributed IB's interpretability.

The signal from the Distributed IB—particularly the compression schemes of feature values (e.g. Fig. 5)—is vulnerable to overinterpretation. This is partly due to a difficulty surrounding intuition about compression schemes (i.e., it is perhaps most natural to imagine 1 bit of information as a single yes/no question, or a hard clustering (Fig. 3b), rather than as a confusion matrix (Fig. 3c)), and partly due to the inability to make claims like "changing input A to input B changes output C to output D". Rather than supplanting other interpretability methods, the Distributed IB should be seen as a supplement. A thorough analysis should incorporate methods where the effects are comprehensible, possibly using a reduced feature set found by the Distributed IB.

References

Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40 (12):2897–2905, 2018. 4

Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. *Advances in Neural Information Processing Systems*, 34:4699–4711, 2021. 3, 7, 9, 15

Iñaki Estella Aguerri and Abdellatif Zaidi. Distributed variational representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):120–138, 2021. doi: 10.1109/TPAMI.2019.2928806. 3, 4

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016. 3, 4

Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *International conference on machine learning*, pp. 1247–1255. PMLR, 2013. 3

Shahab Asoodeh and Flavio P. Calmon. Bottleneck problems: An information and estimation-theoretic view. *Entropy*, 22(11):1325, 2020. 1, 3

Muhammed Fatih Balın, Abubakar Abid, and James Zou. Concrete autoencoders: Differentiable feature selection and reconstruction. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 444–453. PMLR, 09–15 Jun 2019. 2

Seojin Bang, Pengtao Xie, Heewook Lee, Wei Wu, and Eric Xing. Explaining a black-box by using a deep variational information bottleneck approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11396–11404, 2021. 3, 8

Toby Berger, Zhen Zhang, and Harish Viswanathan. The CEO problem [multiterminal source coding]. *IEEE Transactions on Information Theory*, 42(3):887–902, 1996. 3

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 3

Jie Cai, Jiawei Luo, Shulin Wang, and Sheng Yang. Feature selection in machine learning: A new perspective. *Neurocomputing*, 300:70–79, 2018. 2

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noémie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1721–1730, 2015. 1

Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014. 2

Chun-Hao Chang, Sarah Tan, Ben Lengerich, Anna Goldenberg, and Rich Caruana. How interpretable and trustworthy are gams? In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 95–105, 2021. 3

Chun-Hao Chang, Rich Caruana, and Anna Goldenberg. Node-gam: Neural generalized additive model for interpretable deep learning. In *International Conference on Learning Representations*, 2022. 3, 7, 13, 14

Nelson Cowan. The magical mystery four: How is working memory capacity limited, and why? *Current Directions in Psychological Science*, 19(1):51–57, 2010. 9

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017. 1

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml. 7, 13

Iñaki Estella Aguerri and Abdellatif Zaidi. Distributed information bottleneck method for discrete and gaussian sources. In *International Zurich Seminar on Information and Communication (IZS 2018). Proceedings*, pp. 35–39. ETH Zurich, 2018. 2, 3, 4

Feng-Lei Fan, Jinjun Xiong, Mengzhou Li, and Ge Wang. On interpretability of artificial neural networks: A survey. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 5(6):741–760, 2021. 1

Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *J. Mach. Learn. Res.*, 20(177):1–81, 2019. 3

Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7 (2):179–188, 1936. 3

Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pp. 3429–3437, 2017. 2

David R. Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004. 3

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-vae: Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations (ICLR)*, 2017. 2, 4

Clara Higuera, Katheleen J Gardiner, and Krzysztof J Cios. Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome. *PloS one*, 10(6):e0129126, 2015. 7

Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3-4):321–377, 12 1936. ISSN 0006-3444. doi: 10.1093/biomet/28.3-4.321. URL https://doi.org/10.1093/biomet/28.3-4.321. 2

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016. 7, 13

Thomas Kailath. The divergence and bhattacharyya distance measures in signal selection. *IEEE transactions on communication technology*, 15(1):52–60, 1967. 5

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2014. 4

Artemy Kolchinsky, Brendan D. Tracey, and David H. Wolpert. Nonlinear information bottleneck. *Entropy*, 21(12), 2019. ISSN 1099-4300. doi: 10.3390/e21121181. URL https://www.mdpi.com/1099-4300/21/12/1181. 4

Stefan Kolek, Duc Anh Nguyen, Ron Levie, Joan Bruna, and Gitta Kutyniok. A rate-distortion framework for explaining black-box model decisions. In *International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers*, pp. 91–115. Springer, 2022. 2, 8

Ismael Lemhadri, Feng Ruan, Louis Abraham, and Robert Tibshirani. Lassonet: A neural network with feature sparsity. *Journal of Machine Learning Research*, 22(127):1–29, 2021. URL http://jmlr.org/papers/v22/20-848.html. 2, 3, 5, 7, 14

Kaspar Märtens and Christopher Yau. Neural decomposition: Functional ANOVA with variational autoencoders. In *International Conference on Artificial Intelligence and Statistics*, pp. 2917–2927. PMLR, 2020. 3

David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. In *International Conference on Artificial Intelligence and Statistics*, pp. 875–884. PMLR, 2020. 4

Christoph Molnar. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. 2022. 1, 3

Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. Interpretable machine learning–a brief history, state-of-the-art and challenges. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 417–431. Springer, 2020. 1

Kieran A Murphy and Dani S Bassett. The distributed information bottleneck reveals the explanatory structure of complex systems. *arXiv preprint arXiv:2204.07576*, 2022. 2, 3

Andrew Y. Ng. Feature selection, L1 vs. L2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 78, 2004. 3

Amichai Painsky, Meir Feder, and Naftali Tishby. An information-theoretic framework for non-linear canonical correlation analysis. *CoRR*, 2018. 3

Sergei Popov, Stanislav Morozov, and Artem Babenko. Neural oblivious decision ensembles for deep learning on tabular data. *arXiv preprint arXiv:1909.06312*, 2019. 7

Tao Qin and Tie-Yan Liu. Introducing LETOR 4.0 datasets. *CoRR*, abs/1306.2597, 2013. URL http://arxiv.org/abs/1306.2597. 13

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

Cynthia Rudin, Caroline Wang, and Beau Coker. The age of secrecy and unfairness in recidivism prediction. *arXiv preprint arXiv:1811.00731*, 2018. 1

Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16:1–85, 2022. 1, 9

Andrew M. Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D. Tracey, and David D. Cox. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124020, dec 2019. doi: 10.1088/1742-5468/ab3985. URL https://doi.org/10.1088%2F1742-5468%2Fab3985. 4

Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948. 3

Steffen Steiner and Volker Kuehn. Distributed compression using the information bottleneck principle. In *ICC 2021 - IEEE International Conference on Communications*, pp. 1–6, 2021. doi: 10.1109/ICC42927.2021.9500324. 3

Charles J. Stone. The use of polynomial splines and their tensor products in multivariate function estimation. *The Annals of Statistics*, 22(1):118–171, 1994. 3

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020. 7

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pp. 1–5. IEEE, 2015. 5

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000. 1, 3

Qi Wang, Claire Boudreau, Qixing Luo, Pang-Ning Tan, and Jiayu Zhou. Deep multi-view information bottleneck. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pp. 37–45. SIAM, 2019. 3

Weiran Wang, Xinchen Yan, Honglak Lee, and Karen Livescu. Deep variational canonical correlation analysis. *arXiv preprint arXiv:1610.03454*, 2016. 3

Tailin Wu, Ian Fischer, Isaac L. Chuang, and Max Tegmark. Learnability for the information bottleneck. In *Uncertainty in Artificial Intelligence*, pp. 1050–1060. PMLR, 2020. 5

Zebin Yang, Aijun Zhang, and Agus Sudjianto. Gami-net: An explainable neural network based on generalized additive models with structured interactions. *Pattern Recognition*, 120:108192, 2021. 3

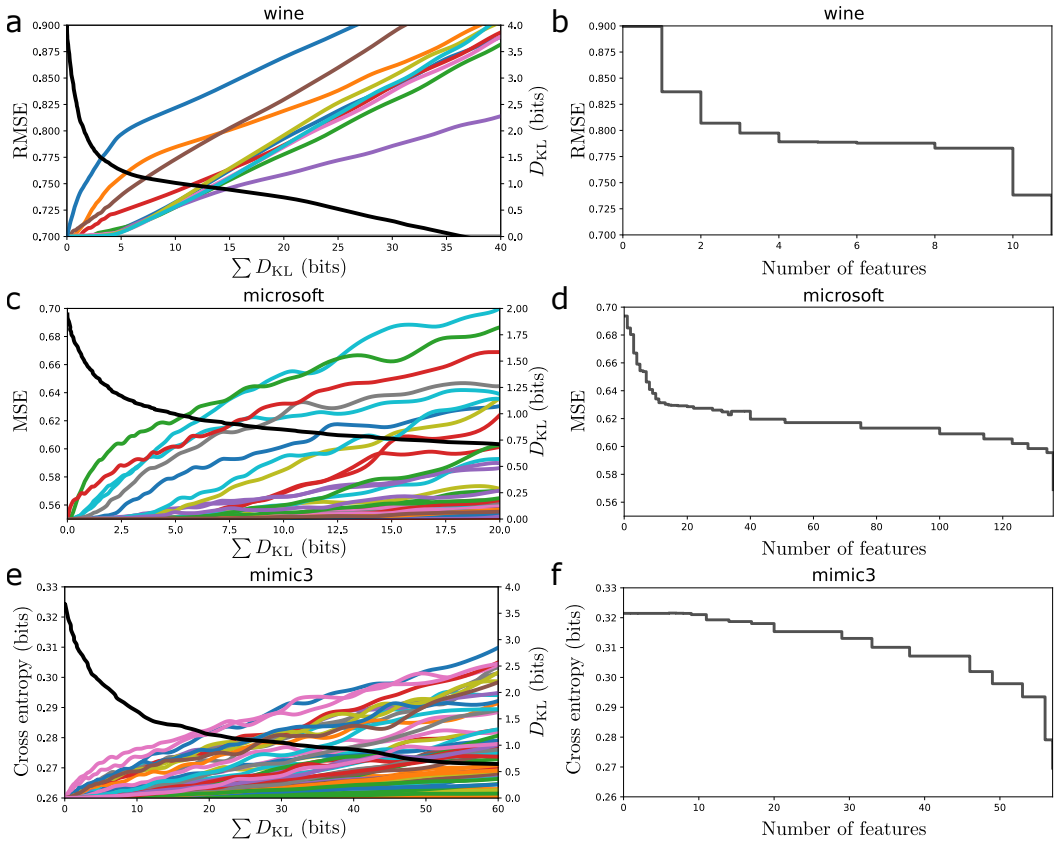

Figure 6: **Information plane optimizations for the remaining three datasets.** We display Distributed IB optimizations alongside LassoNet results. Note the $\sum D_{\mathrm{KL}}$ is not uniform across datasets. **(a, b)** Wine, regress wine quality with 12 features (Dua & Graff, 2017); **(c, d)** Microsoft Learn to Rank, 136 features (Qin & Liu, 2013); **(e, f)** MIMIC-III, similar to MIMIC-II but with 57 features (Johnson et al., 2016).

A Extended Visualizations

We show in Fig. 6 Distributed IB optimizations for the remaining three datasets mentioned in the manuscript, alongside their LassoNet counterparts.

B Implementation Details

All Distributed IB experiments were run on a single computer with a 12 GB GeForce RTX 3060 GPU; 100k steps on the Bikeshare dataset takes a couple minutes, and on Microsoft about 2 hours.

Tensorflow code is attached to this submission and will be posted (in extended form) to Github upon acceptance. The data loading and set up is heavily based on the code from NODE-GAM (Chang et al., 2022). Running the following at the command line should train with Distributed IB on the Bikeshare dataset for 5×10^4 steps, saving distinguishability matrices along the way:

```
> python train.py -dataset bikeshare -fold 0 -name bikes0
-number_annealing_steps 50000 -save_distinguishability_matrices
True
```

Training hyperparameters and architecture details are shown in Tab. 2 (common to all datasets) and Tab. 3 (dataset-specific). In the first stage of hyperparameter tuning, we explored parameter space in an unstructured manner and found that most parameters affected all datasets similarly. After

Parameter	Value
Positional encoding frequencies	[1, 2, 4, 8]
Nonlinear activation	Leaky ReLU ($\alpha = 0.2$)
Feature encoder MLP (one per feature)	[128, 128]
Bottleneck embedding space dimension	8
Joint encoder MLP	[256, 256]
Batch size	128
Optimizer	Adam
Learning rate	3×10^{-4}
β_{initial}	2×10^{-5}
β_{final}	2

Table 2: Training parameters for the Distributed IB experiments.

Parameter	Dataset	Value
Annealing steps	Bikeshare	10^5
	Wine	10^5
	MIMIC 2	5×10^4
	MIMIC 3	5×10^4
	Mice protein	5×10^4
	Microsoft	10^5
Dropout fraction	Bikeshare	0
	Wine	0.1
	MIMIC 2	0.3
	MIMIC 3	0.5
	Mice protein	0.1
	Microsoft	0.1

Table 3: Dataset-specific parameters for the Distributed IB experiments.

some tuning, we settled on the values listed in Tab. 2. We found the effects of dropout rate and the number of annealing steps to differ by dataset, and selected optimal values out of $\{0, 0.1, 0.3, 0.5\}$ and $\{2.5 \times 10^4, 5 \times 10^4, 10^5\}$, respectively (Tab. 3).

B.1 Datasets

We used the dataset preprocessing code released with NODE-GAM (Chang et al., 2022) on Github[1] with a minor modification to use one-hot encoding[2] for categorical variables instead of leave-one-out encoding[3] when input to the Distributed IB models, unless there were more than 100 categories.

B.2 Comparison methods

LassoNet (Lemhadri et al., 2021) We used the publicly released Pytorch code[4] with only minor modifications: (i) fixed a typo in `interfaces.py`[5] (delete '='), (ii) changed the objective outputs

[1] https://github.com/zzzace2000/nodegam
[2] https://contrib.scikit-learn.org/category_encoders/onehot.html
[3] https://contrib.scikit-learn.org/category_encoders/leaveoneout.html
[4] https://github.com/lasso-net/lassonet
[5] https://github.com/lasso-net/lassonet/blob/54d05048eca51c7b57077f0e09d4a72767f3e567/lassonet/interfaces.py#L470

during training[6] to be solely the error metric (excluding the regularization terms) for the plots of Fig. 4.

We used the `LassoNetClassifier` and `LassoNetRegressor` classes, instead of the cross-validation versions that were released after the original publication, and performed our own cross-validation with the same splits for the other methods. For each dataset we tuned the following hyperparameters with a grid search: L2-regularization weight $\gamma \in \{0, 0.01, 0.1, 1\}$ and the dropout fraction in $\{0, 0.1, 0.3, 0.5\}$. We selected the parameters with the best integrated performance on the validation set (metric $\times$ number of features), and used the same hidden dimensions as the joint encoder for the Distributed IB method (2 layers of 256 units).

Neural additive models (Agarwal et al., 2021) We used the publicly released Tensorflow code[7]. We modified the size of the feature encoders: we encountered memory issues with the default feature encoder sizes of [M, 64, 32], with M a function of the number of unique values in the training set for that feature[8]. Instead we made each feature encoder size [256, 256, 256] for the datasets with under 20 features, and [128, 128, 128] for the rest.

For each dataset we tuned the following hyperparameters with a grid search: hidden activation in $\{\texttt{relu}, \texttt{exu}\}$, dropout in $\{0, 0.1, 0.3\}$, weight decay (L2 regularization) in $\{0, 10^{-5}, 10^{-4}\}$.

[6] https://github.com/lasso-net/lassonet/blob/54d05048eca51c7b57077f0e09d4a72767f3e567/lassonet/interfaces.py#L251

[7] https://github.com/google-research/google-research/tree/master/neural_additive_models

[8] https://github.com/google-research/google-research/blob/e79b8f09344935a2ab22660293233eb032df13a1/neural_additive_models/graph_builder.py#L283