A Functional Information Perspective on Model Interpretation

Itai Gat [1] **Nitay Calderon** [1] **Roi Reichart** [1] **Tamir Hazan** [1]

Abstract

Contemporary predictive models are hard to interpret as their deep nets exploit numerous complex relations between input elements. This work suggests a theoretical framework for model interpretability by measuring the contribution of relevant features to the functional entropy of the network with respect to the input. We rely on the log-Sobolev inequality that bounds the functional entropy by the functional Fisher information with respect to the covariance of the data. This provides a principled way to measure the amount of information contribution of a subset of features to the decision function. Through extensive experiments, we show that our method surpasses existing interpretability sampling-based methods on various data signals such as image, text, and audio.

1. Introduction

Machine learning is ubiquitous these days, and its impact on everyday life is substantial. Supervised learning algorithms are instrumental to autonomous driving (Lavin & Gray, 2016; Bojarski et al., 2016; Luss et al., 2019), serving people with disabilities (Tadmor et al., 2016), improve hearing aids (Clauset & Post, 2019; Fedorov et al., 2020; Green et al., 2022), and are being extensively used in medical diagnosis (Deo, 2015; Ophir et al., 2020; Zhou et al., 2021). Unfortunately, since the models that are achieving these advancements are complex, their decisions are usually not well-understood by their operators. Consequently, model interpretability is becoming an important goal in contemporary deep nets research. To facilitate interpretability, in this work, we provide a theoretical framework that allows measuring the information of a decision function by considering the expected ratio between its gradient norm and its value.

Gradient-based approaches are widely used to interpret the model predictions since they bring forth an insight into the internal mechanism of the model (Simonyan et al., 2013; Gu et al., 2018; Sundararajan et al., 2017; Shrikumar et al., 2017; Gat et al., 2021; Shrikumar et al., 2016; Springenberg et al., 2014). These methods produce different explanation maps using the gradient of a class-related output with respect to its input data. While gradient-based visualizations excel at providing class-specific explanations, these explanations could be noisy in part due to local variations in partial derivatives.

In order to overcome such variations, Smilkov et al. (2017); Adebayo et al. (2018) propose to compute the expected output of gradient-based methods with respect to their input. This practice is known as the sampling approach for interpretability methods. Existing sampling-based methods rely on the assumption that the features are uncorrelated, while real-world data such as pixels in images and words in texts are in fact correlated.

We provide a theoretical framework that applies functional entropy as a guiding concept to the amount of information a given deep net holds for a given input with respect to any of the possible labels. The functional entropy (Bakry et al., 2013), originated from functional analysis, measures the possibility of change that is encapsulated in any decision function with respect to its input. The functional entropy contrasts the well-known Shannon's entropy that measures the capacity of probability distributions. We relate the functional entropy to the functional Fisher information that measures the amount of expansion in the decision function through its gradient norm with respect to its distribution. The functional Fisher information is defined for non-negative functions in contrast to the Fisher information, which is defined over probability density functions.

With these mathematical concepts, we are able to construct a sampling framework for gradient-based explanation methods. The connection between functional entropy and functional Fisher information signifies the importance of the covariance encapsulated in the data. It also allows us to seamlessly extract information of a subset of features from their correlated distribution.

The remainder of the work is organized as follows: In Sec. 3 we present the notions of functional entropy and functional Fisher information. Next, in Sec. 4, we present the role of

[1]Technion - Israel Institute of Technology. Correspondence to: Itai Gat <itaigat@technion.ac.il>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

the data covariance matrix when relating these two functional analysis operators. We use the data correlations to extract the relevant importance of a subset of the input. Lastly, in Sec. 5, we demonstrate the effectiveness of our approach on diverse data modalities: vision, text, and audio. We study our approach both quantitatively and qualitatively. We visualize our method's scores in the qualitative experiments and compare them to current explainability sampling-based methods with negative perturbations in the quantitative experiments. Our extensive experiments show that, compared to current methods, our method can better identify the features that drive the predictions of the model and quantitatively estimate their importance. In addition, we compare text explanations of sampling-based methods for different types of architectures. This analysis sheds light on the differences between the architectures and, in particular, highlights the difficulties in interpreting transformer-based models.

In summary, our contributions in this work are: (1) We propose a mathematical framework that measures the amount of information in deep nets. Our framework provides information-theoretical grounds for feature attribution sampling-based methods (Theorem 4.1). Our method uses the functional Fisher information and suggests that the covariance of the data should be considered. (2) We present a novel approach for sampling explanations of a subset of features (Theorem 4.2). According to our method, one should use dependent conditional sampling for explanations of feature subsets. (3) We conduct extensive experiments to evaluate our method on various data modalities, quantitatively and qualitatively. Our code is available at `https://github.com/nitaytech/FunctionalExplanation`.

2. Related Work

Over the years, many gradient-based methods were developed to interpret decisions of deep nets. An explanation of a decision function f_y should associate features with scores reflecting their impact on the decision. Intuitively, the gradients of a decision function with respect to the features, $\nabla f_y(x)$ represent the extent to which perturbations in each feature would change the value of the function, i.e., the impact of the feature x_i on to the output y is then the partial gradient $\nabla f_y(x)_i$. Gradient-based methods are considered white-box since they allow to inspect the internal mechanism of the decision function. In contrast, black-box methods like LIME (Ribeiro et al., 2016) are not based on gradients but instead interpret the model's prediction for a given data point based on a local linear approximation around this input. Another important black-box method is SHapleyAdditive exPlanations (SHAP) (Lundberg & Lee, 2017), introducing a unified game-theoretic framework for attribution methods based on Shapley values (Shapley, 1953).

Gradient-based methods. Early gradient-based methods include Saliency maps (Simonyan et al., 2013), Grad-CAM (Gu et al., 2018), and Guided backpropagation (Springenberg et al., 2014). These methods use the values of the deep net gradients with respect to the features or the latent space. Interestingly, Nie et al. (2018) show that while the visualizations produced by the guided backpropagation method are impressive, it does not explain well the model predictions.

Gradient-based methods typically do not account for potential correlations between the features since they estimate the impact of a feature on the prediction using the partial derivative with respect to this feature. A number of methods address this challenge, for example, Gradient$\times$Input (Shrikumar et al., 2016) and Integrated Gradients (Sundararajan et al., 2017) multiply the partial derivatives by the input itself. In contrast, we consider the covariance of the data in order to account for potential correlations between the features explicitly. Notably, the integration of the covariance into our score is naturally derived from the functional Fisher information component of our framework.

Sampling-based gradient methods. There is no guarantee that the gradients of f_y vary smoothly, and they may fluctuate sharply at small scales (Smilkov et al., 2017). Therefore, explanations based on raw gradients are noisy and might emphasize meaningless local variations in the partial derivatives. To overcome this, Smilkov et al. (2017) proposes SmoothGrad. This method computes $\mathbb{E}_{z\sim\nu}[\nabla f_y(z)_i]$, which is estimated by calculating the average gradients of mutually independent Gaussians ν centered around the input x. Since gradients have signed values, their interpretation is ambiguous. Therefore, Smilkov et al. (2017) proposed the SmoothGradSQ method, which considering the absolute or squared values of the gradients: $\mathbb{E}_{z\sim\nu}[\nabla f_y(z)_i^2]$. Later, Adebayo et al. (2018) proposed VarGrad, estimating the importance of a feature by computing the variance of its partial derivation under Gaussian perturbations. These methods require sampling to compute the expectation, a property that our method shares.

Although gradient-based methods are intuitive, and despite the good visualizations they provide, there is no theoretical framework that explains how they quantify the impact of the features on the decision function. In this work, we introduce such a theoretical framework. Our framework applies the functional entropy as a guiding concept for computing the contribution of each feature to the decision function. Since computing the functional entropy is intractable, we turn to approximation with the functional Fisher information. Our mathematical framework explicitly accounts for correlations between subsets of the data in order to provide

a more reliable estimate of the expectations considered by an explanation method.

Information in deep nets explanation. Information theory aspects have been studied in deep neural networks. A well-known line of work focuses on the information bottleneck criterion (Tishby & Zaslavsky, 2015; Shwartz-Ziv & Tishby, 2017). This criterion is designed to maximize the mutual information of the latent representation of the input and with the label. Learning to explain approach (L2X, (Chen et al., 2018)) takes a different view and uses the mutual information between the input features and each of the labels to select the most informative features for the prediction. These works require to model the joint distribution of features and labels.

Our work uses a functional analysis perspective about information and measures the amount of information in the decision function. Our functional analysis view allows us to avoid modeling the joint probability distributions of features and labels. Recent work by Gat et al. (2020) propose to maximize functional entropy of multi-modal data in order to remove modal-specific biases and facilitate out-of-distribution generalization. Unlike our work, they aim to improve generalization, and their method does not consider the covariance of the input features.

3. Background

A discriminative learner constructs a mapping between a data instance $x \in \mathcal{X}$ and a label $y \in \mathcal{Y}$ given training data $S = \{(x_1, y_1), ..., (x_m, y_m)\}$. For example, in an object recognition task, x is an image that is composed of pixels. Each pixel consists of a three-dimensional vector that represents the color of the pixel. Generally, $x \in \mathbb{R}^d$ resides in the Euclidean space. Throughout the work, we consider classification tasks for which y is a discrete label.

A discriminative learning algorithm searches for parameters w to best fit the relation of (x_i, y_i) in the training data. A popular approach is to measure the goodness of fit using the negative log-likelihood loss: the parameters w of a conditional probability model $p_w(y|x)$ are learned while minimizing $-\log p_w(y_i|x_i)$ over the training data. At test time, the class predicted for a test instance x is the one with the highest probability, namely $\arg\max_y p_w(y|x)$.

The learned probability model assigns a probability for each possible class. In this work, we interpret how the input x relates to each of the possible classes. We suggest to measure the amount of functional information in the data instance x that relates to each of the possible labels y. For each class, we relate the learner's preference to a different non-negative function: $f_y(x) = p_w(y|x)$. In the following, we use tools developed in functional analysis (cf. Bakry et al., 2013). First, we present the notion of the functional entropy. Then, we connect the functional entropy to the functional Fisher information through the log-Sobolev inequality.

3.1. Functional Entropy

Functional entropies (Bakry et al., 2013) are defined over a continuous random variable: a function $f_y(z)$ over the Euclidean space $z \in \mathbb{R}^d$ with a Gaussian probability measure $\mu = \mathcal{N}(x, \Sigma)$ whose probability density functions is

$$d\mu(z) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} e^{-\frac{1}{2}((z-x)^\top \Sigma^{-1}(z-x))} dz. \quad (1)$$

For notational clarity we denote the standard Gaussian measure centered around x by $\nu = \mathcal{N}(x, I)$. It defers from $\mu = \mathcal{N}(x, \Sigma)$ by its covariance matrix. We aim at measuring the functional entropy of a label y and a data instance x. Here and throughout, we use z to refer to a stochastic variable, which we integrate over. The functional entropy of the non-negative label function $f_y(z) \geq 0$ is

$$\mathrm{Ent}_\mu(f_y) \triangleq \int_{\mathbb{R}^d} f_y(z) \log \frac{f_y(z)}{\int_{\mathbb{R}^d} f_y(z) d\mu(z)} d\mu(z) \quad (2)$$

We hence define the functional entropy of a deep net with respect to a label y by the function softmax output $f_y(z)$ when $z \sim \mu$ is sampled from a Gaussian distribution around x. The functional entropy is non-negative, namely $\mathrm{Ent}_\mu(f_y) \geq 0$ and equals zero only if $f_y(z)$ is a constant. This is in contrast to differential entropy of a continuous random variable with probability density function $q(z)$: $h(q) = -\int_{\mathbb{R}^d} q(z) \log q(z) dz$, which is defined for $q(z) \geq 0$ with $\int_{\mathbb{R}^d} q(z) dz = 1$ and may be negative.

The functional entropy can be thought of as the Kullback–Leibler (KL) divergence between the prior distribution $p_\nu(z)$ and the posterior distribution $q_\nu(z) \triangleq p_\nu(z) f_y(z)$ of the decision function $f_y(z) = p(y|z)$ with respect to the data generating distribution over z. For clarity, let's assume that $\int_{\mathbb{R}} p_\nu(z) f_y(z) dz = \mathbb{E}_{z\sim\nu}(f_y) = 1$. Then, we have:

$$\begin{aligned}
\mathrm{Ent}_\nu(f_y) &= \int_{\mathbb{R}} p_\nu(z) f_y(z) \log f_y(z) dz & (3)\\
&= \int_{\mathbb{R}} p_\nu(z) f_y(z) \log \frac{p_\nu(z) f_y(z)}{p_\nu(z)} dz & (4)\\
&= \int_{\mathbb{R}} q_\nu(z) \log \frac{q_\nu(z)}{p_\nu(z)} dz & (5)\\
&= \mathrm{KL}(q_\nu(z), p_\nu(z)). & (6)
\end{aligned}$$

Intuitively, a high KL divergence signifies highly important features, since the learning process forces the the posterior to diverge from the prior for such features.

Unfortunately, the functional entropy is hard to estimate empirically, since it involves the normalized term

$\frac{f_y(z)}{\int_{\mathbb{R}^n} f_y(z)\, d\mu(z)}$. Since the integral can only be estimated by sampling, the log-scale denominator of its estimate is hard to compute in practice. Next, we use the log-Sobolev inequality to overcome this.

3.2. Functional Fisher Information

Instead of estimating the functional entropy directly, we use the log-Sobolev inequality for standard Gaussian measures (cf. Bakry et al., 2013, Section 5.1.1). This permits to bound the functional entropy with the functional Fisher information. We denote the functional Fisher information with

$$\mathcal{I}_\nu(f_y) \triangleq \int_{\mathbb{R}^d} \frac{\|\nabla f_y(z)\|^2}{f_y(z)} d\nu(z) \tag{7}$$

Hereby, $\|\nabla f_y(z)\|$ is the ℓ_2-norm of the gradient of f_y. The functional Fisher information is a natural extension of the Fisher information, which is defined for probability density functions. Specifically, the log-Sobolev inequality for any non-negative function $f_y(z) \geq 0$ is,

$$\mathrm{Ent}_\nu(f_y) \leq \frac{1}{2}\mathcal{I}_\nu(f_y). \tag{8}$$

The above log-Sobolev inequality applies to standard Gaussian distribution $\nu = \mathcal{N}(x, I)$, centered around x. Furthermore, the bound is tight for the exponential family (see example in Appendix A.3), which is a desirable property since the entropy is computed with respect to a label y induced by a softmax output. The assumption that the elements the data are independent is limiting in practice, since the information in a data instance x with respect to an interpreted label y consider correlations between the different elements in $x \in \mathbb{R}^d$. In this work we rely on the functional Fisher information of correlated Gaussian distributions $\mu = \mathcal{N}(x, \Sigma)$ and show that taking into account the correlations in x improves the model's explanation of its prediction of y.

4. Feature Contribution via Functional Fisher Information

In this section, we propose a sampling-based method that can quantify the contribution of an input feature x_i to the decision function f_y. We start by describing our method, which samples perturbations of the input x from ν, and then expand it to overcome two key challenges. The first is the removal of the independence assumption encoded in ν, and the second is the computation of the contribution of a subset of features when conditioning on the others. We develop two theorems to address those challenges. We first introduce Theorem 4.1 which integrates the covariance matrix of y into the functional Fisher information. Then, Corollary 4.2 yields a sampling protocol suitable for features subsets. This allows us to generate sampling-based explanations from the conditional and marginal distributions.

Recall that the functional entropy is bounded by the functional Fisher information, $\mathcal{I}_\nu$. $\mathcal{I}_\nu$ measures the expansion of the decision function through the ℓ_2 norm of the gradient of that function. By considering Eq. (7), we obtain:

$$I_\nu(f_y) = \sum_i \mathbb{E}_{z\sim\nu}\left[\frac{(\nabla f_y(z)_i)^2}{f_y(z)}\right]. \tag{9}$$

Each component in Eq. (9) can be viewed as the contribution of the feature x_i to the total information.

4.1. Covariance

Real-world data features are correlated, and in fact the nature of the correlations changes across classes and modalities. For example, different features are expected to be correlated in images of lions and horses. Likewise, the correlations between words in positive movie reviews are expected to be different from those in negative reviews. As a result, sampling the stochastic variable z from a Gaussian distribution with a diagonal covariance matrix (i.e., encoding features independence), is incompatible with real-world data. To account for correlations within the data, we propose the following theoretical framework for explaining models applied to data with correlated features.

The functional Fisher information for dependent Gaussian measure is:

$$\mathcal{I}_\mu(f_y) \triangleq \int_{\mathbb{R}^d} \frac{\langle \Sigma\nabla f_y(z), \nabla f_y(z)\rangle}{f_y(z)} d\mu(z). \tag{10}$$

We next provide a theorem that extend Eq. (8) to the correlational case:

Theorem 4.1. *For every non-negative function $f_y : \mathbb{R}^d \to \mathbb{R}$ and a Gaussian measure μ,*

$$\mathrm{Ent}_\mu(f_y) \leq \frac{1}{2}\mathcal{I}_\mu(f_y). \tag{11}$$

Proof sketch. Relying on the log-Sobolev inequality, we perform the following variable change: $z \leftarrow \sqrt{\Sigma}(x + z)$. This permits us to adjust the function's input to a standard Gaussian random variable. By doing so we obtain,

$$\mathrm{Ent}_\mu(f_y(z)) = \mathrm{Ent}_\nu(g_y(\sqrt{\Sigma}(x + z))). \tag{12}$$

Then, to conclude the proof, we use integration by substitution of $g_y(z)$ and $f_y(z)$. The full proof is provided in Appendix A.1.

Theorem 4.1 addresses the independence assumption by considering the covariance of the data. It proposes that when sampling an interpretability method, one should use a

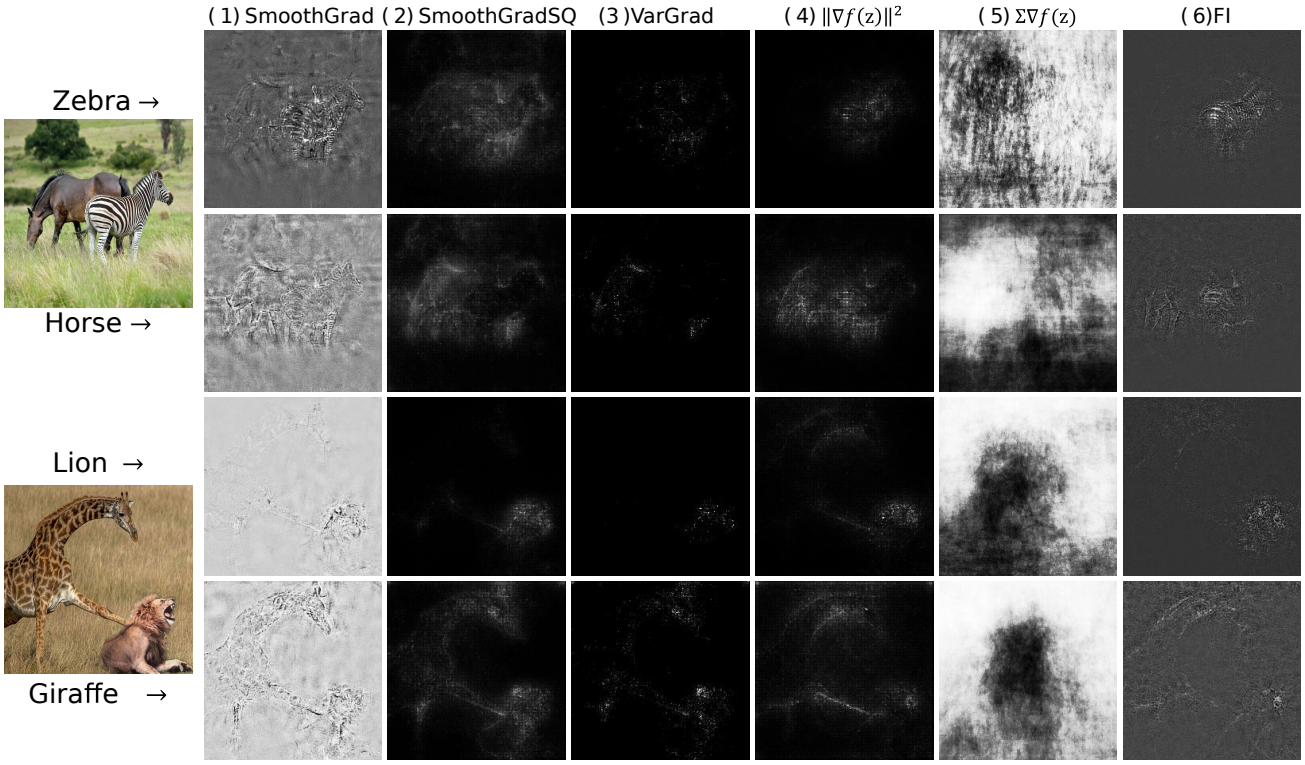

Figure 1. Different explanations of the two possible correct predictions of a fine-tuned ResNet50 network (each image has two animals and hence two correct predictions). We compare our approach (6) with various sampling-based explanation methods: (1) SmoothGrad, (2) SmoothGradSQ and (3) VarGrad. Columns (4) and (5) present the explanations of two components of our method. Particularly, (4) presents the functional information of each pixel when z is sampled from a Gaussian centered around x with the class covariance Σ, and (5) presents the left hand side of the inner product of Eq. (10).

dependent Gaussian distribution around a data point and a covariance matrix computed over the data.

Intuitively, Theorem 4.1 suggests that the contribution of the feature x_i to the total functional information $\mathcal{I}_\mu(f_y)$ is:

$$\int_{\mathbb{R}^d} \Big(\sum_j \Sigma_{ij} \nabla f_y(z)_j \Big) \nabla f_y(z)_i d\mu(z). \tag{13}$$

Note that the left hand side of the multiplication is the sum of the gradients weighted by the class covariances of the feature x_i. This weighted sum explicitly accounts for potential correlations between the features and is naturally derived from Theorem 4.1. To compute this integral we use a standard Monte Carlo sampling procedure.

Figure 1 demonstrates the importance of each of the components of the integral in Eq (13). Column (4) omits the multiplication by the covariance matrix, while column (5) does not multiply the weighted sum by the gradient. Indeed, it can be seen that the full method explanation in column (6) differs from the explanations in columns (4) and (5).

4.2. Subset Information

Multiple scenarios do not necessarily require explaining the entire input but rather only a subset of it. For example, consider the medical imaging task of detecting a tumor in a patient body scan. If the region of interest is only the stomach area, there might be no reason to explain other parts of the body and perturbate the corresponding features. There are other reasons not to perturb a subset of the features, such as when one wants to explain a prediction given fixed values or avoid heavy computations (the covariance matrix has quadratic space complexity).

In this subsection, we present a corollary of Theorem 4.1 to measure the contribution of features to a decision function when conditioning on a fixed value of another subset of the features. As a result, this allows us to measure more appropriately and efficiently the contribution of a desired subset of features or justify conditional sampling when dependencies exist in the data.

In order to do so, we first partition $x \in \mathbb{R}^d$ into (x_1, x_2) where $x_1 \in \mathbb{R}^{d_1}$ and $x_2 \in \mathbb{R}^{d_2}$ $(d_1 + d_2 = d)$. Without loss of generality, let x_1 be the set of features for which we are

interested in computing the contribution scores. Then, the expectation x and the covariance matrix Σ are:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}. \tag{14}$$

A known property of the multivariate Gaussian distribution is that any distribution of a subset of variables condition on known values of another subset of variables is also a Gaussian. We denote the conditional distribution of z_1 on z_2 and the marginal distribution of z_2 by μ_1, μ_2 respectively,

$$\begin{aligned} \mu_1 &= \mathcal{N}\left(x_1 + \Sigma_{12}\Sigma_{22}^{-1}(z_2 - x_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right), \\ \mu_2 &= \mathcal{N}(x_2, \Sigma_{22}). \end{aligned} \tag{15}$$

Corollary 4.2. *For a partitioned input $x = (x_1, x_2)$, a Gaussian measure μ, a conditional distribution μ_1, and a marginal distribution μ_2. For every non-negative function $f_y : \mathbb{R}^d \to \mathbb{R}$,*

$$\mathrm{Ent}_{\mu}(f_y) \leq \frac{1}{2}\mathbb{E}_{z_2 \sim \mu_2}\left[\mathcal{I}_{\mu_1}(f_y | z_2)\right]. \tag{16}$$

And,

$$\mathrm{Ent}_{\mu_1}(f_y | x_2) \leq \frac{1}{2}\mathcal{I}_{\mu_1}(f_y | x_2). \tag{17}$$

Our proof relies on Fubini's theorem and on the fact that μ_1 is a Gaussian. The full proof is provided in Appendix A.2.

Corollary 4.2 justifies conditional sampling and allows us to measure the contribution of a subset of features more appropriately and efficiently when conditioning on another subset.

5. Experiments

Different data modalities like vision, audio, and language have different characteristics. For example, while linguistics and audio signals are sequential, in images the spatial dimension is prominent. As another example, audio and visual signals are continuous, while text is discrete in nature. The different characteristics of these modalities have led to the development of modal-specific models and data processing pipelines. As a result, an explanation method suitable for models of one modality like vision, might fail when applied to others. This section provides a quantitative and qualitative evaluation of our framework and conducts experiments on the audio, visual, and textual modalities.

We first describe the tasks, datasets, and the explained modeling architectures we use. Then, we compare our interpretability method to previous sampling-based methods: SmoothGrad, SmoothGradSQ, and VarGrad (see Sec 2). Consequently, we qualitatively evaluate our framework. The code for our method is in the supplemental material.

Table 1. Quantitative comparison of our proposed method with previous sampling-based explanation approaches. We report AUC values as per negative perturbations evaluation (see text).

		Accuracy		Consistency
Modality	Method	GT	Predicted	Predicted
Audio	SmoothGrad	37.55	37.38	47.22
	SmoothGradSQ	34.11	33.65	39.40
	VarGrad	39.95	40.65	49.19
	Ours	**51.70**	**51.09**	**76.28**
Image	SmoothGrad	46.85	46.64	53.83
	SmoothGradSQ	54.35	51.24	58.35
	VarGrad	47.47	47.14	64.97
	Ours	**57.36**	**54.48**	**65.06**
Text	SmoothGrad	62.45	64.24	73.89
	SmoothGradSQ	64.19	64.66	76.26
	VarGrad	**66.30**	63.45	71.88
	Ours	65.72	**66.30**	**79.36**

5.1. Experimental Setup

Audio. We use the Google speech commands dataset (Warden, 2018). The dataset consists of 105,829 utterances of 35 words recorded by 2,618 speakers. This data was split into train (80%), validation (10%) and test (10%) sets. We use the M5 model proposed by Dai et al. (2017) - a CNN architecture followed by an MLP classification layer. This model achieves 73% accuracy on the validation set.

Vision. We use the CIFAR10 to evaluate our method quantitatively (Krizhevsky et al., 2009). The dataset is constructed of 50,000 images in the train set and 10,000 images in the validation set. Our model consists of two blocks of a convolution layer followed by a max-pooling layer, and this two blocks are followed by three fully connected layers, with ReLU activation (Nair & Hinton, 2010). This model scores 68% in accuracy on the validation set. For the qualitative results, we use a pre-trained version of ResNet50 (He et al., 2016) and fine-tune it on an ImageNet-like animal dataset. [1]

Text. We evaluate our method on the IMDB dataset (Maas et al., 2011), with the task of determining the binary sentiment of reviews. The train and test sets consist of 25,000 reviews, and are balanced for sentiment classes. Our model is a BiLSTM (Hochreiter & Schmidhuber, 1997) which feeds a fully connected classification layer, achieving 78% accuracy on the test set. In addition, in the qualitative evaluation, we present and compare explanations of another three architectures: LSTM, CNN (Zhang & Wallace, 2015), and a transformer (Sanh et al., 2019). Since texts are discrete, we calculate their gradients with respect their embedding.

[1] https://www.kaggle.com/antoreepjana/animals-detection-images-dataset

Table 2. Explanations generated by our method for predictions of a BiLSTM model trained to predict the sentiment of a movie review. The top two examples are erroneous predictions, while the bottom two are correct. Our method highlights the words that can best explain the model predictions.

Label	Prediction	Text
Positive	Negative	If there is a movie to be called perfect then this is it . So bad it wasn ' t intended to be . Do not miss this one !
Negative	Positive	I was very excited about rent##ing this movie but really got disappointed when I saw it . The only good thing about it are the great visual effects .
Positive	Positive	Aside from the " Thor " strand of Marvel features , the " Spider##man " stories were always my favourite##s . This latest movie is certainly the best one .
Negative	Negative	For me when a plot is based upon a fault##y or simply bad premise , everything that follows is equally fault##y and meaning##less . It is just empty .

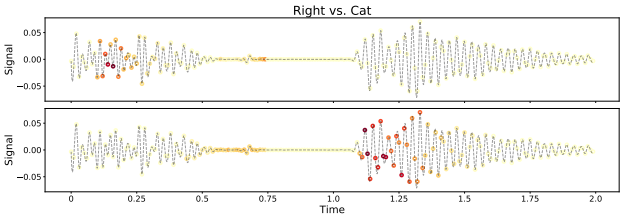

Figure 2. Explanations of the 'Right' (top) and 'Cat' (bottom) predictions for the utterance of the word 'Right' followed by the word 'Cat'. Features with high explainability scores are highlighted in color, and warmer colors correspond to higher contributions.

5.2. Quantitative Evaluation

The evaluation of interpretation methods is challenging due to the lack of a gold standard. As in previous work, we consider negative perturbations evaluation. Within this framework, we apply two evaluation measures, one of them is novelty of this work. We next describe these measures.

Negative perturbations. This procedure is composed of two stages. In the first stage, we use the trained deep net which we would like to explain, and generate an importance score for each feature in the context of each test set example. In the second stage, for each test set example, we mask increasingly large portions of the example (from 10% up to 90%, in 10% steps), in an increasing order of feature importance (from lowest to highest).[2] At each step, we measure the post-hoc accuracy and post-hoc consistency of each model interpretation method (see below). We eventually plot for each measure its measure value against the masking percentage and report the area under the curve (AUC). Since we mask features from the least to the most important, higher AUC values correspond to better explanation quality.

For the masking, in the visual setup we replace pixels with black pixels, in the textual setup we replace words with the 'unknown' token, and in the audio setup we use zero values.

Post-hoc accuracy. Following Gur et al. (2021), in this evaluation setup, we measure the model's accuracy under negative perturbations. We present results when explaining the ground truth label (GT) and the label predicted by the model. The accuracy is measured on the entire test set at each step. Table 1 (accuracy section) presents the AUC of the accuracy. It demonstrates that our method outperforms previous sampling-based methods on two of the three modalities when explaining the GT and on all modalities when explaining the predictions of the model. Additionally, the differences between our method to the previous methods are higher for the audio and image modalities. This suggests that the covariance has a higher effect on continuous data than discrete data like texts.

Post-hoc consistency. Since post-hoc accuracy is calculated with respect to the ground truth of each example, it might not highlight the features that drive the predictions of the model, but rather those features that are highly correlated the gold label. For example, consider a model which performs poorly on the test set. A good explanation method should highlight the features which contribute most to the wrong decisions of the model. Post-hoc accuracy measures how masking out features affects the performance of the model, which might not align with our goal. What we would like to understand is how consistent the model is in its predictions before and after the features are masked.

Therefore, we propose to also measure post-hoc consistency. Post-hoc consistency measures the agreement between the predictions of the model when given the masked input with its predictions when given the original unmasked input. To

[2] In the audio and image modalities, features correspond directly to the raw input. In the textual modality the features are the word embedding coordinates and the score of each word is the sum of the scores of its coordinates.

the best of our knowledge, this measure is a novel contribution of this work and we hope it will be adopted by the research community (see similar considerations in Rosenberg et al. (2021)).

Table 1 (consistency column) presents the AUC values of the post-hoc consistency. Our method, outperforms the previous sampling-based methods in all three modalities. Additionally, the differences between our method and the other methods are bigger than the differences in the post-hoc accuracy for the audio and text modalities. Finally, as discussed in Sec 4, without considering the covariance matrix, our method can be viewed as a normalized version of SmoothGradSQ. Since the fluctuations of $f_y(z)$ are small, the differences between the two methods in the post-hoc metrics are minor. To validate this, we evaluated the average Spearman's correlation between the importance scores assigned by SmoothGradSQ, and our method (without the covariance matrix). In the image modality, the correlation is 0.95 (for comparison, it equals 0.13 when the covariance is considered). Hence, the order of the features w.r.t. their scores is almost identical, and the post-hoc values are roughly the same. This observation highlights the importance of the data's covariance.

5.3. Qualitative Evaluation

A good explanation method should provide information about the features that are most associated with each of the output classes according to the model. We next examine whether our method provide such information. Following Smilkov et al. (2017); Adebayo et al. (2018); Gur et al. (2021), for this analysis we consider examples that have more than one gold label (e.g. consider the Horse and Zebra image in Figure 1).

For audio, we concatenate two utterances yielding an utterance with two commands: 'Right' and 'Cat' (Figure 2). We next run our method in order to explain the 'Right' and 'Cat' predictions of the model (the model can predict additional classes but for the sake of this analysis we are interested in these "correct" classes). The highlighted features indicate that our method concentrates on the correct part of the signal, i.e., for 'Right' it focuses on the left part of the signal and for 'Cat' it focuses on the right part.

For vision, we demonstrate our method's ability to explain the two correct predictions for images that contain two objects. For this end, we use a pre-trained version of ResNet50 (He et al., 2016) and fine-tune it on a dataset consisting of 80 animal classes.[3] We then consider the two images in Figure 1 and explain the two correct predictions of each image i.e., Horse or Zebra and Lion or Girrafe. As

[3] https://www.kaggle.com/antoreepjana/animals-detection-images-dataset

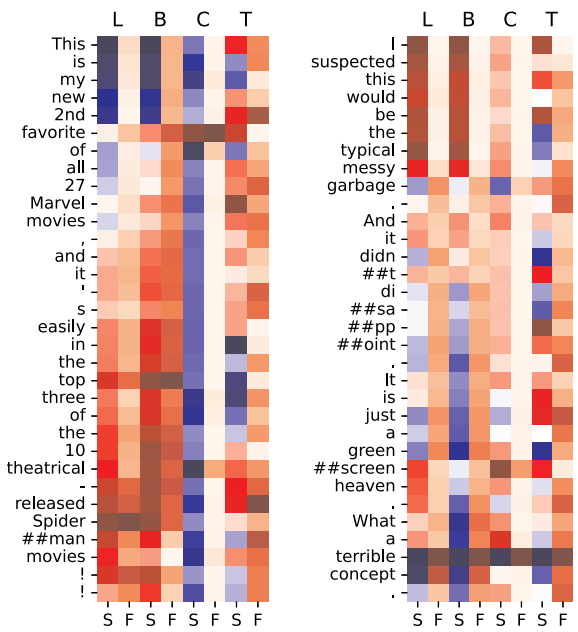

Figure 3. Explanations generated by our method (F) and SmoothGrad (S) for four architectures: LSTM (L), BiLSTM (B), CNN (C), and transformer (T), for two textual reviews. Red scores represent high scores.

can be seen in Figure 1, our method focuses on the object corresponding to the explained prediction. In addition, it reveals strong unique characteristic features of the predicted animals: the lion's mouth, the giraffe's neck and the zebra's stripes. In contrast to the previous sampling-based methods, it seems that our method is capable of distinguishing between the two animals and focus more on the the unique characteristics of each.

For text, Table 2 presents explanations for the two possible classes of four IMDB reviews. For this analysis we selected for each gold label one review with a correct model prediction and one review with an erroneous prediction. The stronger the red color is, the higher the score determined by our method. It can be observed that the highlighted words provide intuitive explanations to the predictions of the model.

Explaining text processing architectures. Unlike for the audio and the image modalities where we use a CNN-based model, for text we use a BiLSTM architecture. However, BiLSTM is not the only architecture that has demonstrated to be effective for text classification, CNNs (Zhang & Wallace, 2015) and transformers (Devlin et al., 2019) are also very prominent. In our last qualitative analysis, we compare text explanations for the following architectures: LSTM (L), BiLSTM (B), CNN (C), and transformer (T). This may

shed light on the value of our method for the different architectures. For a fair comparison, we use the vocabulary, tokenizer, and embedding layer of the pre-trained DISTILBERT model (Sanh et al., 2019) for all architectures. We compare our method (F) to the sign-preserving SmoothGrad method (S).

The heatmaps for two selected reviews are presented in Figure 3. The explanations reveal different patterns of the token contributions to the decisions of each architecture. The sequential architectures, LSTM and BiLSTM, present monotonic patterns of word importance, according to both methods. Additionally, each time the LSTM encounters a negation, contrast, objection or a changing sentiment word, the sign of the SmoothGrad score is changed. A similar pattern is demonstrated for the BiLSTM, however, since it processes the input in both directions (left to right and right to left) it seems to be more robust to these changes.

In contrast, the CNN model is only focused on a small number of relevant words. An intuitive explanation for this behavior is that CNN consists of a funnel of filters. Additionally, our method provides more interpretable explanations than SmoothGrad, highlighting a small number of features with much higher scores than the others.

Lastly, it seems that the two sampling-based methods fail to explain the transformer architecture, generating very similar scores to all participating features (differences are only in the fifth significance digit). We hypothesise that this pattern stems from the tight connections between the features, as enforced by the transformer attention mechanism.

6. Conclusion

Model interpretability plays an essential role in deep nets research. Explanations provide insights about the performance of the model beyond its test-set accuracy and, moreover, they make deep nets accessible to non-experts that can now understand their predictions. Previous sampling-based methods for model interpretability have shown strong empirical results but lack theoretical foundations. In this work we aimed to close this gap and proposed a functional informational viewpoint on sampling-based interpretability methods. We employed the log-Sobolev inequality, which allowed us to compute the functional Fisher information of a set of features. This provides a quantitative feature importance score which takes into account the covariance of the data. We have shown the efficacy of our method both quantitatively and qualitatively.

In future work we would like to explore the importance of the feature covariance matrix more deeply. For example, while we demonstrated the value of our method for three data modalities, image, text and audio, we would like to better understand in which cases the feature covariance should play a significant rule in explainability. Moreover, since the covariance matrix may impose heavy memory requirements we will aim to find effective and efficient approximations.

7. Acknowledgements

Tamir and Itai were funded by Grant No. 2029378 from the United States-Israel Binational Science Foundation (BSF). Roi Reichart and Nitay Calderon were funded by a CornellTech-Techion-AOL grant on learning from multiple modalities - text, vision and video, and by a MOST grant on Natural Interface for Cognitive Robots.

References

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. Sanity checks for saliency maps. *NeurIPS*, 2018.

Bakry, D., Gentil, I., and Ledoux, M. *Analysis and geometry of Markov diffusion operators*. SBM, 2013.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. End to end learning for self-driving cars. In *arXiv preprint arXiv:1604.07316*, 2016.

Chen, J., Song, L., Wainwright, M., and Jordan, M. Learning to explain: An information-theoretic perspective on model interpretation. In *ICML*, 2018.

Clauset, A. and Post, K. Machine learning improves hearing aids. *Science*, 2019.

Dai, W., Dai, C., Qu, S., Li, J., and Das, S. Very deep convolutional neural networks for raw waveforms. In *ICASSP*, 2017.

Deo, R. C. Machine learning in medicine. *Circulation*, 2015.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *ACL*, 2019.

Fedorov, I., Stamenovic, M., Jensen, C., Yang, L.-C., Mandell, A., Gan, Y., Mattina, M., and Whatmough, P. N. Tinylstms: Efficient neural speech enhancement for hearing aids. *arXiv preprint arXiv:2005.11138*, 2020.

Gat, I., Schwartz, I., Schwing, A., and Hazan, T. Removing bias in multi-modal classifiers: Regularization by maximizing functional entropies. In *NeurIPS*, 2020.

Gat, I., Lorberbom, G., Schwartz, I., and Hazan, T. Latent space explanation by intervention, 2021.

Green, T., Hilkhuysen, G., Huckvale, M., Rosen, S., Brookes, M., Moore, A., Naylor, P., Lightburn, L., and Xue, W. Speech recognition with a hearing-aid processing scheme combining beamforming with mask-informed speech enhancement. *Trends in Hearing*, 2022.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., et al. Recent advances in convolutional neural networks. *CVPR*, 2018.

Gur, S., Ali, A., and Wolf, L. Visualization of supervised and self-supervised neural networks via attribution guided factorization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 1997.

Krizhevsky, A., Nair, V., and Hinton, G. Cifar-10 (canadian institute for advanced research). 2009.

Lavin, A. and Gray, S. Fast algorithms for convolutional neural networks. In *CVPR*, 2016.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In *NeurIPS*, 2017.

Luss, R., Chen, P.-Y., Dhurandhar, A., Sattigeri, P., Zhang, Y., Shanmugam, K., and Tu, C.-C. Generating contrastive explanations with monotonic attribute functions. In *arXiv preprint arXiv:1905.12698*, 2019.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *ACL*, 2011.

Nair, V. and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

Nie, W., Zhang, Y., and Patel, A. A theoretical explanation for perplexing behaviors of backpropagation-based visualizations. In *ICML*. PMLR, 2018.

Ophir, Y., Tikochinski, R., Asterhan, C. S., Sisso, I., and Reichart, R. Deep neural networks detect suicide risk from textual facebook posts. *Scientific reports*, 2020.

Ribeiro, M. T., Singh, S., and Guestrin, C. Why should I trust you?: Explaining the predictions of any classifier. In *SIGKDD*, 2016.

Rosenberg, D., Gat, I., Feder, A., and Reichart, R. Are VQA systems rad? measuring robustness to augmented data with focused interventions. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *ACL*, 2021.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv:1910.01108*, 2019.

Shapley, L. S. A value for n-person games. *Contributions to the Theory of Games*, 1953.

Shrikumar, A., Greenside, P., Shcherbina, A., and Kundaje, A. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences. In *ICML*, 2017.

Shwartz-Ziv, R. and Tishby, N. Opening the black box of deep neural networks via information. *https://arxiv.org/abs/1703.00810*, 2017.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR Workshop*, 2013.

Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. Smoothgrad: removing noise by adding noise. In *ICML Workshop*, 2017.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. Striving for simplicity: The all convolutional net. In *ICLR Workshop*, 2014.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *ICML*, 2017.

Tadmor, O., Wexler, Y., Rosenwein, T., Shalev-Shwartz, S., and Shashua, A. Learning a metric embedding for face recognition using the multibatch method. *arXiv preprint arXiv:1605.07270*, 2016.

Tishby, N. and Zaslavsky, N. Deep learning and the information bottleneck principle. In *ITW*, 2015.

Warden, P. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv e-prints*, pp. arXiv–1804, 2018.

Zhang, Y. and Wallace, B. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*, 2015.

Zhou, S. K., Greenspan, H., Davatzikos, C., Duncan, J. S., Van Ginneken, B., Madabhushi, A., Prince, J. L., Rueckert, D., and Summers, R. M. A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises. *Proceedings of the IEEE*, 2021.

In this appendix, we provide complete proofs for the theorems in the main paper. Code for our method is attached.

A. Proofs

Functional entropies are defined over a continuous random variable, i.e., a function $f_y(z)$ over the Euclidean space $z \in \mathbb{R}^d$ with a Gaussian probability measure $\mu = \mathcal{N}(x, \Sigma)$ whose probability density functions is

$$d\mu(z) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} e^{-\frac{1}{2}((z-x)^\top \Sigma^{-1}(z-x))} dz. \tag{18}$$

For notational clarity we denote the standard Gaussian measure $\nu = \mathcal{N}(x, I)$, centered around x. It defers from $\mu = \mathcal{N}(x, \Sigma)$ by its covariance matrix. We aim at measuring the functional entropy of a label y of a data instance x. Here and throughout, we use z to refer to a stochastic variable, which we integrate over. The functional entropy of the non-negative label function $f_y(z) \geq 0$ is

$$\mathrm{Ent}_\mu(f_y) \triangleq \int_{\mathbb{R}^d} f_y(z) \log \frac{f_y(z)}{\int_{\mathbb{R}^d} f_y(z) d\mu(z)} d\mu(z). \tag{19}$$

We hence define the functional entropy of a deep net with respect to a label y by the function softmax output $f_y(z)$ when $z \sim \mu$ is sampled from a Gaussian distribution around x. The functional entropy is non-negative, namely $\mathrm{Ent}_\mu(f_y) \geq 0$ and equals zero only if $f_y(z)$ is a constant. This is in contrast to differential entropy of a continuous random variable with probability density function $q(z)$: $h(q) = -\int_{\mathbb{R}^d} q(z) \log q(z) dz$, which is defined for $q(z) \geq 0$ with $\int_{\mathbb{R}^d} q(z) dz = 1$ and may be negative.

We denote the functional Fisher information with

$$\mathcal{I}_\nu(f_y) \triangleq \int_{\mathbb{R}^d} \frac{\|\nabla f_y(z)\|^2}{f_y(z)} d\nu(z) \tag{20}$$

Hereby, $\|\nabla f_y(z)\|$ is the ℓ_2 norm of the gradient of f. The functional Fisher information is a natural extension of the Fisher information, which is defined for probability density functions. Specifically, the log-Sobolev inequality for any non-negative function $f_y(z) \geq 0$ is,

$$\mathrm{Ent}_\nu(f_y) \leq \frac{1}{2} \mathcal{I}_\nu(f_y). \tag{21}$$

A.1. Theorem 1

We derive a log-Sobolev inequality for dependent Gaussian measures. The functional Fisher information for dependent Gaussian measure is

$$\mathcal{I}_\mu(f_y) \triangleq \int_{\mathbb{R}^d} \frac{\langle \Sigma \nabla f_y(z), \nabla f_y(z) \rangle}{f_y(z)} d\mu(z). \tag{22}$$

Theorem A.1. *For every non-negative function $f_y : \mathbb{R}^d \to \mathbb{R}$ and a Gaussian measure μ,*

$$\mathrm{Ent}_\mu(f_y) \leq \frac{1}{2} \mathcal{I}_\mu(f_y). \tag{23}$$

Proof. We use integration by substitution to adjust the function's input to a standard Gaussian random variable. We denote by $\sqrt{\Sigma}$ the matrix for which $\Sigma = \sqrt{\Sigma}^\top \sqrt{\Sigma}$. Define $\phi(t) = t \log t$,

$$\mathrm{Ent}_\mu(f_y) = \int_{\mathbb{R}^d} \phi(f_y(z)) d\mu(z) - \phi(\int_{\mathbb{R}^d} f_y(z) d\mu(z)). \tag{24}$$

With a change of variable $z \leftarrow \sqrt{\Sigma}(x + z)$, we get $\int_{\mathbb{R}^d} \phi(f_y(z)) d\mu(z) =$

$$\int_{\mathbb{R}^d} \frac{e^{-\frac{1}{2}(z^\top z)}}{\sqrt{(2\pi)^d}} \phi(f_y(\sqrt{\Sigma}(x + z))) dz. \tag{25}$$

Therefore, after a change of variables, the distribution changes to a Gaussian distribution with a zero expectation and an identity covariance matrix,

$$\int_{\mathbb{R}^d} \phi(f_y(z))d\mu(z) = \int_{\mathbb{R}^d} \phi(f_y(\sqrt{\Sigma}(x+z)))d\nu(z). \tag{26}$$

Similarly, for the second term of the functional entropy,

$$\phi(\int_{\mathbb{R}^d} f_y(z)d\mu(z)) = \phi(\int_{\mathbb{R}^d} f_y(\sqrt{\Sigma}(x+z))d\nu(z)). \tag{27}$$

Consequently,

$$\mathrm{Ent}_\mu(f_y(z)) = \mathrm{Ent}_\nu(g_y(\sqrt{\Sigma}(x+z))). \tag{28}$$

With this, we can apply the log-Sobolev inequality for the standard normal distribution,

$$\mathrm{Ent}_\nu(g_y) \leq \frac{1}{2}\int_{\mathbb{R}^d} \frac{\langle \nabla g_y(z), \nabla g_y(z)\rangle}{g_y(z)} d\nu(z). \tag{29}$$

We conclude the proof by applying the chain rule,

$$\nabla g_y(z) = \sqrt{\Sigma}\nabla f_y(\sqrt{\Sigma}(x+z)). \tag{30}$$

Hence,

$$\int_{\mathbb{R}^d} \frac{\|\nabla g_y(z)\|^2}{g_y(z)} d\nu(z) = \int_{\mathbb{R}^d} \frac{\langle \sqrt{\Sigma}\nabla f_y(\sqrt{\Sigma}(x+z)), \sqrt{\Sigma}\nabla f_y(\sqrt{\Sigma}(x+z))\rangle}{f_y(\sqrt{\Sigma}(x+z))} d\nu(z) \tag{31}$$

Lastly, we apply integration by substitution, thus it equals

$$\frac{1}{2}\int_{\mathbb{R}^d} \frac{\langle \sqrt{\Sigma}\nabla f_y(z), \sqrt{\Sigma}\nabla f_y(z)\rangle}{f_y(z)} d\mu(z). \tag{32}$$

□

A.2. Corollary 2

We partition $x \in \mathbb{R}^d$ into (x_1, x_2) where $x_1 \in \mathbb{R}^{d_1}$ and $x_2 \in \mathbb{R}^{d_2}$. Without loss of generality, x_1 is the set of features we are interested in. Then, the expectation x and the covariance matrix Σ are

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \tag{33}$$

Corollary A.2 (Conditional functional Fisher information)**.** *For a partitioned input $x = (x_1, x_2)$, a Gaussian measure μ, a conditional distribution μ_1, and a marginal distribution μ_2. For every non-negative function $f_y : \mathbb{R}^d \to \mathbb{R}$,*

$$\mathrm{Ent}_\mu(f_y) \leq \frac{1}{2}\mathbb{E}_{z_2\sim\mu_2}\left[\mathcal{I}_{\mu_1}(f_y|z_2)\right] \tag{34}$$

And,

$$\mathrm{Ent}_{\mu_1}(f_y|x_2) \leq \frac{1}{2}\mathcal{I}_{\mu_1}(f_y|x_2). \tag{35}$$

Proof. We extend Theorem A.1 computing the information of a subset of features. From Theorem A.1,

$$\mathrm{Ent}_\mu(f_y) \leq \frac{1}{2}\int_{\mathbb{R}^d} \frac{\langle \sqrt{\Sigma}\nabla f_y(z), \sqrt{\Sigma}\nabla f_y(z)\rangle}{f_y(z)} d\mu(z). \tag{36}$$

By applying Fubini's theorem on Eq. (36) we get

$$\mathrm{Ent}_\mu(f_y) \leq \frac{1}{2}\int_{\mathbb{R}^{d_2}}\int_{\mathbb{R}^{d_1}} \frac{\langle \Sigma\nabla f_y(z)), \nabla f_y(z))\rangle}{f_y(z)} d\mu_1(z_1)\, d\mu_2(z_2). \tag{37}$$

□

A.3. Log-Sobolev tightness example

In the following, we derive the log-Sobolev bound for $f(x) = e^x$ to demonstrate the tightness of it for exponential family.

$$\begin{aligned} \mathrm{Ent}_\mu\, e^x &= \int_{\mathbb{R}^d} x e^x d\mu(x) - \int_{\mathbb{R}^d} e^x d\mu(x) \log\left(\int_{\mathbb{R}^d} e^x d\mu(x)\right) && (38)\\ &= (\mu+\sigma^2)e^{\mu+\frac{1}{2}\sigma^2} - (\mu+\frac{1}{2}\sigma^2)e^{\mu+\frac{1}{2}\sigma^2} && (39)\\ &= \frac{1}{2}\sigma^2 e^{\mu+\frac{1}{2}\sigma^2} = \frac{1}{2}\mathcal{I}_\mu(e^x). && (40) \end{aligned}$$

Hence, the bound hold with equality for e^x. This result can be easily extended to a multivariate Gaussian distribution.

B. Covariance Matrix Calculation

The covariance matrix is a crucial component of our proposed explainability method. In order to explain an output class y, the covariance matrix of that class Σ need to be estimated empirically. The covariance matrix may impose heavy memory requirements (the size of the covariance matrix of d-dimensional feature vectors is d^2).

In cases of high-dimensional feature vectors, we can partition the features into subsets and sample according to the sampling protocol discussed in Sec. 4.2. Alternatively, one may partition the features into subsets and assume that each subset shares the same covariance matrix. For example, partitioning the features of an image into three subsets, one for each color channel, and assuming all color channels share the same covariance matrix, resulting in a nine times smaller memory usage.

In our experiments, we used shared covariances for vision and text modalities. For the qualitative vision experiments, we used a shared covariance matrix in the size of $H \cdot W \times H \cdot W$ while assuming each color channel share the same covariance matrix. For text, the size of the covariance matrix is $d \times d$, where d is the embedding size, assuming words in the embedding space share the same covariance matrix regardless of their position in a sentence.

Lastly, Σ is required to be a positive-definite matrix. In the case where some of the features are constant (e.g., the top row in the MNIST dataset is always black), or when the dimension of the feature vectors is higher than the size of the examples of class y, Σ will not be a positive-definite matrix. Hence, we suggest adding a small noise to the diagonal of Σ, which is a well-known practice to modify the matrix to be positive-definite.