Contextual Explanation Networks

Maruan Al-Shedivat ALSHEDIVAT@CS.CMU.EDU
Avinava Dubey AKDUBEY@CS.CMU.EDU
Carnegie Mellon University

Eric P. Xing EPXING@CS.CMU.EDU
Carnegie Mellon University & Petuum Inc.

Abstract

Modern learning algorithms excel at producing accurate but complex models of the data. However, deploying such models in the real-world requires extra care: we must ensure their reliability, robustness, and absence of undesired biases. This motivates development of models that are equally accurate but can be also easily inspected and assessed beyond their predictive performance. To this end, we introduce *contextual explanation networks* (CENs)—a class of architectures that learn to predict by generating and utilizing intermediate, simplified probabilistic models. Specifically, CENs generate parameters for intermediate graphical models which are further used for prediction and play the role of explanations. Contrary to the existing *post-hoc* model-explanation tools, CENs learn to predict and to explain jointly. Our approach offers two major advantages: (i) for each prediction, valid, instance-specific explanations are generated with no computational overhead and (ii) prediction via explanation acts as a regularizer and boosts performance in low-resource settings. We analyze the proposed framework theoretically and experimentally. Our results on image and text classification and survival analysis tasks demonstrate that CENs are not only competitive with the state-of-the-art methods but also offer additional insights behind each prediction, that are valuable for decision support. We also show that while post-hoc methods may produce misleading explanations in certain cases, CENs are always consistent and allow to detect such cases systematically.

1. Introduction

Model interpretability is a long-standing problem in machine learning that has become quite acute with the accelerating pace of the widespread adoption of complex predictive algorithms. While high performance often supports our belief in the predictive capabilities of a system, perturbation analysis reveals that black-box models can be easily broken in an unintuitive and unexpected manner (Szegedy et al., 2013; Nguyen et al., 2015). Therefore, for a machine learning system to be used in a social context (*e.g.*, in healthcare) it is imperative to provide sound reasoning for each prediction or decision it makes.

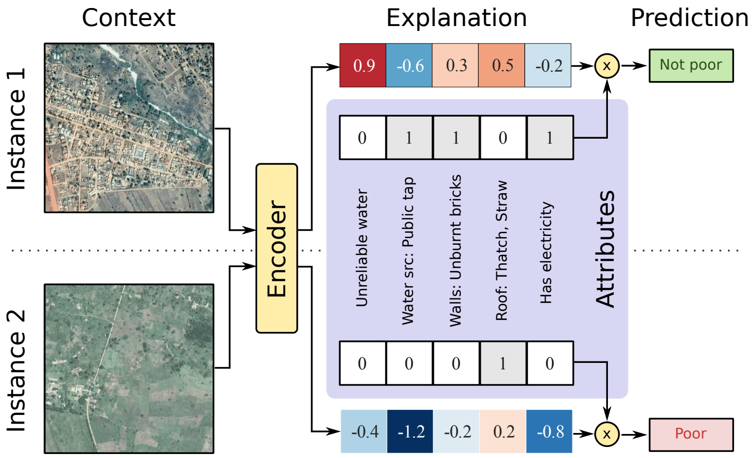


Figure 1: High-level functionality of CENs: The context is represented by satellite imagery and used to generate instance-specific linear models (explanations). The latter act on a set of interpretable attributes from regional survey data and produce predictions.

To design such systems, we may restrict the class of models to only *human-intelligible* (Caruana et al., 2015). However, such an approach is often limiting in modern practical settings. Alternatively, we may fit a complex model and explain its predictions *post-hoc*, *e.g.*, by searching for linear local approximations of the decision boundary (Ribeiro et al., 2016). While such methods achieve their goal, explanations are generated *a posteriori*, require additional computation per data instance, and most importantly are never the basis for the predictions made in the first place, which may lead to erroneous interpretations[1].

Explanation is a fundamental part of the human learning and decision process (Lombrozo, 2006). Inspired by this fact, we introduce *contextual explanation networks* (CENs)—a class of architectures that learn to predict and to explain jointly, alleviating the drawbacks of the post-hoc methods. To make a prediction, CENs operate as follows (Figure 1). First, they process a subset of inputs and generate parameters for a simple probabilistic model (*e.g.*, sparse linear model) which is regarded interpretable by a domain expert. Then, the generated model is applied to another subset of inputs and produces a prediction. To motivate such an architecture, we consider the following example.

A motivating illustration. One of the tasks we consider in this paper is classification of households into poor and not poor having access to satellite imagery and categorical data from surveys (Jean et al., 2016). If a human were to solve this task, to make predictions, they might assign weights to features in the categorical data and explain their predictions in terms of the most relevant variables. Moreover, depending on the type of the area (based on

---

[1] As we show in the paper, meaning and quality of generated explanations highly depends on the quality of the features in terms of which predictions are explained. These so-called "interpretable features" are often a matter of an arbitrary choice, but may significantly affect explanations produced post-hoc.

the available imagery), they might select slightly different weights for different areas (*e.g.*, when features indicative of poverty are different for urban and rural areas).

The CEN architecture given in Figure 1 imitates this process by learning an encoder that maps images (the context) to parameters of sparse linear models which are further used for prediction. The learned encoder is sensitive to the infrastructure presented in the input images and generates different linear models for urban and rural areas. The generated models not only are used for prediction but also play the role of explanations and can encode arbitrary prior knowledge. CENs can represent complex model classes by using powerful encoders. At the same time, by offsetting complexity into the encoding process, we achieve simplicity of explanations and can interpret predictions in terms the variables of interest.

The proposed architecture opens a number of questions: What are the fundamental advantages and limitations of CEN? How much of the performance should be attributed to the context encoder and how much to the explanations? Are there any degenerate cases and do they happen in practice? Finally, how do CEN-generated explanations compare to alternatives, *e.g.*, produced with LIME (Ribeiro et al., 2016)? In the sequel, we formalize our intuitions and answer these questions theoretically and experimentally.

1.1 Contributions

The main four contributions of this paper are as follows:

(i) We formally define CENs as a class of probabilistic models, consider special cases, and derive learning and inference algorithms for scalar and structured outputs.
(ii) We design CENs in the form of new deep learning architectures trainable end-to-end for prediction and survival analysis tasks.
(iii) Empirically, we demonstrate the value of learning with explanations for both prediction and model diagnostics. Moreover, we find that explanations can act as a regularizer and result in improved sample efficiency.
(iv) We also show that noisy features can render post-hoc explanations inconsistent and misleading, and how CENs can help to detect and avoid such situations.

The code for reproducing experiments presented in this paper will be made publicly available.

1.2 Organization

The paper is organized as follows. Section 2 gives an overview of the related work. Section 3 introduces our notation and some background on post-hoc interpretability methods. In Sections 4, we introduce the general CEN framework, describe specific implementations, learning, and inference. In Section 5, we discuss and analyze properties of CEN theoretically. In Sections 6.1–6.2, we present experimental results for scalar prediction tasks and analyze consistency of linear explanations generated by CEN vs. alternatives. Finally, Section 6.3 shows how CENs with structured explanations can efficiently solve survival analysis tasks.

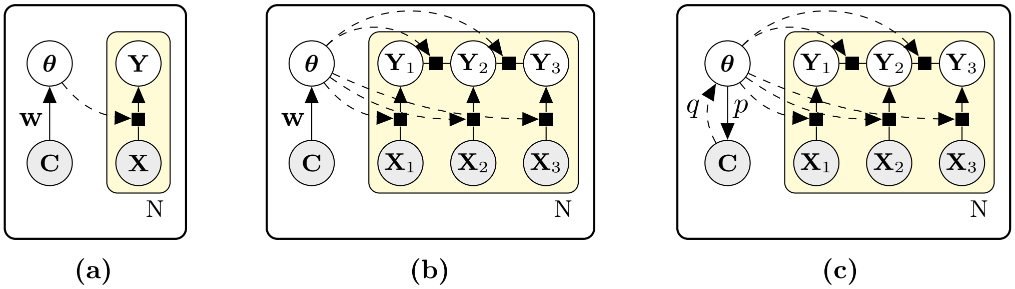


Figure 2: (a) A graphical model for CEN with a context encoder parameterized by $\mathbf{w}$ and linear explanations. (b) A graphical model for CEN with context encoder and CRF-based explanations. The model is parameterized by $\mathbf{w}$. (c) A graphical model for CEN with context autoencoding via the inference, q, and generator, p, networks and CRF-based explanations.

2. Related work

Contextual explanation networks combine multiple threads of research that we discuss below.

2.1 Deep graphical models

The idea of combining deep networks with graphical models has been explored extensively. Notable threads of recent work include: replacing task-specific feature engineering with task-agnostic general representations (or embeddings) discovered by deep networks (Collobert et al., 2011; Rudolph et al., 2016, 2017), representing energy functions (Belanger and McCallum, 2016) and potential functions (Jaderberg et al., 2014) with neural networks, encoding learnable structure into Gaussian processes with deep and recurrent networks (Wilson et al., 2016; Al-Shedivat et al., 2017), or learning state-space models on top of nonlinear embeddings of the observations (Gao et al., 2016; Johnson et al., 2016; Krishnan et al., 2017). The goal of this body of work is to design principled structured probabilistic models that enjoy the flexibility of deep learning. The key difference between CENs and the previous art is that the latter directly integrate neural networks *into* graphical models as components (embeddings, potential functions, etc.). While flexible, the resulting *deep graphical models* could no longer be interpreted in terms of crisp relationships between specific variables of interest[2]. CENs, on the other hand, preserve simplicity of the explanations and shift complexity into conditioning on the context.

---

[2] To see why this is the case, consider graphical models given in Figure 2 which relate input, $\mathbf{X}$, and target, $\mathbf{Y}$, variables using linear pairwise potential functions. Linearity allows to directly interpret parameters of the model as associations between the variables. Substituting inputs, $\mathbf{X}$, with deep representations or defining potentials via neural networks would result in a more powerful model. However, precise relationships between the variables will be no longer directly readable from the model parameters.

2.2 Context representation

Generating probabilistic models after conditioning on a context is the key aspect of our approach. Previous work on context-specific graphical models represented contexts with a discrete variable that enumerated a finite number of possible contexts (Koller and Friedman, 2009, Ch. 5.3). CENs, on the other hand, are designed to handle arbitrary complex context representations. Context-specific approaches are widely used in language modeling where the context is typically represented with trainable embeddings (Rudolph et al., 2016). We also note that few-shot learning explicitly considers a setup where the context is represented by a small set of labeled examples (Santoro et al., 2016; Garnelo et al., 2018).

2.3 Meta-learning

The way CENs operate resembles the meta-learning setup. In meta-learning, the goal is to learn a meta-model which, given a task, can produce another model capable of solving the task (Thrun and Pratt, 1998). The representation of the task can be seen as the context while produced task-specific models are similar to CEN-generated explanations. Meta-training a deep network that generates parameters for another network has been successfully used for zero-shot (Lei Ba et al., 2015; Changpinyo et al., 2016) and few-shot (Edwards and Storkey, 2016; Vinyals et al., 2016) learning, cold-start recommendations (Vartak et al., 2017), and a few other scenarios (Bertinetto et al., 2016; De Brabandere et al., 2016; Ha et al., 2016), but is not suitable for interpretability purposes. In contrast, CENs generate parameters for models from a restricted class (potentially, based on domain knowledge) and use the attention mechanism (Xu et al., 2015) to further improve interpretability.

2.4 Model interpretability

While there are many ways to define interpretability (Lipton, 2016; Doshi-Velez and Kim, 2017), our discussion focuses on explanations defined as simple models that locally approximate behavior of a complex model. A few methods that allow to construct such explanations in a *post-hoc* manner have been proposed recently (Ribeiro et al., 2016; Shrikumar et al., 2017; Lundberg and Lee, 2017), some of which we review in the next section. In contrast, CENs learn to generate such explanations along with predictions. There are multiple other complementary approaches to interpretability ranging from a variety of visualization techniques (Simonyan and Zisserman, 2014; Yosinski et al., 2015; Mahendran and Vedaldi, 2015; Karpathy et al., 2015), to explanations by example (Caruana et al., 1999; Kim et al., 2014, 2016; Koh and Liang, 2017), to natural language rationales (Lei et al., 2016). Finally, our framework encompasses the so-called *personalized* or *instance-specific* models that learn to partition the space of inputs and fit local sub-models (Wang and Saligrama, 2012).

3. Background

We start by introducing the notation and reviewing post-hoc model explanations, with a focus on LIME (Ribeiro et al., 2016) as one of the most popular frameworks to date.

Given a collection of data where each instance is represented by inputs, $\mathbf{c} \in \mathcal{C}$, and targets, $\mathbf{y} \in \mathcal{Y}$, our goal is to learn an accurate predictive model, $f : \mathcal{C} \mapsto \mathcal{Y}$. To explain predictions, we can assume that each data point has another set of features, $\mathbf{x} \in \mathcal{X}$. We construct explanations in the form of simpler models, $g_{\mathbf{c}} : \mathcal{X} \mapsto \mathcal{Y}$, so that they are consistent with the original model in the neighborhood of the corresponding data instance, *i.e.*, $g_{\mathbf{c}}(\mathbf{x}) = f(\mathbf{c})$. While the original inputs, $\mathbf{c}$, can be of complex, low-level, unstructured data types (*e.g.*, text, image pixels, sensory inputs), we assume that $\mathbf{x}$ are high-level, meaningful variables (*e.g.*, categorical features). In the post-hoc explanation literature, it is assumed that $\mathbf{x}$ are derived from $\mathbf{c}$ and are often binary (Lundberg and Lee, 2017) (*e.g.*, $\mathbf{c}$ can be images, while $\mathbf{x}$ can be vectors of binary indicators over the corresponding super-pixels). We consider a more general setup where $\mathbf{c}$ and $\mathbf{x}$ can be arbitrary, non-derivative modalities of the data. Throughout the paper, we call $\mathbf{c}$ the *context* and $\mathbf{x}$ the *attributes* or *variables of interest*.

Locally Interpretable Model-agnostic Explanations (LIME)

Given a trained model, f, and a data instance with features $(\mathbf{c}, \mathbf{x})$, LIME constructs an explanation, $g_{\mathbf{c}}$, as follows:

$$g_{\mathbf{c}} = \underset{g \in \mathcal{G}}{\arg\min}\, \mathcal{L}(f, g, \pi_{\mathbf{c}}) + \Omega(g) \tag{1}$$

where $\mathcal{L}(f, g, \pi_{\mathbf{c}})$ is the loss that measures how well g approximates f in the neighborhood defined by the similarity kernel, $\pi_{\mathbf{c}} : \mathcal{X} \mapsto \mathbb{R}_+$, in the space of attributes, $\mathcal{X}$, and $\Omega(g)$ is the penalty on the complexity of explanation[3]. Now more specifically, Ribeiro et al. (2016) assume that $\mathcal{G}$ is the class of linear models, $g_{\mathbf{c}}(\mathbf{x}) := b_{\mathbf{c}} + \mathbf{w}_{\mathbf{c}} \cdot \mathbf{x}$, and define the loss and the similarity kernel as follows:

$$\mathcal{L}(f, g, \pi_{\mathbf{c}}) := \sum_{\mathbf{x}' \in \mathcal{X}} \pi_{\mathbf{c}}(\mathbf{x}') \left(f(\mathbf{c}') - g(\mathbf{x}')\right)^2, \quad \pi_{\mathbf{c}}(\mathbf{x}') := \exp\left\{-D(\mathbf{x}, \mathbf{x}')^2 / \sigma^2\right\} \tag{2}$$

where the data instance of interest is represented by $(\mathbf{c}, \mathbf{x})$, $\mathbf{x}'$ and the corresponding $\mathbf{c}'$ are the perturbed features, $D(\mathbf{x}, \mathbf{x}')$ is some distance function, and σ is the scale parameter of the kernel. The regularizer, $\Omega(g)$, is often chosen to favor sparse explanations.

The model-agnostic property is the key advantage of LIME (and variations)—we can solve (1) for any trained model, f, any class of explanations, $\mathcal{G}$, at any point of interest, $(\mathbf{c}, \mathbf{x})$. While elegant, predictive and explanatory models in this framework are learned independently and hence never affect each other. In the next section, we propose a class of models that ties prediction and explanation together in a joint probabilistic framework.

---

[3] Ribeiro et al. (2016) argue that only simple models of low complexity (*e.g.*, sufficiently sparse linear models) are human-interpretable and support that by human studies.

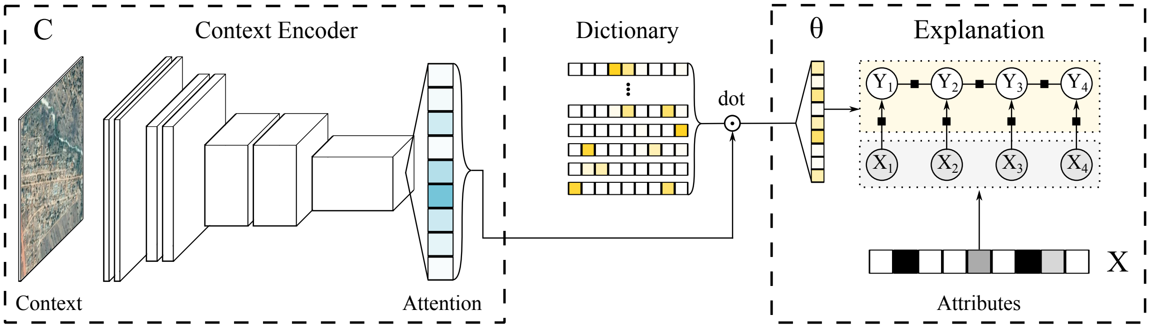


Figure 3: An example of CEN architecture. The context is represented by an image and transformed by a convnet encoder into an attention vector, which is used to construct a contextual hypothesis from a dictionary of sparse atoms.

4. Contextual Explanation Networks

We consider the same problem of learning from a collection of data represented by context variables, $\mathbf{c} \in \mathcal{C}$, attributes, $\mathbf{x} \in \mathcal{X}$, and targets, $\mathbf{y} \in \mathcal{Y}$. We denote the corresponding random variables by capital letters, $\mathbf{C}$, $\mathbf{X}$, and $\mathbf{Y}$, respectively. Our goal is to learn a model, $\mathbb{P}_{\mathbf{w}}\left(\mathbf{Y} \mid \mathbf{x}, \mathbf{c}\right)$, parametrized by $\mathbf{w}$ that can predict $\mathbf{y}$ from $\mathbf{x}$ and $\mathbf{c}$. We define contextual explanation networks as probabilistic models that assume the following form[4] (Figure 2):

$$\mathbf{y} \sim \mathbb{P}\left(\mathbf{Y} \mid \mathbf{x}, \boldsymbol{\theta}\right), \quad \boldsymbol{\theta} \sim \mathbb{P}_{\mathbf{w}}\left(\boldsymbol{\theta} \mid \mathbf{c}\right), \quad \mathbb{P}_{\mathbf{w}}\left(\mathbf{Y} \mid \mathbf{x}, \mathbf{c}\right) = \int \mathbb{P}\left(\mathbf{Y} \mid \mathbf{x}, \boldsymbol{\theta}\right) \mathbb{P}_{\mathbf{w}}\left(\boldsymbol{\theta} \mid \mathbf{c}\right) d\boldsymbol{\theta} \tag{3}$$

where $\mathbb{P}\left(\mathbf{Y} \mid \mathbf{x}, \boldsymbol{\theta}\right)$ is a predictor parametrized by $\boldsymbol{\theta}$. We call such predictors *explanations*, since they explicitly relate interpretable attributes, $\mathbf{x}$, to the targets, $\mathbf{y}$. For example, when the targets are scalar and binary, explanations may take the form of linear logistic models; when the targets are more complex, dependencies between the components of $\mathbf{y}$ can be represented by a graphical model, *e.g.*, *conditional random field* (Lafferty et al., 2001).

CENs assume that each explanation is context-specific: $\mathbb{P}_{\mathbf{w}}\left(\boldsymbol{\theta} \mid \mathbf{c}\right)$ defines a conditional probability of an explanation $\boldsymbol{\theta}$ being valid in the context $\mathbf{c}$. To make a prediction, we marginalize out $\boldsymbol{\theta}$. To interpret a prediction, $\hat{\mathbf{y}}$, for a given data instance, $(\mathbf{x}, \mathbf{c})$, we infer the posterior, $\mathbb{P}_{\mathbf{w}}\left(\boldsymbol{\theta} \mid \hat{\mathbf{y}}, \mathbf{x}, \mathbf{c}\right)$. The main advantage of this approach is to allow modeling conditional probabilities, $\mathbb{P}_{\mathbf{w}}\left(\boldsymbol{\theta} \mid \mathbf{c}\right)$, in a black-box fashion while keeping the class of explanations, $\mathbb{P}\left(\mathbf{Y} \mid \mathbf{x}, \boldsymbol{\theta}\right)$, simple and interpretable. For instance, when the context is given as raw text, we may choose $\mathbb{P}_{\mathbf{w}}\left(\boldsymbol{\theta} \mid \mathbf{c}\right)$ to be represented with a recurrent neural network, while $\mathbb{P}\left(\mathbf{Y} \mid \mathbf{x}, \boldsymbol{\theta}\right)$ be in the class of linear models.

Implications of these assumptions are discussed in Section 5. Here, we continue with a discussion of a number of practical choices for $\mathbb{P}_{\mathbf{w}}\left(\boldsymbol{\theta} \mid \mathbf{c}\right)$ and $\mathbb{P}\left(\mathbf{Y} \mid \mathbf{x}, \boldsymbol{\theta}\right)$ (Table 1).

---

[4] While we focus on predictive modeling, CENs are applicable beyond that. For example, instead of learning a predictive distribution, $\mathbb{P}_{\mathbf{w}}\left(\mathbf{Y} \mid \mathbf{x}, \mathbf{c}\right)$, we may want to learn a contextual marginal distribution, $\mathbb{P}_{\mathbf{w}}\left(\mathbf{X} \mid \mathbf{c}\right)$, over a set random variables $\mathbf{X}$, where $\mathbb{P}\left(\mathbf{X} \mid \boldsymbol{\theta}\right)$ is defined by an arbitrary graphical model.

Table 1: Different types of encoders and explanations used in CEN.

Encoder	**Parameter distribution, $\mathbb{P}\left(\boldsymbol{\theta} \mid \mathbf{c}\right)$**	**Explanation**	**Predictive distribution, $\mathbb{P}\left(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}\right)$**
Deterministic	$\delta\left(\phi(\mathbf{c}), \boldsymbol{\theta}\right)$ where $\phi(\mathbf{c})$ is arbitrary	Linear	$\operatorname{softmax}\left(\boldsymbol{\theta}^\top \mathbf{x}\right)$
Constrained	$\delta\left(\phi(\mathbf{c}), \boldsymbol{\theta}\right)$ where $\phi(\mathbf{c}) := \boldsymbol{\alpha}(\mathbf{c})^\top \mathbf{D}$	Structured	$\propto \exp\left\{-E_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y})\right\}$ where $E_{\boldsymbol{\theta}}(\cdot, \cdot)$ is some energy function, linear in $\boldsymbol{\theta}$
MoE	$\sum_{k=1}^K \mathbb{P}\left(k \mid \mathbf{c}\right) \delta(\boldsymbol{\theta}, \boldsymbol{\theta}_k)$		

4.1 Context Encoders

In practice, we represent $\mathbb{P}_{\mathbf{w}}\left(\boldsymbol{\theta} \mid \mathbf{c}\right)$ with a neural network that encodes the context into the parameter space of the explanation models. There are two simple ways to construct an encoder, which we consider below.

4.1.1 Deterministic encoding

Let $\mathbb{P}_{\mathbf{w}}\left(\boldsymbol{\theta} \mid \mathbf{c}\right) := \delta\left(\phi_{\mathbf{w}}(\mathbf{c}), \boldsymbol{\theta}\right)$, where $\delta(\cdot, \cdot)$ is a delta-function and $\phi_{\mathbf{w}}$ is the network that maps $\mathbf{c}$ to $\boldsymbol{\theta}$. Collapsing the conditional distribution to a delta-function makes $\boldsymbol{\theta}$ depend deterministically on $\mathbf{c}$ and results into the following conditional likelihood:

$$\mathbb{P}\left(\mathbf{y} \mid \mathbf{x}, \mathbf{c}; \mathbf{w}\right) = \int \mathbb{P}\left(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}\right) \delta\left(\phi_{\mathbf{w}}(\mathbf{c}), \boldsymbol{\theta}\right) d\boldsymbol{\theta} = \mathbb{P}\left(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta} = \phi_{\mathbf{w}}(\mathbf{c})\right) \tag{4}$$

Modeling $\mathbb{P}_{\mathbf{w}}\left(\boldsymbol{\theta} \mid \mathbf{c}\right)$ with a delta-function is convenient since the posterior, $\mathbb{P}_{\mathbf{w}}\left(\boldsymbol{\theta} \mid \mathbf{y}, \mathbf{x}, \mathbf{c}\right) \propto \mathbb{P}\left(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}\right) \delta\left(\phi_{\mathbf{w}}(\mathbf{c}), \boldsymbol{\theta}\right)$ also collapses to $\boldsymbol{\theta}^\star = \phi_{\mathbf{w}}(\mathbf{c})$, hence the inference is done via a single forward pass and the posterior can be regularized by imposing L_1 or L_2 losses on $\phi_{\mathbf{w}}(\mathbf{c})$.

4.1.2 Constrained deterministic encoding

The downside of deterministic encoding is the lack of constraints on the generated explanations. There are multiple reasons why this might be an issue: (i) when the context encoder is unrestricted, it might generate unstable, overfitted local models, (ii) when we want to reason about the patterns in the data as a whole, local explanations are not enough. To address these issues, we constrain the space of explanations by introducing a *global dictionary*, $\mathbf{D} := \{\boldsymbol{\theta}_k\}_{k=1}^K$, where each atom of the dictionary, $\boldsymbol{\theta}_k$, is sparse. The encoder generates context-specific explanations using *soft attention* over the dictionary (Figure 3):

$$\phi_{\mathbf{w}, \mathbf{D}}(\mathbf{c}) = \sum_{k=1}^K \mathbb{P}_{\mathbf{w}}\left(k \mid \mathbf{c}\right) \boldsymbol{\theta}_k = \boldsymbol{\alpha}_{\mathbf{w}}(\mathbf{c})^\top \mathbf{D}, \quad \sum_{k=1}^K \boldsymbol{\alpha}_{\mathbf{w}}^{(k)}(\mathbf{c}) = 1, \quad \forall k : \boldsymbol{\alpha}_{\mathbf{w}}^{(k)}(\mathbf{c}) \geq 0, \tag{5}$$

where $\boldsymbol{\alpha}_{\mathbf{w}}(\mathbf{c})$ is the attention over the dictionary produced by the encoder. Attention-based construction of explanations using a global dictionary (i) forces the encoder to produce models shared across different contexts, (ii) allows us to interpret the learned dictionary atoms as global "explanation modes." Again, since $\mathbb{P}_{\mathbf{w}}\left(\boldsymbol{\theta} \mid \mathbf{c}\right)$ is a delta-distribution, the likelihood is the same as given in (4) and inference is conveniently done via a forward pass.

The two proposed context encoders represent $\mathbb{P}(\boldsymbol{\theta} \mid \mathbf{c})$ with delta-functions, which simplifies learning, inference, and interpretation of the model, and are used in our experiments. Other ways to represent $\mathbb{P}(\boldsymbol{\theta} \mid \mathbf{c})$ include: (i) using a mixture of delta-functions (which makes CEN function similar to a mixture-of-experts model and further discussed in Section 5.1), or (ii) using variational autoencoding. We leave more complex approaches to future research.

4.2 Explanations

In this paper, we consider two types of explanations: *linear* that can be used for regression or classification and *structured* that are suitable for structured prediction.

4.2.1 Linear Explanations

In case of classification, CENs with linear explanations assume the following $\mathbb{P}(\mathbf{Y} \mid \mathbf{x}, \boldsymbol{\theta})$:

$$\mathbb{P}(\mathbf{Y} = i \mid \mathbf{x}, \boldsymbol{\theta}) := \frac{\exp\left\{(\mathbf{W}\mathbf{x} + \mathbf{b})_i\right\}}{\sum_{j \in \mathcal{Y}} \exp\left\{(\mathbf{W}\mathbf{x} + \mathbf{b})_j\right\}}, \tag{6}$$

where $\boldsymbol{\theta} := (\mathbf{W}, \mathbf{b})$ and i, j index classes in $\mathcal{Y}$. If $\mathbf{x}$ is d-dimensional and we are given m-class classification problem, then $\mathbf{W} \in \mathbb{R}^{m \times d}$ and $\mathbf{b} \in \mathbb{R}^m$. The case of regression is similar.

In Section 5.4, we show that if we apply LIME to interpret CEN with linear explanations, the local linear models inferred by LIME are guaranteed to recover the original CEN-generated explanations. In other words, linear explanations generated by CEN have similar properties, *e.g.*, local faithfulness (Ribeiro et al., 2016). However, we emphasize the key difference between LIME and CEN: the former regards explanation as a post-processing step (done after training) while the latter integrates explanation into the learning process.

4.2.2 Structured Explanations

While post-hoc methods, such as LIME, can easily generate local linear explanations for scalar outputs, using such methods for structured outputs is non-trivial. At the same time, CENs let us represent $\mathbb{P}(\mathbf{Y} \mid \mathbf{x}, \boldsymbol{\theta})$ using arbitrary graphical models. To be concrete, we consider the case where the targets are binary vectors, $\mathbf{y} \in \{0, 1\}^m$, and explanations are represented by CRFs (Lafferty et al., 2001) with linear potential functions.

The predictive distribution $\mathbb{P}(\mathbf{Y} \mid \mathbf{x}, \boldsymbol{\theta})$ represented by a CRF takes the following form:

$$\mathbb{P}(\mathbf{Y} \mid \mathbf{x}, \boldsymbol{\theta}) := \frac{1}{Z_{\boldsymbol{\theta}}(\mathbf{x})} \prod_{a \in \mathcal{A}} \Psi_a(\mathbf{y}_a, \mathbf{x}_a; \boldsymbol{\theta}) \tag{7}$$

where $Z_{\boldsymbol{\theta}}(\mathbf{x})$ is the normalizing constant and $a \in \mathcal{A}$ indexes subsets of variables in $\mathbf{x}$ and $\mathbf{y}$ that correspond to the factors:

$$\Psi_a(\mathbf{y}_a, \mathbf{x}_a; \boldsymbol{\theta}) := \exp\left\{\sum_{k=1}^{K} \boldsymbol{\theta}_{ak} f_{ak}(\mathbf{x}_a, \mathbf{y}_a)\right\}, \tag{8}$$

where $\{f_{ak}(\mathbf{x}_a, \mathbf{y}_a)\}_{k=1}^{K}$ is a collection of feature vectors associated with factor $\Psi_a(\mathbf{y}_a, \mathbf{x}_a; \boldsymbol{\theta})$. For interpretability purposes, we are interested in CRFs with feature vectors that are linear or bi-linear in $\mathbf{x}$ and $\mathbf{y}$. There is a variety of application-specific CRF models developed in the literature (*e.g.*, see Sutton et al., 2012). While in the following section, we discuss learning and inference more generally, in Section 6.3 we develop a CEN model with linear chain CRF explanations for solving survival analysis tasks.

4.3 Inference and Learning

CENs with deterministic encoders are convenient since the posterior, $\mathbb{P}\left(\boldsymbol{\theta} \mid \mathbf{y}, \mathbf{x}, \mathbf{c}\right)$, collapses to a point $\boldsymbol{\theta}^\star = \phi(\mathbf{c})$. Inference in such models is done in two steps: (1) first, compute $\theta^\star$, then (2) using $\theta^\star$ as parameters, compute the predictive distribution, $\mathbb{P}\left(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}^\star\right)$. To train the model, we can optimize its log likelihood on the training data. To make a prediction using a trained CEN model, we infer $\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} \mathbb{P}\left(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}^\star\right)$. For classification (and regression) computing predictions is straightforward. Below, we show how to compute predictions for CEN with CRF-based explanations.

4.3.1 Inference for CEN with Structured Explanations

Given a CRF model (7), we can make a prediction $\hat{\mathbf{y}}$ for inputs $(\mathbf{c}, \mathbf{x})$ by performing inference:

$$\hat{\mathbf{y}}(\boldsymbol{\theta}^\star) = \operatorname*{arg\,max}_{\mathbf{y} \in \mathcal{Y}} \mathbb{P}\left(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}^\star\right) = \operatorname*{arg\,max}_{\mathbf{y} \in \mathcal{Y}} \sum_{a=1}^{A} \sum_{k=1}^{K} \boldsymbol{\theta}^\star_{ak} f_{ak}(\mathbf{x}_a, \mathbf{y}_a) \tag{9}$$

Depending on the structure of the CRF model (*e.g.*, linear chain, tree-structured model, etc.), we could use different inference algorithms, such the Viterbi algorithm or variational inference, in order to solve (9) (see Ch. 4, Sutton et al., 2012, for an overview and examples). The key point here is that having $\mathbb{P}\left(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}^\star\right)$ or $\hat{\mathbf{y}}(\boldsymbol{\theta}^\star)$ computable in an (approximate) functional form, lets us construct different objective functions, *e.g.*, $\mathcal{L}(\{\mathbf{y}_i, \mathbf{x}_i, \mathbf{c}_i\}_{i=1}^{N}, \mathbf{w})$, and learn parameters of the CEN model end-to-end using gradient methods, which are standard in deep learning. In Section 6.3, we construct a specific objective function for survival analysis.

4.3.2 Learning via Likelihood Maximization and Posterior Regularization

In this paper, we use the negative log likelihood (NLL) objective for learning CEN models:

$$\mathcal{L}(\{\mathbf{y}_i, \mathbf{x}_i, \mathbf{c}_i\}_{i=1}^{N}, \mathbf{w}) := \frac{1}{N} \sum_{i=1}^{N} \log \mathbb{P}\left(\mathbf{y}_i \mid \mathbf{x}_i, \boldsymbol{\theta} = \phi_{\mathbf{w}}(\mathbf{c}_i)\right) \tag{10}$$

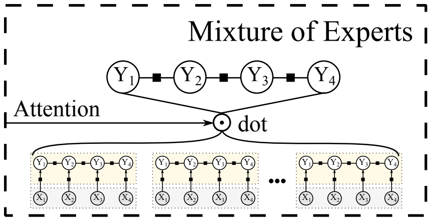
L_1, L_2, and other types of regularization imposed on $\boldsymbol{\theta}$ can be added to the objective (10). Such regularizers, as well as the dictionary constraint introduced in Section 4.1.2, can be seen as a form of *posterior regularization* (Ganchev et al., 2010) and are important for achieving the best performance and interpretability.

5. Analysis

In this section, we dive into the analysis of CEN as a class of probabilistic models. First, we mention special cases of CEN model class known in the literature, such as mixture-of-experts (Jacobs et al., 1991) and varying-coefficient models (Hastie and Tibshirani, 1993). Then, we discuss implications of the CEN structure, a potential failure mode of CEN with deterministic encoders and how to rectify it using conditional entropy regularization, and finally analyze relationship between CEN-generated and post-hoc explanations. Readers who are mostly interested in empirical properties and applications may skip this section.

5.1 Special Cases of CEN

Mixtures of Experts. So far, we have represented $\mathbb{P}_{\mathbf{w}}\left(\boldsymbol{\theta} \mid \mathbf{c}\right)$ by a delta-function centered around the output of the encoder. It is natural to extend $\mathbb{P}_{\mathbf{w}}\left(\boldsymbol{\theta} \mid \mathbf{c}\right)$ to a mixture of delta-distributions, in which case CENs recover the mixtures-of-experts model (MoE, Jacobs et al., 1991). To see this, let $\mathbf{D}$ be a dictionary of experts, and define $\mathbb{P}_{\mathbf{w},\mathbf{D}}\left(\boldsymbol{\theta} \mid \mathbf{c}\right) := \sum_{k=1}^{K} \mathbb{P}_{\mathbf{w}}\left(k \mid \mathbf{c}\right) \delta(\boldsymbol{\theta}, \boldsymbol{\theta}_k)$. The log-likelihood for CEN in such case is the same as for MoE:



$$\begin{aligned} \log \mathbb{P}_{\mathbf{w},\mathbf{D}}\left(\mathbf{y}_i \mid \mathbf{x}_i, \mathbf{c}_i\right) & \\ = \log \int \mathbb{P}\left(\mathbf{y}_i \vert \mathbf{x}_i, \boldsymbol{\theta}\right) \mathbb{P}_{\mathbf{w},\mathbf{D}}\left(\boldsymbol{\theta} \vert \mathbf{c}_i\right) d\boldsymbol{\theta} & \\ = \log \sum_{k=1}^{K} \mathbb{P}_{\mathbf{w}}\left(k \vert \mathbf{c}_i\right) \mathbb{P}\left(\mathbf{y}_i \vert \mathbf{x}_i, \boldsymbol{\theta}_k\right) & \end{aligned} \tag{11}$$

As in Section 4.1.2, $\mathbb{P}_{\mathbf{w}}\left(k \mid \mathbf{C}\right)$ is represented with a soft attention over the dictionary, $\mathbf{D}$, which is now used to combine predictions of the experts with parameters $\{\boldsymbol{\theta}_k\}_{k=1}^{K}$ instead of constructing a single context-specific explanation. Learning of MoE models is done either by optimizing the likelihood or via expectation maximization (EM). Note another difference between CEN and MoE is that the latter assumed that $\mathbf{c} \equiv \mathbf{x}$ and that both $\mathbb{P}\left(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}\right)$ and $\mathbb{P}\left(\boldsymbol{\theta} \mid \mathbf{c}\right)$ can be represented by arbitrary complex model classes, ignoring interpretability.

Varying-Coefficient Models. In statistics, there is a class of (generalized) regression models, called varying-coefficient models (VCMs, Hastie and Tibshirani, 1993), in which coefficients of linear models are allowed to be smooth deterministic functions of other variables (called the "effect modifiers"). Interestingly, the motivation for VCM was to increase flexibility of linear regression. In the original work, Hastie and Tibshirani (1993) focused on simple dynamic (temporal) linear models and on nonparametric estimation of the varying coefficients, where each coefficient depended on a different effect variable. CEN generalizes VCM by (i) allowing parameters, $\boldsymbol{\theta}$, to be random variables that depend on the context, $\mathbf{c}$, nondeterministically, (ii) letting the "effect modifiers" to be high-dimensional context variables (not just scalars), and (iii) modeling the effects using deep neural networks. In other words, CEN alleviates the limitations of VCM by leveraging the probabilistic graphical models and deep learning frameworks.

5.2 Implications of the structure of CENs

CENs represent the predictive distribution in a compound form (Lindsay, 1995):

$$\mathbb{P}\left(\mathbf{Y} \mid \mathbf{X}, \mathbf{C}\right) = \int \mathbb{P}\left(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\theta}\right) \mathbb{P}\left(\boldsymbol{\theta} \mid \mathbf{C}\right) d\boldsymbol{\theta}$$

and we assume that the data is generated according to $\mathbf{Y} \sim \mathbb{P}\left(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\theta}\right)$, $\boldsymbol{\theta} \sim \mathbb{P}\left(\boldsymbol{\theta} \mid \mathbf{C}\right)$. We would like to understand:

> *Can CEN represent any conditional distribution, $\mathbb{P}\left(\mathbf{Y} \mid \mathbf{X}, \mathbf{C}\right)$, when the class of explanations is limited (*e.g.*, to linear models)? If not, what are the limitations?*

Generally, CEN can be seen as a mixture of predictors. Such mixture models could be quite powerful as long as the mixing distribution, $\mathbb{P}\left(\boldsymbol{\theta} \mid \mathbf{C}\right)$, is rich enough. In fact, even a finite mixture exponential family regression models can approximate any smooth d-dimensional density at a rate $O(m^{-4/d})$ in the KL-distance (Jiang and Tanner, 1999). This result suggests that representing the predictive distribution with contextual mixtures should not limit the representational power of the model. However, there are two caveats:

(i) In practice, $\mathbb{P}\left(\boldsymbol{\theta} \mid \mathbf{C}\right)$ is limited, since we represent it either with a delta-function, a finite mixture, or a simple distribution parametrized by a deep network.

(ii) Classical predictive mixtures (including MoE) do not separate input features into two subsets, $\mathbf{c}$ and $\mathbf{x}$. We do this intentionally to produce explanations in terms of specific variables of interest that could be useful for interpretability or model diagnostics down the line. However, it could be the case that $\mathbf{x}$ contains only some limited information about $\mathbf{y}$, which could limit the predictive power of the full model.

To address point (i), we consider $\mathbb{P}\left(\boldsymbol{\theta} \mid \mathbf{c}\right)$ that fully factorizes over the dimensions of $\boldsymbol{\theta}$: $\mathbb{P}\left(\boldsymbol{\theta} \mid \mathbf{c}\right) = \prod_j \mathbb{P}\left(\theta_j \mid \mathbf{c}\right)$, and assume that explanations, $\mathbb{P}\left(\mathbf{Y} \mid \mathbf{x}, \boldsymbol{\theta}\right)$, also factorize according to some underlying graph, $\mathcal{G}_{\mathbf{Y}} = (\mathcal{V}_{\mathbf{Y}}, \mathcal{E}_{\mathbf{Y}})$. The following proposition shows that in such case $\mathbb{P}\left(\mathbf{Y} \mid \mathbf{x}, \mathbf{c}\right)$ inherits the factorization properties of the explanation class.

Proposition 1 *Let $\mathbb{P}\left(\boldsymbol{\theta} \mid \mathbf{c}\right) := \prod_j \mathbb{P}\left(\theta_j \mid \mathbf{c}\right)$ and let $\mathbb{P}\left(\mathbf{Y} \mid \mathbf{x}, \boldsymbol{\theta}\right)$ factorize according to some graph $\mathcal{G}_{\mathbf{Y}} = (\mathcal{V}_{\mathbf{Y}}, \mathcal{E}_{\mathbf{Y}})$. Then, $\mathbb{P}\left(\mathbf{Y} \mid \mathbf{x}, \mathbf{c}\right)$ defined by CEN with $\mathbb{P}\left(\boldsymbol{\theta} \mid \mathbf{c}\right)$ encoder and $\mathbb{P}\left(\mathbf{Y} \mid \mathbf{x}, \boldsymbol{\theta}\right)$ explanations also factorizes according to $\mathcal{G}$.*

Proof The statement directly follows from the definition of CEN (see Appendix A.1). ■

Remark 2 *All encoders, $\mathbb{P}\left(\boldsymbol{\theta} \mid \mathbf{c}\right)$, considered in this paper, including delta functions and their mixtures, fully factorize over the dimensions of $\boldsymbol{\theta}$.*

Remark 3 *The proposition has no implications for the case of scalar targets, $\mathbf{y}$. However, in case of structured prediction, regardless of how good the context encoder is, CEN will strictly assume the same set of independencies as given by the explanation class, $\mathbb{P}\left(\mathbf{Y} \mid \mathbf{x}, \boldsymbol{\theta}\right)$.*

As indicated in point (ii), CENs assume a fixed split of the input features into context, $\mathbf{c}$, and variables of interest, $\mathbf{x}$, which has interesting implications. Ideally, we would like $\mathbf{x}$ to be a good predictor of $\mathbf{y}$ in any context $\mathbf{c}$. For instance, following our motivation example (see Figure 1), if $\mathbf{c}$ distinguishes between urban and rural areas, $\mathbf{x}$ must encode enough information for predicting poverty *within* urban or rural neighborhoods. However, since the variables of interest are often manually selected (*e.g.*, by a domain expert) and limited, we may encounter the following (not mutually exclusive) situations:

(a) $\mathbf{c}$ may happen to be a strong predictor of $\mathbf{y}$ and already contain information available in $\mathbf{x}$ (*e.g.*, it is the case when $\mathbf{x}$ is derived from $\mathbf{c}$).

(b) $\mathbf{x}$ may happen to be a poor predictor of $\mathbf{y}$, even within the context specified by $\mathbf{c}$.

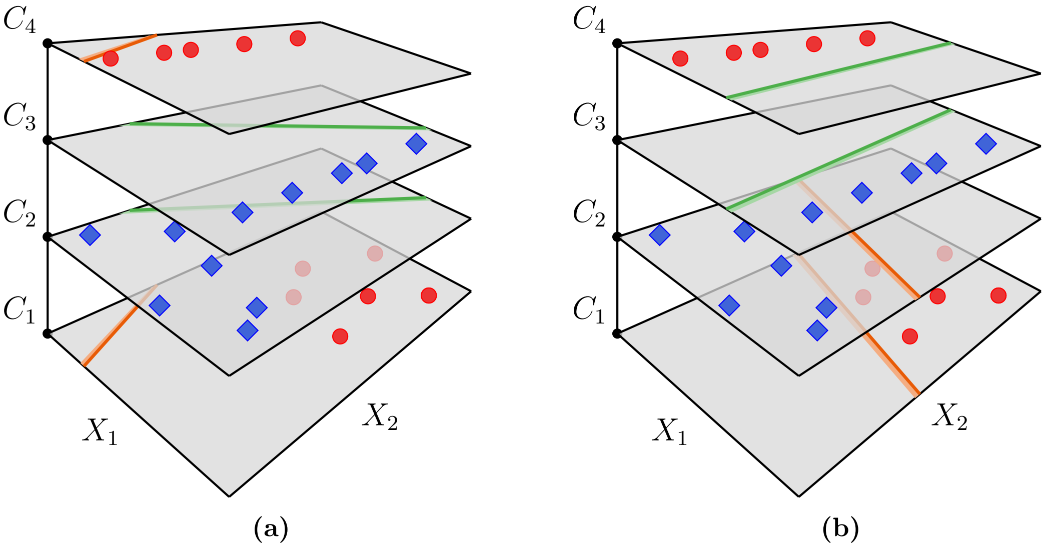
In both cases, CEN may learn to ignore $\mathbf{x}$, leading to essentially meaningless explanations. In the next section, we show that, if (a) is the case, regularization can help eliminate such behavior. Additionally, if (b) is the case, *i.e.*, $\mathbf{x}$ are bad features for predicting $\mathbf{y}$ (and for seeking explanation in terms of these features), CEN must indicate that. It turns out that the accuracy of CEN depends on the quality of $\mathbf{x}$, as empirically shown in Section 6.2.2.

5.3 Conditional Entropy Regularization

CEN has a failure mode: when the context $\mathbf{c}$ is highly predictive of the targets $\mathbf{y}$ and the encoder is represented by a powerful model, CEN may learn to rely entirely on the context variables. In such case, the encoder would generate spurious explanations, one for each target class. For example, for binary targets, $\mathbf{y} \in \{0, 1\}$, CEN may learn to always map $\mathbf{c}$ to either $\boldsymbol{\theta}_0$ or $\boldsymbol{\theta}_1$ when $\mathbf{y}$ is 0 or 1, respectively. In other words, $\boldsymbol{\theta}$ (as a function of $\mathbf{c}$) would become highly predictive of $\mathbf{y}$ on its own, and hence $\mathbb{P}\left(\mathbf{Y} \mid \mathbf{x}, \boldsymbol{\theta}\right) \approx \mathbb{P}\left(\mathbf{Y} \mid \boldsymbol{\theta}\right)$, *i.e.*, $\mathbf{Y}$ would be (approximately) conditionally independent of $\mathbf{X}$ given $\boldsymbol{\theta}$. This is problematic from the interpretation point of view since explanations would become spurious, *i.e.*, no longer used to make predictions from the variables of interest.

Note that such a model would be accurate only when the generated $\boldsymbol{\theta}$ is always highly predictive of $\mathbf{Y}$, *i.e.*, when the conditional entropy $\mathcal{H}(\mathbf{Y} \mid \boldsymbol{\theta})$ is low. Following this observation, we propose to regularize the model by approximately *maximizing* $\mathcal{H}(\mathbf{Y} \mid \boldsymbol{\theta})$. For a CEN with a deterministic encoder (Sections 4.1.1 and 4.1.2), we can compute an unbiased estimate of $\mathcal{H}(\mathbf{Y} \mid \boldsymbol{\theta})$ given a mini-batch of samples from the dataset as follows:

$$\begin{aligned}
\mathcal{H}(\mathbf{Y} \mid \boldsymbol{\theta}) &= \int \mathbb{P}\left(\mathbf{y}, \boldsymbol{\theta}\right) \log \mathbb{P}\left(\mathbf{y} \mid \boldsymbol{\theta}\right) d\mathbf{y} d\boldsymbol{\theta} && (12) \\
&= \mathbb{E}_{(\mathbf{c},\mathbf{x}) \sim \mathbb{P}(\mathbf{c},\mathbf{x})} \left[\int \mathbb{P}\left(\mathbf{y} \mid \mathbf{x}, \phi(\mathbf{c})\right) \log \mathbb{E}_{\mathbf{x}' \sim \mathbb{P}(\mathbf{x})} \left[\mathbb{P}\left(\mathbf{y} \mid \mathbf{x}', \phi(\mathbf{c})\right) \right] d\mathbf{y} \right] && (13) \\
&\approx \frac{1}{|B|} \sum_{i \in B} \int \mathbb{P}\left(\mathbf{y} \mid \mathbf{x}_i, \phi(\mathbf{c}_i)\right) \log \left[\frac{1}{|B|} \sum_{j \in B} \mathbb{P}\left(\mathbf{y} \mid \mathbf{x}_j, \phi(\mathbf{c}_i)\right) \right] d\mathbf{y} && (14)
\end{aligned}$$



(a) **(b)**

Figure 4: A toy synthetic dataset and two linear explanations (green and orange) produced by a CEN model trained (a) with no regularization or (b) with conditional entropy regularization.

In the given expressions, elements of B index training samples (*e.g.*, B represents a mini-batch), (13) is obtained by using the definition of CEN and marginalizing out $\boldsymbol{\theta}$, (14) is a stochastic estimate that approximates expectations with a mini-batch of samples. Intuitively, if the predictions are accurate while $\mathcal{H}(\mathbf{Y} \mid \boldsymbol{\theta})$ is high, we can be sure that CEN learned to generate contextual $\boldsymbol{\theta}$'s that are uncorrelated with the targets but result into accurate conditional models, $\mathbb{P}\left(\mathbf{Y} \mid \mathbf{x}, \boldsymbol{\theta}\right)$.

An illustration on synthetic data. To illustrate the problem, we consider a toy synthetic 3D dataset with 2 classes that are not separable linearly (Figure 4). The coordinates along the vertical axis C correspond to different contexts, and (X_1, X_2) represent variables of interest. Note we can perfectly distinguish between the two classes by using only the context information. CEN with a dictionary of size 2 learns to select one of the two linear explanations for each of the contexts. When trained without regularization (Figure 4a), selected explanations are spurious hyperplanes since each of them is used for points of a single class only. Adding entropy regularization (Figure 4b) makes CEN select hyperplanes that meaningfully distinguish between the classes within different contexts.

Quantifying contribution of the explanations. Starting from the introduction, we have argued that explanations are meaningful when they are used for prediction. In other words, we would like explanations have a non-zero contribution to the overall accuracy of the model. The following theorem quantifies the contribution of explanations to the predictive performance of entropy-regularized CEN.

Proposition 4 *Let CEN with linear explanations have the expected predictive accuracy*

$$\mathbb{E}_{\mathbf{X},\boldsymbol{\theta}\sim\mathbb{P}(\mathbf{X},\boldsymbol{\theta})}\left[\mathbb{P}\left(\hat{\mathbf{Y}} = \mathbf{Y} \mid \mathbf{X}, \boldsymbol{\theta}\right)\right] \geq 1 - \varepsilon, \tag{15}$$

where $\varepsilon \in (0, 1)$ is small. Let also the conditional entropy be $\mathcal{H}(\mathbf{Y} \mid \boldsymbol{\theta}) \geq \delta$ for some $\delta \geq 0$. Then, the expected contribution of the explanations to the predictive performance of CEN is given by the following lower bound:

$$\mathbb{E}_{\mathbf{X},\boldsymbol{\theta}\sim\mathbb{P}(\mathbf{X},\boldsymbol{\theta})}\left[\mathbb{P}\left(\hat{\mathbf{Y}} = \mathbf{Y} \mid \mathbf{X}, \boldsymbol{\theta}\right) - \mathbb{P}\left(\hat{\mathbf{Y}} = \mathbf{Y} \mid \boldsymbol{\theta}\right)\right] \geq \frac{\delta - 1}{\log |\mathcal{Y}|} - \varepsilon, \tag{16}$$

where $|\mathcal{Y}|$ denotes the cardinality of the target space.

Proof The statement follows from Fano's inequality. For details, see Appendix A.2. ■

Remark 5 *The proposition states that explanations are meaningful (as contextual models) only when CEN is accurate (*i.e.*, the expected predictive error is less than ε) and the conditional entropy $\mathcal{H}(\mathbf{Y} \mid \boldsymbol{\theta})$ is high. High accuracy and low entropy imply spurious explanations. Low accuracy and high entropy imply that $\mathbf{x}$ features are not predictive of $\mathbf{y}$ within the class of explanations, suggesting to reconsider our modeling assumptions.*

5.4 CEN-generated vs. Post-hoc Explanations

In this section, we analyze the relationship between CEN-generated and LIME-generated *post-hoc* explanations. Given a trained CEN, we can use LIME to approximate its decision boundary and compare the explanations produced by both methods. The question we ask:

> *How does the local approximation, $\hat{\boldsymbol{\theta}}$, relate to the actual explanation, $\boldsymbol{\theta}^\star$, generated and used by CEN to make a prediction in the first place?*

For the case of binary[5] classification, it turns out that when the context encoder is deterministic and the space of explanations is *linear*, local approximations, $\hat{\boldsymbol{\theta}}$, obtained by solving (1) recover the original CEN-generated explanations, $\boldsymbol{\theta}^\star$. Formally, our result is stated in the following theorem.

Theorem 6 *Let the explanations and the local approximations be in the class of linear models, $\mathbb{P}(Y = 1 \mid \mathbf{x}, \boldsymbol{\theta}) \propto \exp\left\{\mathbf{x}^\top \boldsymbol{\theta}\right\}$. Further, let the encoder be L-Lipschitz and pick a sampling distribution, $\pi_{\mathbf{x},\mathbf{c}}$, that concentrates around the point $(\mathbf{x}, \mathbf{c})$, such that $\mathbb{P}_{\pi_{\mathbf{x},\mathbf{c}}}(\|\mathbf{z}' - \mathbf{z}\| > t) < \varepsilon(t)$, where $\mathbf{z} := (\mathbf{x}, \mathbf{c})$ and $\varepsilon(t) \to 0$ as $t \to \infty$. Then, if the loss function is defined as*

$$\mathcal{L} = \frac{1}{K}\sum_{k=1}^{K}\left(\operatorname{logit}\left\{\mathbb{P}\left(Y = 1 \mid \mathbf{x}_k, \mathbf{c}_k\right)\right\} - \operatorname{logit}\left\{\mathbb{P}\left(Y = 1 \mid \mathbf{x}_k, \boldsymbol{\theta}\right)\right\}\right)^2, \ (\mathbf{x}_k, \mathbf{c}_k) \sim \pi_{\mathbf{x},\mathbf{c}}, \tag{17}$$

the solution of (1) concentrates around $\boldsymbol{\theta}^\star$ as $\mathbb{P}_{\pi_{\mathbf{x},\mathbf{c}}}\left(\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\| > t\right) \leq \delta_{K,L}(t)$, $\delta_{K,L} \underset{t\to\infty}{\longrightarrow} 0$.

---

[5] Analysis of the multi-class case can be reduced to the binary in the one-vs-all fashion.

Intuitively, by sampling from a distribution sharply concentrated around $(\mathbf{x}, \mathbf{c})$, we ensure that $\hat{\boldsymbol{\theta}}$ will recover $\boldsymbol{\theta}^\star$ with high probability.

This result establishes an equivalence between the explanations generated by CEN and those produced by LIME post-hoc *when approximating CEN*. Note that when LIME is applied to a model other than CEN, equivalence between explanations is not guaranteed. Moreover, as we further show experimentally, certain conditions such as incomplete or noisy interpretable features may lead to LIME producing inconsistent and erroneous explanations. The proof of the theorem is given in Appendix A.3.

6. Case Studies

In this section, we move to a number of case studies where we empirically analyze properties of the proposed CEN framework on classification and survival analysis tasks. In particular, we evaluate CEN with linear explanations on a few classification tasks that involve different data modalities of the context (*e.g.*, images or text). For survival prediction, we design CEN architectures with structured explanations, derive learning and inference algorithms, and showcase our models on problems from the healthcare domain.

6.1 Solving Classification using CEN with Linear Explanations

We start by examining the properties of CEN with linear explanations (Table 1) on a few classification tasks. Our experiments are designed to answer the following questions:

(i) When explanation is a part of the learning and prediction process, how does that affect performance of the final predictive model *quantitatively*?

(ii) *Qualitatively*, what kind of insight can we gain by inspecting explanations?

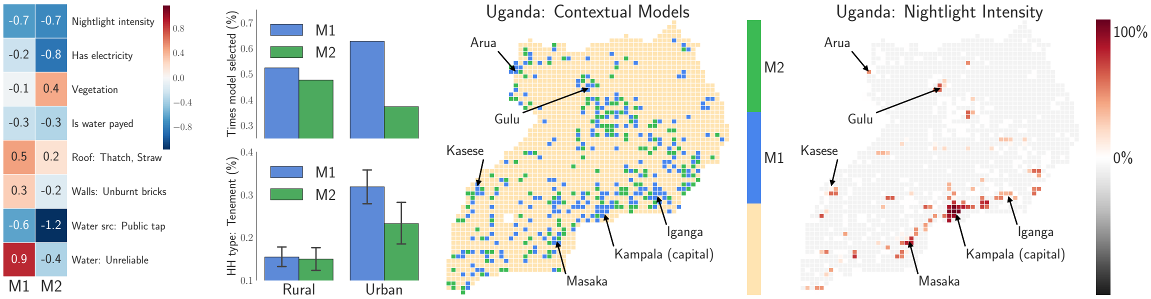
(iii) Finally, we analyze *consistency* of linear explanations generated by CEN versus those generated using LIME (Ribeiro et al., 2016), a popular post-hoc method.

Details on our experimental setup, all hyperparameters, and training procedures are given in the tables in Appendix B.3.

6.1.1 Poverty Prediction

We consider the problem of poverty prediction for household clusters in Uganda from satellite imagery and survey data. Each household cluster is represented by a collection of 400 × 400 satellite images (used as the context) and 65 categorical variables from living standards measurement survey (used as the interpretable attributes). The task is binary classification of the households into poor and not poor.

We follow the original study of Jean et al. (2016) and use a `VGG-F` network (pre-trained on nightlight intensity prediction) to compute 4096-dimensional embeddings of the satellite images on top of which we build contextual models. Note that this datasets is fairly small (500/142 train/test points), and so we keep the `VGG-F` part frozen to avoid overfitting.



(a) (b) (c) (d)

Figure 5: Qualitative results for the Satellite dataset: (a) Weights given to a subset of features by the two models (M1 and M2) discovered by CEN. (b) How frequently M1 and M2 are selected for areas marked rural or urban (top) and the average proportion of Tenement-type households in an urban/rural area for which M1 or M2 was selected. (c) M1 and M2 models selected for different areas on the Uganda map. M1 tends to be selected for more urbanized areas while M2 is picked for the rest. (d) Nightlight intensity of different areas of Uganda.

Models. For baselines, we use logistic regression (LR) and multi-layer perceptrons (MLP) with 1 hidden layer. The LR baseline uses either `VGG-F` embeddings ($\mathrm{LR}_{\texttt{emb}}$) or the categorical attributes ($\mathrm{LR}_{\texttt{att}}$) as inputs. The input of the MLP baseline is concatenated `VGG-F` embeddings and categorical attributes. Context encoder of the CEN model uses `VGG-F` to process images, followed by an attention layer over a dictionary of 16 trainable linear explanations defined over the categorical features (Figure 3). Finally, we evaluate a mixture-of-experts (MoE) model of the same architecture as CEN, since it is a special case (see Section 5.1). Both CEN and MoE are trained with the dictionary constraint and L_1 regularization to encourage sparse explanations. Details on the architectures and training are given in Table 7b.

Table 2: Performance of the models on the poverty prediction.

	Acc (%)	**AUC (%)**
$\mathrm{LR}_{\texttt{emb}}$	62.5%	68.1
$\mathrm{LR}_{\texttt{att}}$	75.7%	82.2
MLP	77.4%	78.7
$\mathrm{MoE}_{\texttt{att}}$	77.9%	**85.4**
$\mathrm{CEN}_{\texttt{att}}$	**81.5%**	84.2

Performance. The results are presented in Table 2. Both in terms of accuracy and AUC, CEN models outperform both simple logistic regression and vanilla MLP. Even though the results suggest that categorical features are better predictors of poverty than `VGG-F` embeddings of images, note that using embeddings to *contextualize* linear models reduces the error. This indicates that *different* linear models are optimal in different contexts.

Qualitative analysis. We discovered that, on this task, CEN encoder tends to sharply select one of the two explanations (M1 and M2) for different household clusters in Uganda (Figure 5a). In the survey data, each household cluster is marked as either urban or rural. We notice that, conditional on a satellite image, CEN tends to pick M1 for urban areas and M2 for rural (Figure 5b). Notice that different explanations weigh categorical features, such

Table 3: Sentiment classification error rate on IMDB dataset. It is interesting to note that CENs gets state of the art performance by only using supervised data (25 thousand labeled reviews) while Miyato et al. (2016) obtain their result by using additional 50K unlabelled reviews.

Method	**Reference**	**Error**
BoW (bnc)	Maas et al. (2011a)	12.20%
BoW (bΔtc)	Maas et al. (2011a)	11.77%
LDA	Maas et al. (2011a)	32.58%
Full + BoW	Maas et al. (2011a)	11.67%
Full + Unlabelled + BoW	Maas et al. (2011a)	11.11%
WRRBM	Dahl et al. (2012)	12.58%
WRRBM + BoW	Dahl et al. (2012)	10.77%
MNB-uni	Wang and Manning (2012)	16.45%
MNB-bi	Wang and Manning (2012)	13.41%
SVM-uni	Wang and Manning (2012)	13.05%
SVM-bi	Wang and Manning (2012)	10.84%
NBSVM-uni	Wang and Manning (2012)	11.71%
NBSVM-bi	Wang and Manning (2012)	8.78%
NBSVM-bi	Wang and Manning (2012)	8.78%
seq2-brown-CNN	Johnson and Zhang (2014)	14.70%
Paragraph Vector	Le and Mikolov (2014)	7.42%
SA-LSTM with joint training	Dai and Le (2015)	14.70%
LSTM with tuning and dropout	Dai and Le (2015)	13.50%
LSTM initialized with word2vec embeddings	Dai and Le (2015)	10.00%
SA-LSTM with linear gain	Dai and Le (2015)	9.17%
LM-TM	Dai and Le (2015)	7.64%
SA-LSTM	Dai and Le (2015)	7.24%
Virtual Adversarial	Miyato et al. (2016)	$\mathbf{5.94 \pm 0.12}$%
TopicRNN	Dieng et al. (2017)	6.28%
CEN-`bow`		$\mathbf{5.92 \pm 0.05}$ %
CEN-`tpc`		$\mathbf{6.25 \pm 0.09}$ %

as *reliability of the water source* or the *proportion of houses with walls made of unburnt brick*, quite differently. When visualized on the map, we see that CEN selects M1 more frequently around the major city areas, which also correlates with high nightlight intensity in those areas (Figures 5c,5d). We also estimate the approximate conditional entropy of the binary targets (poor vs. not poor) given the selected model and find: $\mathcal{H}(\mathbf{Y} \mid \boldsymbol{\theta} = \text{M1}) \approx 77\%$ and $\mathcal{H}(\mathbf{Y} \mid \boldsymbol{\theta} = \text{M2}) \approx 72\%$. High performance of the model along with high conditional entropy makes us confident in the produced explanations (see Section 5.3) and allows us to draw conclusions about *what causes the model* to classify certain households in different neighborhoods as poor in terms of interpretable categorical variables.

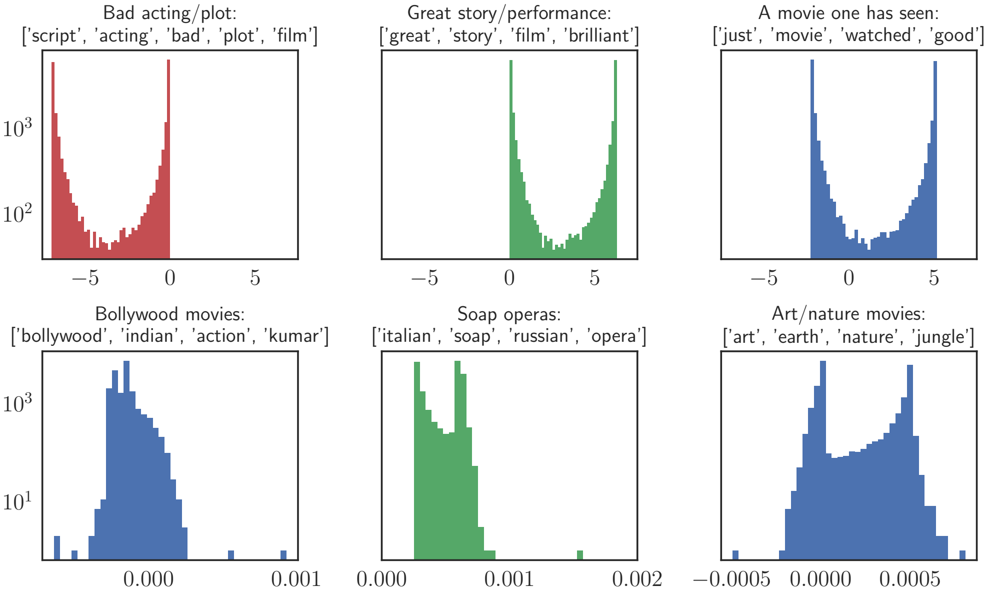


Figure 6: Histograms of test weights assigned by CEN to 6 topics: acting- and plot-related topics (upper charts), genre topics (bottom charts).

6.1.2 Sentiment Analysis

The next problem we consider is sentiment prediction of IMDB reviews (Maas et al., 2011b). The reviews are given in the form of English text (sequences of words) and the sentiment labels are binary (good/bad movie). This dataset has 25k labelled reviews used for training and validation and 25k labelled reviews that are held out for test. The data also contains an additional set of 50k *unlabelled* reviews that are used by some models in the literature. We emphasize that we do *not* use the unlabelled part of the data for training CENs.

Models. Following Johnson and Zhang (2016), we use a bi-directional LSTM with max-pooling as our baseline that predicts sentiment directly from text sequences. The same architecture is used as the context encoder in CEN that produces parameters for linear explanations. The explanations are applied to either (a) a bag-of-words (BoW) features (with a vocabulary limited to 5,000 most frequent words) or (b) a 100-dimensional topic representation produced by a separately trained off-the-shelf topic model (Blei et al., 2003).

Performance. Comparison of CEN with other models from the literature is given in Table 3. Not only CEN achieves a near state-of-the-art accuracy on this dataset, we emphasize that we do *not* use any unlabeled data when training our models. This indicates that the inductive biases provided by the architecture lead to a more significant performance improvement that many of the semi-supervised training methods on this dataset.

Table 4: Prediction error of the models on image classification tasks (averaged over 5 runs; the std. are on the order of the least significant digit). The subscripts denote the features on which the linear models are built: pixels (`pxl`), HOG (`hog`).

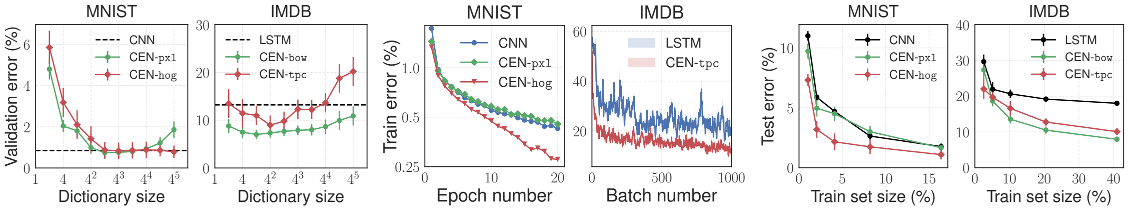
MNIST							**CIFAR10**						
$\text{LR}_{\texttt{pxl}}$	$\text{LR}_{\texttt{hog}}$	CNN	$\text{MoE}_{\texttt{pxl}}$	$\text{MoE}_{\texttt{hog}}$	$\text{CEN}_{\texttt{pxl}}$	$\text{CEN}_{\texttt{hog}}$	$\text{LR}_{\texttt{pxl}}$	$\text{LR}_{\texttt{hog}}$	VGG	$\text{MoE}_{\texttt{pxl}}$	$\text{MoE}_{\texttt{hog}}$	$\text{CEN}_{\texttt{pxl}}$	$\text{CEN}_{\texttt{hog}}$
8.00	2.98	**0.75**	1.23	1.10	**0.76**	**0.73**	60.1	48.6	9.4	13.0	11.7	9.6	**9.2**

Qualitative analysis. After training CEN-`tpc` with linear explanations in terms of topics on the IMDB dataset, we generate explanations for each test example and visualize histograms of the weights assigned by the explanations to the 6 selected topics in Figure 6. The 3 topics in the top row are acting- and plot-related (and intuitively have positive, negative, or neutral connotation), while the 3 topics in the bottom are related to particular genre of the movies. Note that acting-related topics turn out to be bimodal, *i.e.*, contributing either positively, negatively, or neutrally to the sentiment prediction in different contexts. CEN assigns a high negative weight to the topic related to "bad acting/plot" and a high positive weight to "great story/performance" in most of the contexts (and treats those neutrally conditional on some of the reviews). Interestingly, genre-related topics almost always have a negligible contribution to the sentiment which indicates that the learned model does not have any particular bias towards or against a given genre.

Figure 14 in Appendix visualizes the full dictionary of size 16 learned by CEN-`tpc`. Each column corresponds to a dictionary atom that represents a typical explanation pattern that CEN attends to before making a prediction. By inspecting the dictionary, we can find interesting patterns. For instance, atoms 5 and 11 assign inverse weights to the following topics denoted by the top 4 words: `[kid, child, disney, family]` and `[sexual, violence, nudity, sex]` (*i.e.*, good family movies must not be violent and vice versa). Depending on the context of the review, CEN may select one of these patterns to predict the sentiment. Note that these two topics are negatively correlated across all dictionary elements, which again is quite intuitive.

6.1.3 Image Classification

For the purpose of completeness, we also provide results on two classical image datasets: MNIST and CIFAR-10. For CEN, full images are used as the context; to imitate high-level features, we use (a) the original images cubically downscaled to 20×20 pixels, gray-scaled and normalized, and (b) HOG descriptors computed using 3×3 blocks (Dalal and Triggs, 2005). For each task, we use linear regression and vanilla convolutional networks as baselines (a small convnet for MNIST and `VGG-16` for CIFAR-10). The results are reported in Table 4. CENs are competitive with the baselines and do not exhibit deterioration in performance. Visualization and analysis of the learned explanations is given in Appendix B.2 and the details on the architectures, hyperparameters, and training are given in Appendix B.3



(a)

(b)

(c)

Figure 7: (a) Validation error vs. dictionary size. (b) Training error vs. iteration (epoch or batch) for baselines and CEN. (c) Test error for models trained on random subsets of data of different sizes.

6.2 Properties of Explanations

In this section, we look at the explanations from the regularization and consistency point of view. As we show next, prediction via explanation not only has a strong regularization effect, but also always produces consistent locally linear models.

6.2.1 Explanations as a Regularizer

By controlling the dictionary size, we can control the expressivity of the model class specified by CEN. For example, when the dictionary size is 1, CEN becomes equivalent to a linear model. For larger dictionaries, CEN becomes as flexible as a deep network (Figure 7a). Adding a small sparsity penalty to each element of the dictionary (between 10^{-6} and 10^{-3}, see Appendix B.3) helps to avoid overfitting for very large dictionary sizes, so that the model learns to use only a few dictionary atoms for prediction while shrinking the rest to zero.

If explanations can act as a proper regularizer, we must observe improved sample complexity of the model. To verify this, we trained CEN models on subsets of data (size varied between 1% and 20% for MNIST and 2% and 40% for IMDB) and then evaluated accuracy on the validation set. As seen from the error reported in Figure 7c, CENs require much fewer samples to attain a near top accuracy (as if trained on the full dataset). Finally, we also observe that CEN models tend to converge faster (Figure 7b) which indicates that prediction via explanation improves the geometry of the optimization problem.

6.2.2 Consistency of Explanations

While regularization is a useful aspect, the main use case for explanations is model diagnostics. Linear explanations assign weights to the interpretable features, $\mathbf{X}$, and hence their quality depends on the way we select these features. In this section, we evaluate explanations generated by CEN or a post-hoc method (LIME). In particular, we consider two cases where (a) the features are corrupted with additive noise, and (b) the selected features are incomplete. For analysis, we use MNIST and IMDB datasets. Our key question is:

Can we trust the explanations built on noisy or incomplete features?

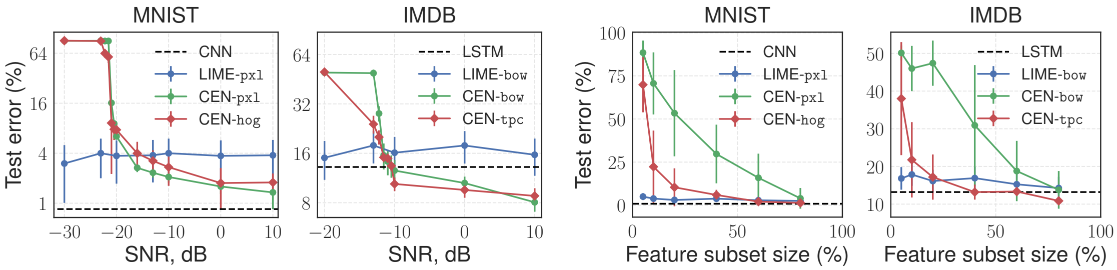


Figure 8: The effect of feature quality on explanations. (a) Explanation test error vs. the level of the noise added to the interpretable features. (b) Explanation test error vs. the total number of interpretable features.

The effect of noisy features. In this experiment, we inject noise[6] into the features $\mathbf{X}$ and ask LIME and CEN to fit explanations to the corrupted features. Note that after injecting noise, each data point has a noiseless representation $\mathbf{C}$ and noisy $\mathbf{X}$. LIME constructs explanations by approximating the decision boundary of the baseline model trained to predict $\mathbf{Y}$ from $\mathbf{C}$ features only. CEN is trained to construct explanations given $\mathbf{C}$ and then make predictions by applying explanations to $\mathbf{X}$. The predictive performance of the produced explanations on noisy features is given on Figure 8. Since baselines take only $\mathbf{C}$ as inputs, their performance stays the same and, regardless of the noise level, LIME "successfully" overfits explanations—it is able to almost perfectly approximate the decision boundary of the baselines essentially using pure noise. On the other hand, performance of CEN gets worse with the increasing noise level indicating that the model fails to learn when the selected interpretable representation is of low quality.

The effect of feature selection. Here, we use the same setup, but instead of injecting noise into $\mathbf{X}$, we construct $\mathbf{X}$ by randomly subsampling a set of dimensions. Figure 8 demonstrates the result. While performance of CENs degrades proportionally to the size of $\mathbf{X}$, we see that, again, LIME is able to fit explanations to the decision boundary of the original models despite the loss of information.

These two experiments indicate a major drawback of explaining predictions post-hoc: when constructed on poor, noisy, or incomplete features, such explanations can overfit the decision boundary of a predictor and are likely to be meaningless or misleading. For example, predictions of a perfectly valid model might end up getting absurd explanations which is unacceptable from the decision support point of view. On the other hand, if we use CEN to generate explanations, high predictive performance would indicate presence of a meaningful signal the selected interpretable features and explanations.

---

[6] We use Gaussian noise with zero mean and select variance for each signal-to-noise ratio level appropriately.

6.3 Solving Survival Analysis using CEN with Structured Explanations

In this final case study, we design CENs with structured explanations for survival prediction. We provide some general background on survival analysis and the structured prediction approach proposed by Yu et al. (2011), then introduce CENs with CRF-based explanations for survival analysis, and conclude with experimental results on two public datasets from the healthcare domain.

6.3.1 Background on Survival Analysis via Structured Prediction

In survival time prediction, our goal is to estimate the risk and occurrence time of an undesirable event in the future (*e.g.*, death of a patient, earthquake, hard drive failure, customer turnover, etc.). A common approach is to model the *survival time*, T, either for a population (i.e., average survival time) or for each instance. Classical approaches, such as Aalen additive hazard (Aalen, 1989) and Cox proportional hazard (Cox, 1972) models, view survival analysis as continuous time prediction and hence a regression problem.

Alternatively, the time can be discretized into intervals (*e.g.*, days, weeks, etc.), and the survival time prediction can be converted into a multi-task classification problem (Efron, 1988). Taking this approach one step further, Yu et al. (2011) noticed that the output space of such a multitask classifier is structured in a particular way, and proposed a model called *sequence of dependent regressors*. The model is essentially a CRF with a particular structure of the pairwise potentials between the labels. We introduce the setup in our notation below.

Let the data instances be represented by tuples $(\mathbf{c}, \mathbf{x}, \mathbf{y})$, where targets are now sequences of m binary variables, $\mathbf{y} := (y^1, \dots, y^m)$, that indicate occurrence of an event at the corresponding time intervals.[7] If the event occurred at time $t \in [t_i, t_{i+1})$, then $y^j = 0$, $\forall j \leq i$ and $y^k = 1$, $\forall k > i$. If the event was *censored* (*i.e.*, we lack information for times after t), we represent targets $(y^{i+1}, \dots, y^m)$ with latent variables. Importantly, only $m + 1$ sequences are valid under these conditions, *i.e.*, assigned non-zero probability by the model. This suggests a linear CRF model defined as follows:

$$\mathbb{P}\left(\mathbf{Y} = (y^1, y^2, \dots, y^m) \mid \mathbf{x}, \boldsymbol{\theta}^{1:m}\right) \propto \exp\left\{\sum_{t=1}^{m} y^i(\mathbf{x}^\top \boldsymbol{\theta}^t) + \omega(y^t, y^{t+1})\right\} \tag{18}$$

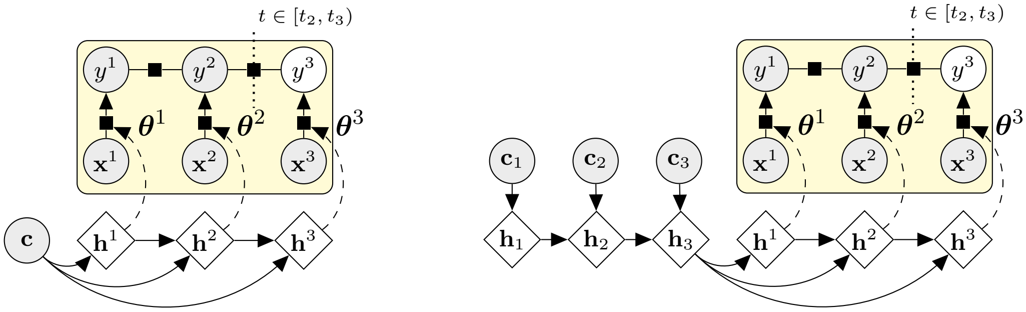
The potentials between $\mathbf{x}$ and $y^{1:m}$ are linear functions parameterized by $\boldsymbol{\theta}^{1:m}$. The pairwise potentials between targets, $\omega(y_i, y_{i+1})$, ensure that non-permissible configurations where $(y_i = 1, y_{i+1} = 0)$ for some $i \in \{0, \dots, m-1\}$ are improbable (*i.e.*, $\omega(1, 0) = -\infty$ and $\omega(0, 0) = \omega_{00}$, $\omega(0, 1) = \omega_{01}$, $\omega(1, 1) = \omega_{10}$ are learnable parameters).

To train the model, Yu et al. (2011) optimize the following objective:

$$\min_{\boldsymbol{\Theta}} C_1 \sum_{t=1}^{m} \|\theta^t\|^2 + C_2 \sum_{t=1}^{m-1} \|\theta^{t+1} - \theta^t\|^2 - \log \mathcal{L}(\mathbf{Y}, \mathbf{X}; \boldsymbol{\theta}^{1:m}) \tag{19}$$

---

[7] We assume that the occurrence time is lower bounded by $t_0 = 0$, upper bounded by some $t_m = T$, and discretized into intervals $[t_i, t_{i+1})$, where $i \in \{0, \dots, m-1\}$.



(a) Architecture used for SUPPORT2. **(b)** Architecture used for PhysioNet.

Figure 9: CEN architectures used in our survival analysis experiments. Context encoders were (a) single hidden layer MLP and (b) LSTM. Encoders produced inputs for another LSTM over the output time intervals (denoted with $\mathbf{h}^1$, $\mathbf{h}^2$, $\mathbf{h}^3$ hidden states respectively).

where the first two terms are regularization and the last term is the log of the likelihood:

$$\mathcal{L}(\mathbf{Y}, \mathbf{X}; \boldsymbol{\Theta}) = \sum_{i \in \mathrm{NC}} \mathbb{P}\left(T = t_i \mid \mathbf{x}_i, \boldsymbol{\Theta}\right) + \sum_{j \in \mathrm{C}} \mathbb{P}\left(T > t_j \mid \mathbf{x}_j, \boldsymbol{\Theta}\right) \tag{20}$$

where NC denotes the set of non-censored instances (for which we know the outcome times, t_i) and C is the set of censored inputs (for which we only know the censorship times, t_j). The likelihood of an uncensored and a censored event at time $t \in [t_j, t_{j+1})$ are as follows:

$$\begin{aligned} \mathbb{P}\left(T = t \mid \mathbf{x}, \boldsymbol{\theta}^{1:m}\right) &= \exp\left\{\sum_{i=j}^{m} \mathbf{x}^\top \boldsymbol{\theta}^i\right\} \Big/ \sum_{k=0}^{m} \exp\left\{\sum_{i=k+1}^{m} \mathbf{x}^\top \boldsymbol{\theta}^i\right\} \\ \mathbb{P}\left(T \geq t \mid \mathbf{x}, \boldsymbol{\theta}^{1:m}\right) &= \sum_{k=j+1}^{m} \exp\left\{\sum_{i=k+1}^{m} \mathbf{x}^\top \boldsymbol{\theta}^i\right\} \Big/ \sum_{k=0}^{m} \exp\left\{\sum_{i=k+1}^{m} \mathbf{x}^\top \boldsymbol{\theta}^i\right\} \end{aligned} \tag{21}$$

6.3.2 CEN with Structured Explanations for Survival Analysis

To construct CEN for survival analysis, we follow the structured survival prediction setup described in the previous section. We define CEN with linear CRF explanations as follows:

$$\begin{aligned} &\boldsymbol{\theta}^t \sim \mathbb{P}_{\mathbf{w}}\left(\boldsymbol{\theta}^t \mid \mathbf{c}\right), \mathbf{y} \sim \mathbb{P}\left(\mathbf{Y} \mid \mathbf{x}, \boldsymbol{\theta}^{1:m}\right), \\ &\mathbb{P}\left(\mathbf{Y} = (y^1, y^2, \ldots, y^m) \mid \mathbf{x}, \boldsymbol{\theta}^{1:m}\right) \propto \exp\left\{\sum_{t=1}^{m} y^i (\mathbf{x}^\top \boldsymbol{\theta}^t) + \omega(y^t, y^{t+1})\right\}, \\ &\mathbb{P}_{\mathbf{w}}\left(\boldsymbol{\theta}^t \mid \mathbf{c}\right) := \delta(\boldsymbol{\theta}^t, \phi^t_{\mathbf{w},\mathbf{D}}(\mathbf{c})),\ \phi^t_{\mathbf{w},\mathbf{D}}(\mathbf{c}) := \boldsymbol{\alpha}(\mathbf{h}^t)^\top \mathbf{D},\ \mathbf{h}^t := \mathrm{RNN}(\mathbf{h}^{t-1}, \mathbf{c}) \end{aligned} \tag{22}$$

Note that an RNN-based context encoder generates different explanations for each time point, $\boldsymbol{\theta}^t$ (Figure 9). All $\boldsymbol{\theta}^t$ are generated using context- and time-specific attention $\boldsymbol{\alpha}(\mathbf{h}^t)$ over the dictionary $\mathbf{D}$. We adopt the training objective from (19) with the same likelihood (20). The model is a special case of CENs with structured explanations (Section 4.2.2).

Table 5: Performance of the baselines and CENs with structured explanations. The numbers are averages from 5-fold cross-validation; the std. are on the order of the least significant digit. "Acc@K" denotes accuracy at the K-th temporal quantile (see main text for explanation).

SUPPORT2					**PhysioNet Challenge 2012**				
Model	**Acc@25**	**Acc@50**	**Acc@75**	**RAE**	**Model**	**Acc@25**	**Acc@50**	**Acc@75**	**RAE**
Cox	84.1	73.7	47.6	0.90	Cox	93.0	69.6	49.1	0.24
Aalen	87.1	66.2	45.8	0.98	Aalen	93.3	78.7	57.1	0.31
CRF	84.4	89.3	79.2	0.59	CRF	93.2	85.1	65.6	0.14
MLP-CRF	**87.7**	89.6	80.1	0.62	LSTM-CRF	93.9	86.3	68.1	**0.11**
MLP-CEN	85.5	**90.8**	**81.9**	**0.56**	LSTM-CEN	**94.8**	**87.5**	**70.1**	**0.09**

6.3.3 Survival Analysis of Patients in Intense Care Units

We evaluate the proposed model against baselines on two survival prediction tasks.

Datasets. We use two publicly available datasets for survival analysis of of the intense care unit (ICU) patients: (a) SUPPORT2,[8] and (b) data from the PhysioNet 2012 challenge.[9] The data was preprocessed and used as follows.

`SUPPORT2`: The data had 9105 patient records (7105 training, 1000 validation, 1000 test) and 73 variables. We selected 50 variables for both $\mathbf{C}$ and $\mathbf{X}$ features (*i.e.*, the context and the variables of interest were identical). Categorical features (such as `race` or `sex`) were one-hot encoded. The values of all features were non-negative, and we filled the missing values with -1 to preserve the information about missingness. For CRF-based predictors, we capped the survival timeline at 3 years and converted it into 156 discrete 7-day intervals.

`PhysioNet`: The data had 4000 patient records, each represented by a 48-hour irregularly sampled 37-dimensional time-series of different measurements taken during the patient's stay at the ICU. We resampled and mean-aggregated the time-series at 30 min frequency. This resulted in a large number of missing values that we filled with 0. The resampled time-series were used as the context, $\mathbf{C}$. For the attributes, $\mathbf{X}$, we took the values of the last available measurement for each variable in the series. For CRF-based predictors, we capped the survival timeline at 60 days and converted into 60 discrete intervals.

Models. For baselines, we use the classical Aalen and Cox models[10] and the CRF from (Yu et al., 2011). All the baselines used $\mathbf{X}$ as their inputs. Next, we combine CRFs with neural encoders in two ways:

(i) We apply CRFs to the outputs from the neural encoders (the models denoted MLP-CRF and LSTM-CRF).[11] Note that parameters of such CRF layer assign weights to the latent features and are not interpretable in terms of the attributes of interest.

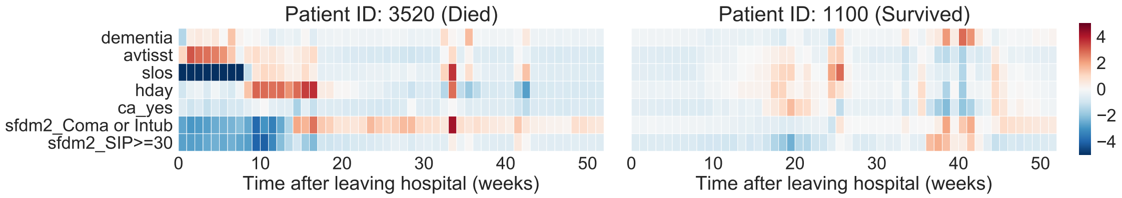
---

[8] `http://biostat.mc.vanderbilt.edu/wiki/Main/DataSets`.

[9] `https://physionet.org/challenge/2012/`.

[10] Implementation based on `https://github.com/CamDavidsonPilon/lifelines`.

[11] Similar models have been very successful in the natural language applications (Collobert et al., 2011).


Patient ID: 3520 (Died)

Patient ID: 1100 (Survived)

Figure 10: Weights of the CEN-generated CRF explanations for two patients from SUPPORT2 dataset for a set of the most influential features: `dementia` (comorbidity), `avtisst` (avg. TISS, days 3-25), `slos` (days from study entry to discharge), `hday` (day in hospital at study admit), `ca_yes` (the patient had cancer), `sfdm2_Coma or Intub` (intubated or in coma at month 2), `sfdm2_SIP` (sickness impact profile score at month 2). Higher weight values correspond to higher contributions to the risk of death after a given time.

(ii) We use CENs with CRF-based explanations, that process the context variables, $\mathbf{C}$, using the same neural networks as in (i) and output the sequence of parameters $\boldsymbol{\theta}^{1:m}$ for CRFs, while the latter act on the attributes, $\mathbf{X}$, to make structured predictions.

More details on the architectures and training are given in Appendix B.3.

Metrics. Following Yu et al. (2011), we use two metrics specific to survival analysis:

(a) Accuracy of correctly predicting survival of a patient at times that correspond to 25%, 50%, and 75% population-level temporal quantiles (*i.e.*, the time points such that the corresponding % of the population in the data were discharged from the study due to censorship or death).

(b) The relative absolute error (RAE) between the predicted and actual time of death for non-censored patients.

Performance. The results for all models are given in Table 5. Our implementation of the CRF baseline slightly improves upon the performance reported by Yu et al. (2011). MLP-CRF and LSTM-CRF improve upon plain CRFs but, as we noted, can no longer be interpreted in terms of the original variables. CENs outperform or closely match neural CRF models on all metrics while providing interpretable explanations for the predicted risk for each patient at each point in time.

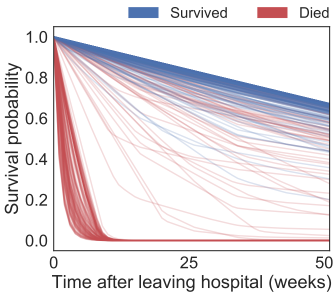


Figure 11: CEN-predicted survival curves for 500 random patients from SUPPORT2. Color indicates death within 1 year after leaving the hospital.

Qualitative analysis. To inspect predictions of CENs qualitatively, for any given patient, we can visualize the weights assigned by the corresponding explanation to the respective attributes. Figure 10 shows weights of the explanations for a subset of the most influential features for two patients from SUPPORT2 dataset who were predicted as survivor/non-survivor.

These temporal charts help us (a) to better understand which features the model selects as the most influential at each point in time, and (b) to identify potential inconsistencies in the model or the data—for example, using a chart as in Figure 10 we identified and excluded a feature (`hospdead`) from SUPPORT2 data, which initially was included but leaked information about the outcome as it directly indicated in-hospital death. Finally, explanations also allow us to better understand patient-specific temporal dynamics of the contributing factors to the survival rates predicted by the model (Figure 11).

7. Conclusion

In this paper, we have introduced contextual explanation networks (CENs)—a class of models that learn to predict by generating and leveraging intermediate context-specific explanations. We have formally defined CENs as a class of probabilistic models, considered a number of special cases (*e.g.*, the mixture-of-experts model), and derived learning and inference algorithms within the encoder-decoder framework for simple and sequentially-structured outputs. We have shown that there are certain conditions when post-hoc explanations are erroneous and misleading. Such cases are hard to detect unless explanation is a part of the prediction process itself, as in CEN. Finally, learning to predict and to explain jointly turned out to have a number of benefits, including strong regularization, consistency, and ability to generate explanations with no computational overhead, as shown in our case studies.

We would like to point out a few limitations of our approach and potential ways of addressing those in the future work. Firstly, while each prediction made by CEN comes with an explanation, the process of conditioning on the context is still uninterpretable. Ideas similar to context selection (Liu et al., 2017) or rationale generation (Lei et al., 2016) may help improve interpretability of the conditioning. Secondly, the space of explanations considered in this work assumes the same graphical structure and parameterization for all explanations and uses a simple sparse dictionary constraint. This might be limiting, and one could imagine using a more hierarchically structured space of explanations instead, bringing to bear amortized inference techniques (Rudolph et al., 2017). Nonetheless, we believe that the proposed class of models is useful not only for improving prediction capabilities, but also for model diagnostics, pattern discovery, and general data analysis, especially when machine learning is used for decision support in high-stakes applications.

8. Acknowledgements

We thank Willie Neiswanger and Mrinmaya Sachan for many useful comments on an early draft of the paper, and Ahmed Hefny, Shashank J. Reddy, Bryon Aragam, and Ruslan Salakhutdinov for helpful discussions. This work was supported by NIH R01GM114311. M.A. was supported in part by the CMLH Fellowship.

Appendix A. Proofs

A.1 Proof of Proposition 1

Assume that $\mathbb{P}\left(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\theta}\right)$ factorizes as $\prod_{\mathbf{a} \in \mathcal{V}_{\mathbf{Y}}} \mathbb{P}\left(\mathbf{Y}_{\mathbf{a}} \mid \mathbf{Y}_{\mathrm{MB}(\mathbf{a})}, \mathbf{X}, \boldsymbol{\theta}_{\mathbf{a}}\right)$, where $\mathbf{a}$ denotes subsets of the $\mathbf{Y}$ variables and $\mathrm{MB}(\mathbf{a})$ stands for the corresponding Markov blankets. Using the definition of CEN given in (3), we have:

$$
\begin{aligned}
\mathbb{P}\left(\mathbf{Y} \mid \mathbf{X}, \mathbf{C}\right) &= \int \mathbb{P}\left(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\theta}\right) \mathbb{P}\left(\boldsymbol{\theta} \mid \mathbf{C}\right) d\boldsymbol{\theta} \\
&= \int \prod_{\mathbf{a} \in \mathcal{V}_{\mathbf{Y}}} \mathbb{P}\left(\mathbf{Y}_{\mathbf{a}} \mid \mathbf{Y}_{\mathrm{MB}(\mathbf{a})}, \mathbf{X}, \boldsymbol{\theta}_{\mathbf{a}}\right) \prod_{j} \mathbb{P}\left(\theta_j \mid \mathbf{C}\right) d\boldsymbol{\theta} \\
&= \prod_{\mathbf{a} \in \mathcal{V}_{\mathbf{Y}}} \left[\int \mathbb{P}\left(\mathbf{Y}_{\mathbf{a}} \mid \mathbf{Y}_{\mathrm{MB}(\mathbf{a})}, \mathbf{X}, \boldsymbol{\theta}_{\mathbf{a}}\right) \prod_{j \in \mathbf{a}} \mathbb{P}\left(\theta_j \mid \mathbf{C}\right) d\boldsymbol{\theta}_{\mathbf{a}}\right] \\
&= \prod_{\mathbf{a} \in \mathcal{V}_{\mathbf{Y}}} \mathbb{P}\left(\mathbf{Y}_{\mathbf{a}} \mid \mathbf{Y}_{\mathrm{MB}(\mathbf{a})}, \mathbf{X}, \mathbf{C}\right)
\end{aligned}
\tag{A.1}
$$

A.2 Proof of Proposition 4

To derive the lower bound on the contribution of explanations in terms of expected accuracy, we first need to bound the probability of the error when only $\boldsymbol{\theta}$ are used for prediction:

$$
\mathbb{P}_e := \mathbb{P}\left(\hat{\mathbf{Y}}(\boldsymbol{\theta}) \neq \mathbf{Y}\right) = \mathbb{E}_{\boldsymbol{\theta} \sim \mathbb{P}(\boldsymbol{\theta})}\left[\mathbb{P}\left(\hat{\mathbf{Y}} \neq \mathbf{Y} \mid \boldsymbol{\theta}\right)\right],
$$

which we bound using the Fano's inequality (Ch. 2.11, Cover and Thomas, 2012):

$$
\mathcal{H}\left(\mathbb{P}_e\right) + \mathbb{P}_e \log\left(|\mathcal{Y}| - 1\right) \geq \mathcal{H}\left(\mathcal{Y} \mid \boldsymbol{\theta}\right) \tag{A.2}
$$

Since the error $(\hat{\mathbf{Y}}(\boldsymbol{\theta}) \neq \mathbf{Y})$ is a binary random variable, then $\mathcal{H}\left(\mathbb{P}_e\right) \leq 1$. After weakening the inequality and using $\mathcal{H}\left(\mathcal{Y} \mid \boldsymbol{\theta}\right) \geq \delta$ from the proposition statement, we get:

$$
\mathbb{E}_{\boldsymbol{\theta} \sim \mathbb{P}(\boldsymbol{\theta})}\left[\mathbb{P}\left(\hat{\mathbf{Y}} \neq \mathbf{Y} \mid \boldsymbol{\theta}\right)\right] \geq \frac{\mathcal{H}\left(\mathcal{Y} \mid \boldsymbol{\theta}\right) - 1}{\log |\mathcal{Y}|} \geq \frac{\delta - 1}{\log |\mathcal{Y}|} \tag{A.3}
$$

The claimed lower bound (16) follows after we combine (A.3) and the assumed bound on the accuracy of the model in terms of ε given in (15).

A.3 Proof of Theorem 6

To prove the theorem, consider the case when f is defined by a CEN, instead of $\mathbf{x}$ we have $(\mathbf{c}, \mathbf{x})$, and the class of approximations, G, coincides with the class of explanations, and hence can be represented by $\boldsymbol{\theta}$. In this setting, we can pose the same problem as:

$$
\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(f, \boldsymbol{\theta}, \pi_{\mathbf{c}, \mathbf{x}}) + \Omega(\boldsymbol{\theta}) \tag{A.4}
$$

Suppose that CEN produces $\boldsymbol{\theta}^\star$ explanation for the context $\mathbf{c}$ using a deterministic encoder, ϕ. The question is whether and under which conditions $\hat{\boldsymbol{\theta}}$ can recover $\boldsymbol{\theta}^\star$. Theorem 6 answers the question in affirmative and provides a concentration result for the case when hypotheses are linear. Here, we prove Theorem 6 for a little more general class of log-linear explanations: $\operatorname{logit}\left\{\mathbb{P}\left(Y = 1 \mid \mathbf{x}, \theta\right)\right\} = \mathbf{a}(\mathbf{x})^\top \boldsymbol{\theta}$, where $\mathbf{a}$ is a C-Lipschitz vector-valued function whose values have a zero-mean distribution when $(\mathbf{x}, \mathbf{c})$ are sampled from $\pi_{\mathbf{x},\mathbf{c}}$[12]. For simplicity of the analysis, we consider binary classification and omit the regularization term, $\Omega(g)$. We define the loss function, $\mathcal{L}(f, \boldsymbol{\theta}, \pi_{\mathbf{x},\mathbf{c}})$, as:

$$\mathcal{L} = \frac{1}{K} \sum_{k=1}^{K} \left(\operatorname{logit}\left\{\mathbb{P}\left(Y = 1 \mid \mathbf{x}_k - \mathbf{x}, \mathbf{c}_k\right)\right\} - \operatorname{logit}\left\{\mathbb{P}\left(Y = 1 \mid \mathbf{x}_k - \mathbf{x}, \boldsymbol{\theta}\right)\right\}\right)^2 \tag{A.5}$$

where $(\mathbf{x}_k, \mathbf{c}_k) \sim \pi_{\mathbf{x},\mathbf{c}}$ and $\pi_{\mathbf{x},\mathbf{c}} := \pi_{\mathbf{x}} \pi_{\mathbf{c}}$ is a distribution concentrated around $(\mathbf{x}, \mathbf{c})$. Without loss of generality, we also drop the bias terms in the linear models and assume that $\mathbf{a}(\mathbf{x}_k - \mathbf{x})$ are centered.

Proof The optimization problem (A.4) reduces to the least squares linear regression:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{K} \sum_{k=1}^{K} \left(\operatorname{logit}\left\{\mathbb{P}\left(Y = 1 \mid \mathbf{x}_k - \mathbf{x}, \mathbf{c}_k\right)\right\} - \mathbf{a}(\mathbf{x}_k - \mathbf{x})^\top \boldsymbol{\theta}\right)^2 \tag{A.6}$$

We consider deterministic encoding, $\mathbb{P}\left(\boldsymbol{\theta} \mid \mathbf{c}\right) := \delta(\boldsymbol{\theta}, \boldsymbol{\phi}(\mathbf{c}))$, and hence we have:

$$\begin{aligned} \operatorname{logit}\left\{\mathbb{P}\left(Y = 1 \mid \mathbf{x}_k - \mathbf{x}, \mathbf{c}_k\right)\right\} &= \operatorname{logit}\left\{\mathbb{P}\left(Y = 1 \mid \mathbf{x}_k - \mathbf{x}, \boldsymbol{\theta} = \boldsymbol{\phi}(\mathbf{c}_k)\right)\right\} \\ &= \mathbf{a}(\mathbf{x}_k - \mathbf{x})^\top \boldsymbol{\phi}(\mathbf{c}_k) \end{aligned} \tag{A.7}$$

To simplify the notation, we denote $\mathbf{a}_k := \mathbf{a}(\mathbf{x}_k - \mathbf{x})$, $\boldsymbol{\phi}_k := \boldsymbol{\phi}(\mathbf{c}_k)$, and $\boldsymbol{\phi} := \boldsymbol{\phi}(\mathbf{c})$. The solution of (A.6) now can be written in a closed form:

$$\hat{\boldsymbol{\theta}} = \left[\frac{1}{K} \sum_{k=1}^{K} \mathbf{a}_k \mathbf{a}_k^\top\right]^{+} \left[\frac{1}{K} \sum_{k=1}^{K} \mathbf{a}_k \mathbf{a}_k^\top \boldsymbol{\phi}_k\right] \tag{A.8}$$

Note that $\hat{\boldsymbol{\theta}}$ is a random variable since $(\mathbf{x}_k, \mathbf{c}_k)$ are randomly generated from $\pi_{\mathbf{x},\mathbf{c}}$. To further simplify the notation, denote $M := \frac{1}{K} \sum_{k=1}^{K} \mathbf{a}_k \mathbf{a}_k^\top$. To get a concentration bound on $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\|$, we will use the continuity of $\phi(\cdot)$ and $\mathbf{a}(\cdot)$, concentration properties of $\pi_{\mathbf{x},\mathbf{c}}$ around $(\mathbf{x}, \mathbf{c})$, and some elementary results from random matrix theory. To be more concrete, since we assumed that $\pi_{\mathbf{x},\mathbf{c}}$ factorizes, we further let $\pi_{\mathbf{x}}$ and $\pi_{\mathbf{c}}$ concentrate such that $\mathbb{P}_{\pi_{\mathbf{x}}}\left(\|\mathbf{x}' - \mathbf{x}\| > t\right) < \varepsilon_{\mathbf{x}}(t)$ and $\mathbb{P}_{\pi_{\mathbf{c}}}\left(\|\mathbf{c}' - \mathbf{c}\| > t\right) < \varepsilon_{\mathbf{c}}(t)$, respectively, where $\varepsilon_{\mathbf{x}}(t)$ and $\varepsilon_{\mathbf{c}}(t)$ both go to 0 as $t \to \infty$, potentially at different rates.

---

[12] In case of logistic regression, $\mathbf{a}(\mathbf{x}) = [1, x_1, \ldots, x_d]^\top$.

First, we have the following bound from the convexity of the norm:

$$\begin{aligned}
\mathbb{P}\left(\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\| > t\right) &= \mathbb{P}\left(\left\|\frac{1}{K}\sum_{k=1}^{K}\left[M^{+}\mathbf{a}_k\mathbf{a}_k^\top(\boldsymbol{\phi}_k - \boldsymbol{\phi})\right]\right\| > t\right) && \text{(A.9)}\\
&\le \mathbb{P}\left(\frac{1}{K}\sum_{k=1}^{K}\left\|M^{+}\mathbf{a}_k\mathbf{a}_k^\top(\boldsymbol{\phi}_k - \boldsymbol{\phi})\right\| > t\right) && \text{(A.10)}
\end{aligned}$$

By making use of the inequality $\|Ax\| \le \|A\|\|x\|$, where $\|A\|$ denotes the spectral norm of the matrix A, the L-Lipschitz property of $\phi(\mathbf{c})$, the C-Lipschitz property of $\mathbf{a}(\mathbf{x})$, and the concentration of $\mathbf{x}_k$ around $\mathbf{x}$, we have

$$\begin{aligned}
\mathbb{P}\left(\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\| > t\right) &\le \mathbb{P}\left(L\frac{1}{K}\sum_{k=1}^{K}\left\|M^{+}\mathbf{a}_k\mathbf{a}_k^\top\right\|\|\mathbf{c}_k - \mathbf{c}\| > t\right) && \text{(A.11)}\\
&\le \mathbb{P}\left(CL\left\|M^{+}\right\|\frac{1}{K}\sum_{k=1}^{K}\left\|\mathbf{a}_k\mathbf{a}_k^\top\right\|\|\mathbf{c}_k - \mathbf{c}\| > t\right) && \text{(A.12)}\\
&\le \mathbb{P}\left(\frac{CL}{\lambda_{\min}(M)}\frac{1}{K}\sum_{k=1}^{K}\|\mathbf{x}_k - \mathbf{x}\|\|\mathbf{c}_k - \mathbf{c}\| > t\right) && \text{(A.13)}\\
&\le \mathbb{P}\left(\frac{CL\tau^2}{\lambda_{\min}(M)} > t\right) + \mathbb{P}\left(\|\mathbf{x}_k - \mathbf{x}\|\|\mathbf{c}_k - \mathbf{c}\| > \tau^2\right) && \text{(A.14)}\\
&\le \mathbb{P}\left(\lambda_{\min}\left(M/(C\tau)^2\right) < \frac{L}{C^2 t}\right) + \varepsilon_{\mathbf{x}}(\tau) + \varepsilon_{\mathbf{c}}(\tau) && \text{(A.15)}
\end{aligned}$$

Note that we used the fact that the spectral norm of a rank-1 matrix, $\mathbf{a}(\mathbf{x}_k)\mathbf{a}(\mathbf{x}_k)^\top$, is simply the norm of $\mathbf{a}(\mathbf{x}_k)$, and the spectral norm of the pseudo-inverse of a matrix is equal to the inverse of the least non-zero singular value of the original matrix: $\|M^+\| \le \lambda_{\max}(M^+) = \lambda_{\min}^{-1}(M)$.

Finally, we need a concentration bound on $\lambda_{\min}\left(M/(C\tau)^2\right)$ to complete the proof. Note that $\frac{M}{C^2\tau^2} = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{\mathbf{a}_k}{C\tau}\right)\left(\frac{\mathbf{a}_k}{C\tau}\right)^\top$, where the norm of $\left(\frac{\mathbf{a}_k}{C\tau}\right)$ is bounded by 1. If we denote $\mu_{\min}(C\tau)$ the minimal eigenvalue of $\mathsf{Cov}\left[\frac{\mathbf{a}_k}{C\tau}\right]$, we can write the matrix Chernoff inequality (Tropp, 2012) as follows:

$$\mathbb{P}\left(\lambda_{\min}\left(M/(C\tau)^2\right) < \alpha\right) \le d\exp\left\{-KD(\alpha\|\mu_{\min}(C\tau))\right\}, \quad \alpha \in [0, \mu_{\min}(C\tau)]$$

where d is the dimension of $\mathbf{a}_k$, $\alpha := \frac{L}{C^2 t}$, and $D(a\|b)$ denotes the binary information divergence:

$$D(a\|b) = a\log\left(\frac{a}{b}\right) + (1-a)\log\left(\frac{1-a}{1-b}\right)$$

The final concentration bound has the following form:

$$\mathbb{P}\left(\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\| > t\right) \leq d \exp\left\{-KD\left(\frac{L}{C^2 t}\|\mu_{\min}(C\tau)\right)\right\} + \varepsilon_{\mathbf{x}}(\tau) + \varepsilon_{\mathbf{c}}(\tau) \tag{A.16}$$

We see that as $\tau \to \infty$ and $t \to \infty$ all terms on the right hand side vanish, and hence $\hat{\boldsymbol{\theta}}$ concentrates around $\boldsymbol{\theta}^\star$. Note that as long as $\mu_{\min}(C\tau)$ is far from 0, the first term can be made negligibly small by sampling more points around $(\mathbf{x}, \mathbf{c})$. Finally, we set $\tau \equiv t$ and denote the right hand side by $\delta_{K,L,C}(t)$ that goes to 0 as $t \to \infty$ to recover the statement of the original theorem. ■

Remark 7 *We have shown that* $\hat{\boldsymbol{\theta}}$ *concentrates around* $\boldsymbol{\theta}^\star$ *under mild conditions. With more assumptions on the sampling distribution,* $\pi_{\mathbf{x},\mathbf{c}}$*, (e.g., sub-gaussian) one could derive precise convergence rates. Note that we are in total control of any assumptions we put on* $\pi_{\mathbf{x},\mathbf{c}}$ *since precisely that distribution is used for sampling. This is a major difference between the local approximation setup here and the setup of linear regression with random design; in the latter case, we have no control over the distribution of the design matrix, and any assumptions we make could potentially be unrealistic.*

Remark 8 *Note that concentration analysis of a more general case when the loss* $\mathcal{L}$ *is a general convex function and* $\Omega(g)$ *is a decomposable regularizer could be done by using results from the M-estimation theory (Negahban et al., 2009), but would be much more involved and unnecessary for our purposes.*

Appendix B. Experimental Details

This section provides details on the experimental setups including architectures, training procedures, etc. Additionally, we provide and discuss qualitative results for CENs on the MNIST and IMDB datasets.

B.1 Additional Details on the Datasets and Experiment Setups

MNIST. We used the classical split of the dataset into 50k training, 10k validation, and 10k testing points. All models were trained for 100 epochs using the AMSGrad optimizer () with the learning rate of 10^{-3}. No data augmentation was used in any of our experiments. HOG representations were computed using 3 × 3 blocks.

CIFAR10. For this set of experiments, we followed the setup given by Zagoruyko (2015), reimplemented in Keras with TensorFlow (Abadi et al., 2016) backend. The input images were global contrast normalized (a.k.a. GCN whitened) while the rescaled image representations were simply standardized. Again, HOG representations were computed using 3 × 3 blocks. No data augmentation was used in our experiments.

IMDB. We considered the labeled part of the data only (50,000 reviews total). The data were split into 20,000 train, 5,000 validation, and 25,000 test points. The vocabulary was limited to 20,000 most frequent words (and 5,000 most frequent words when constructing BoW representations). All models were trained with the AMSGrad optimizer () with 10^{-2} learning rate. The models were initialized randomly; no pre-training or any other unsupervised/semi-supervised technique was used.

Satellite. As described in the main text, we used a pre-trained `VGG-16` network[13] to extract features from the satellite imagery. Further, we added one fully connected layer network with 128 hidden units used as the context encoder. For the VCEN model, we used dictionary-based encoding with Dirichlet prior and logistic normal distribution as the output of the inference network. For the decoder, we used an MLP of the same architecture as the encoder network. All models were trained with Adam optimizer with 0.05 learning rate. The results were obtained by 5-fold cross-validation.

Medical data. We have used minimal pre-processing of both SUPPORT2 and PhysioNet datasets limited to standardization and missing-value filling. We found that denoting missing values with negative entries (-1) often led a slightly improved performance compared to any other NA-filling techniques. PhysioNet time series data was irregularly sampled across the time, so we had to resample temporal sequences at regular intervals of 30 minutes (consequently, this has created quite a few missing values for some of the measurements). All models were trained using Adam optimizer with 10^{-2} learning rate.

B.2 More on Qualitative Analysis

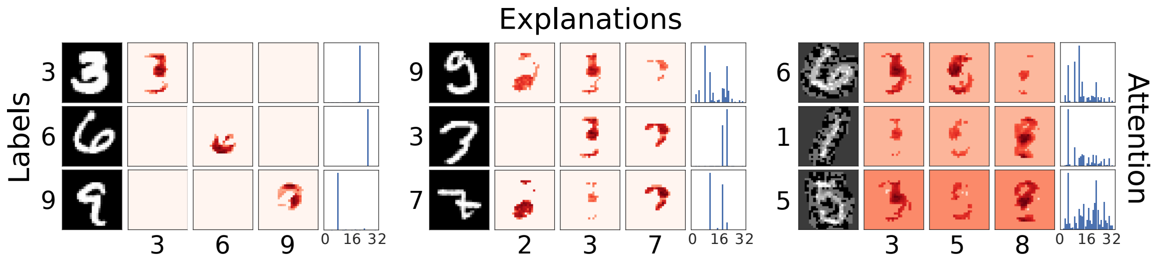
B.2.1 MNIST

Figures 12a, 12b, and 12c visualize explanations for predictions made by CEN-`pxl` on MNIST. The figures correspond to 3 cases where CEN (a) made a correct prediction, (b) made a mistake, and (c) was applied to an adversarial example (and made a mistake). Each chart consists of the following columns: true labels, input images, explanations for the top 3 classes (as given by the activation of the final softmax layer), and attention vectors used to select explanations from the global dictionary. A small subset of explanations from the dictionary is visualized in Figure 12d (the full dictionary is given in Figure 13), where each image is a weight vector used to construct the pre-activation for a particular class. Note that different elements of the dictionary capture different patterns in the data (in Figure 12d, different styles of writing the 0 digit) which CEN actually uses for prediction.

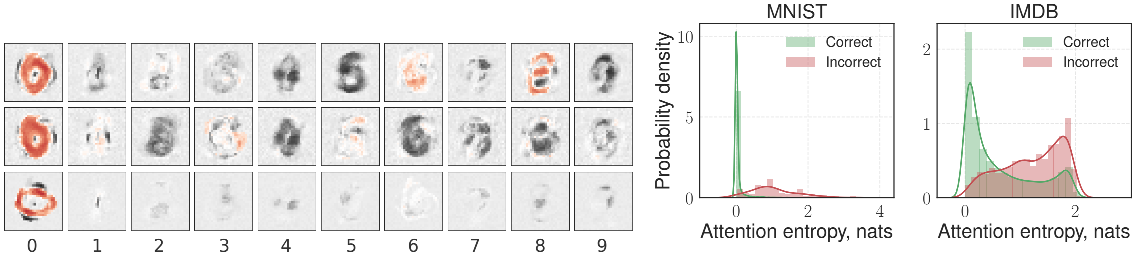
Also note that confident correct predictions (Figures 12a) are made by selecting a single explanation from the dictionary using a sharp attention vector. However, when the model makes a mistake, its attention is often dispersed (Figures 12b and 12c), i.e., there is uncertainty in which pattern it tries to use for prediction. Figure 12e further quantifies this

---

[13] The model was taken form `https://github.com/nealjean/predicting-poverty`.



(a) Correct **(b)** Misclassified **(c)** Adversarial



(d) Selected elements of the explanation dictionary **(e)** Attention

Figure 12: Explanations generated by CEN for the 3 top classes and the corresponding attention vectors for (a) correctly classified, (b) misclassified, and (c) adversarially constructed images. Adversarial examples were generated using the fast gradient sign method (FGSM) (Papernot et al., 2016). (d) Elements from the learned 32-element dictionary that correspond to different writing styles of 0 digits. (e) Histogram of the attention entropy for correctly and incorrectly classified test instances for CEN-`pxl` on MNIST and CEN-`tpc` on IMDB.

phenomenon by plotting histogram of the attention entropy for all test examples which were correctly and incorrectly classified. While CENs are certainly not adversarial-proof, high entropy of the attention vectors is indicative of ambiguous or out-of-distribution examples which is helpful for model diagnostics.

B.2.2 IMDB

Similar to MNIST, we train CEN-`tpc` with linear explanations in terms of topics on the IMDB dataset. In addition to Figure 6, Figure 14 visualizes the full dictionary of size 16 learned by CEN-`tpc`. Each column corresponds to a dictionary atom that represents a typical explanation pattern that CEN attends to before making a prediction. By inspecting the dictionary, we can find interesting patterns. For instance, atoms 5 and 11 assign inverse weights to topics `[kid, child, disney, family]` and `[sexual, violence, nudity, sex]`. Depending on the context of the review, CEN may use one of these patterns to predict the sentiment. Note that these two topics are negatively correlated across all dictionary elements, which again is quite intuitive.

B.2.3 Satellite

We visualize the two explanations, M1 and M2, learned by CEN-`att` on the Satellite dataset in full in Figures 15a and provide additional correlation plots between the selected explanation and values of each survey variable in Figure 15b.

B.3 Model Architectures

Architectures of the model used in our experiments are summarized in Tables 6, 7, 8.

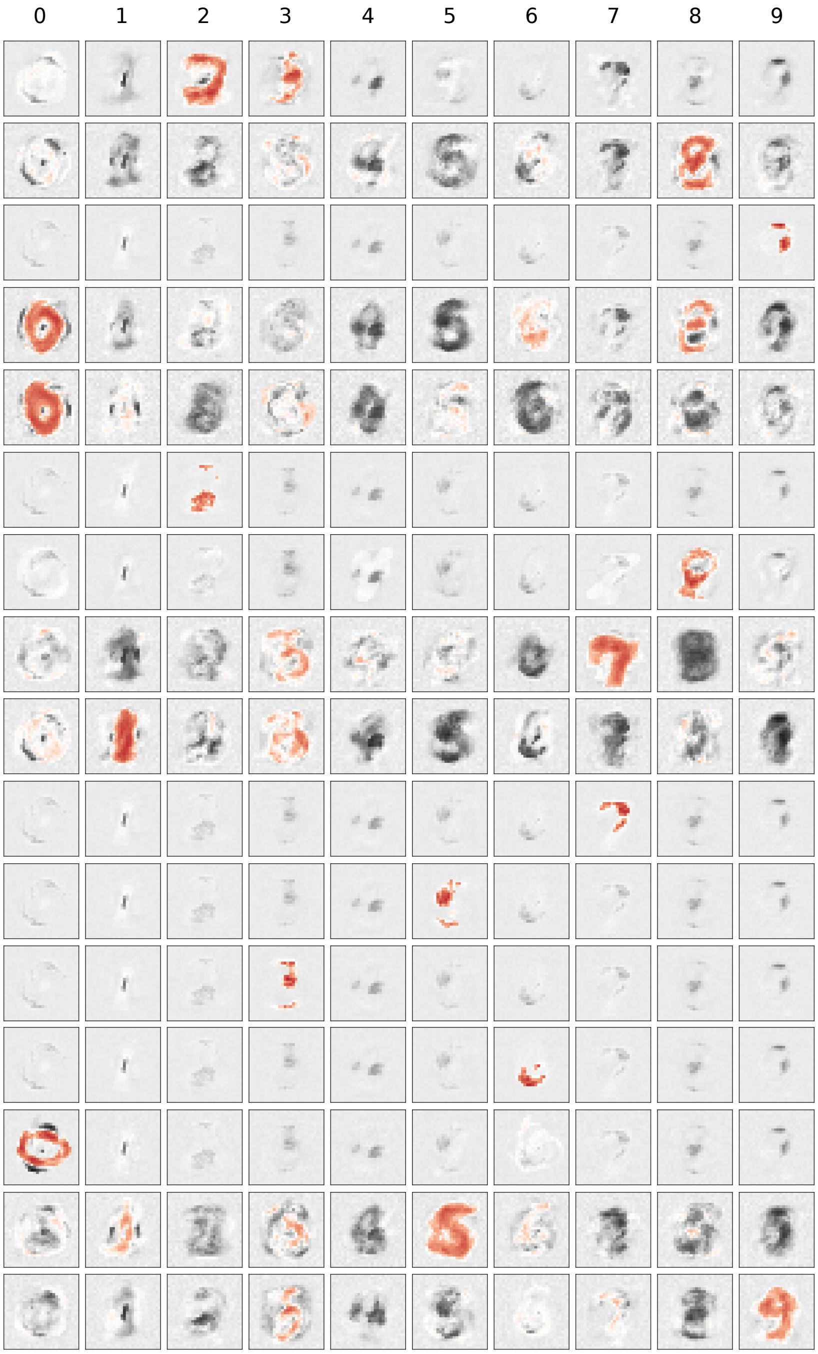
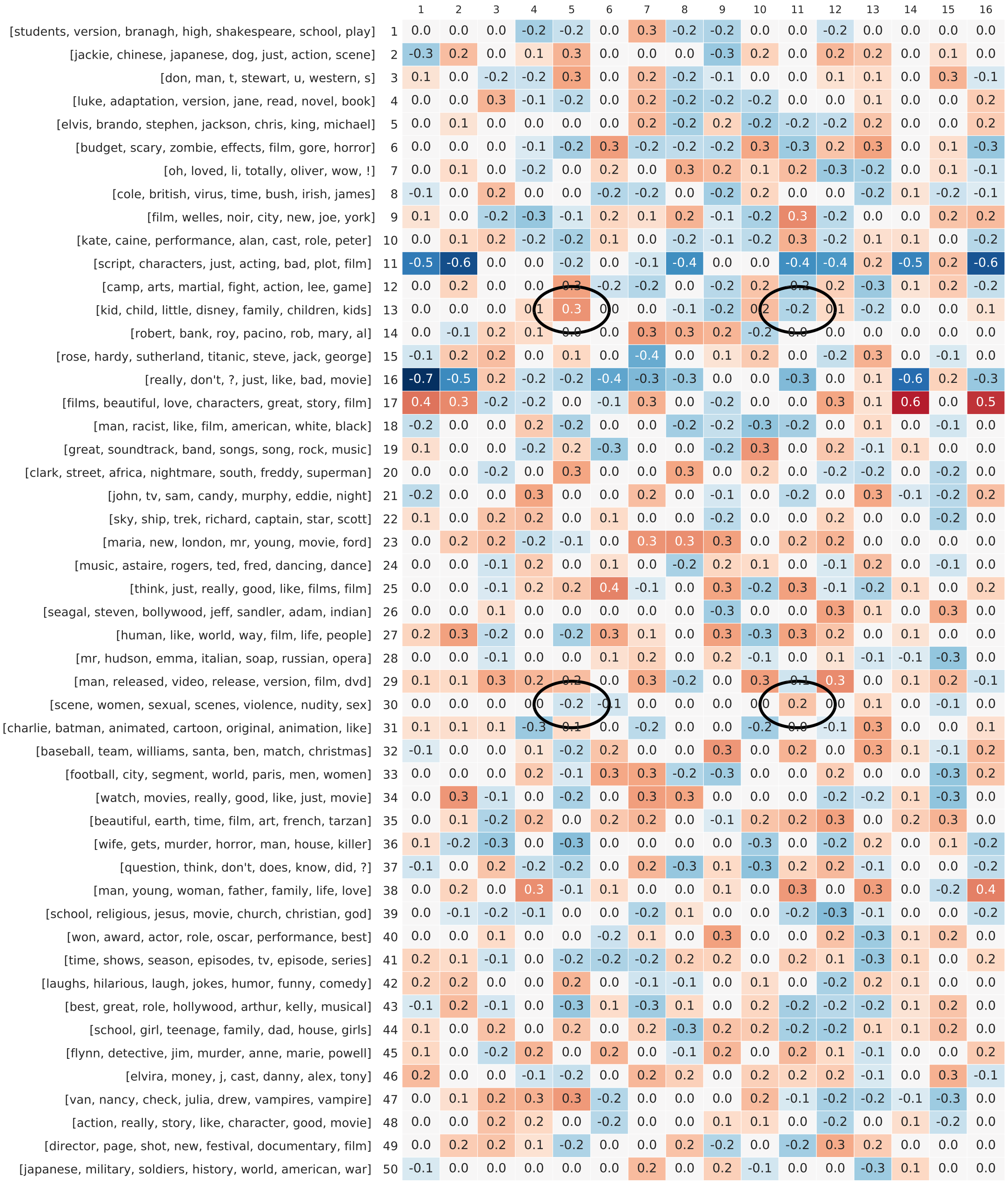
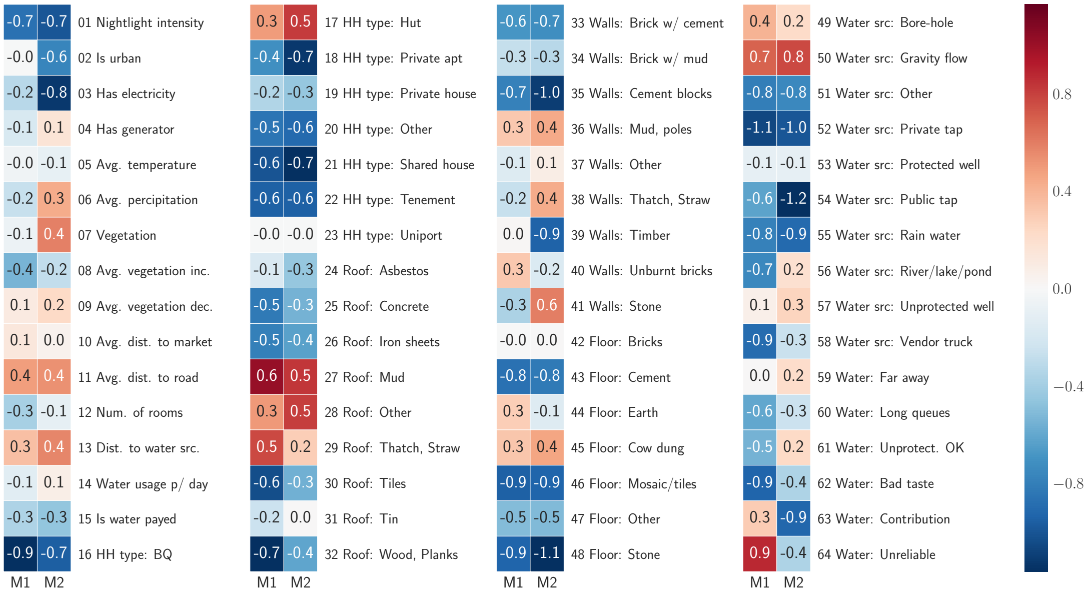


Figure 13: Visualization of the model dictionary learned by CEN on MNIST. Each row corresponds to a dictionary element, and each column corresponds to the weights of the model voting for each class of digits. Images visualize the weights of the models. Red corresponds to high positive values, dark gray to high negative values, and white to values that are close to 0.



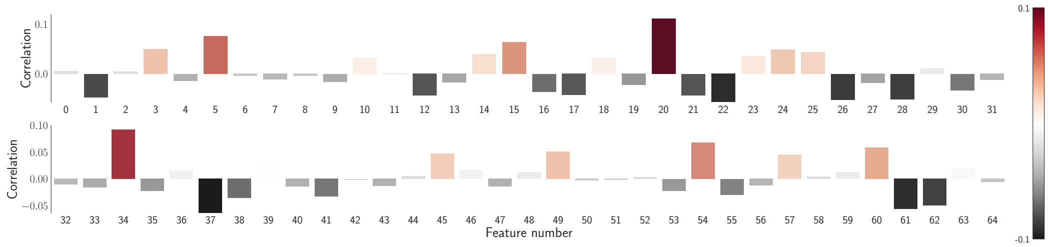
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
[students, version, branagh, high, shakespeare, school, play]	1	0.0	0.0	0.0	-0.2	-0.2	0.0	0.3	-0.2	-0.2	0.0	0.0	-0.2	0.0	0.0	0.0	0.0
[jackie, chinese, japanese, dog, just, action, scene]	2	-0.3	0.2	0.0	0.1	0.3	0.0	0.0	0.0	-0.3	0.2	0.0	0.2	0.2	0.0	0.1	0.0
[don, man, t, stewart, u, western, s]	3	0.1	0.0	-0.2	-0.2	0.3	0.0	0.2	-0.2	-0.1	0.0	0.0	0.1	0.1	0.0	0.3	-0.1
[luke, adaptation, version, jane, read, novel, book]	4	0.0	0.0	0.3	-0.1	-0.2	0.0	0.2	-0.2	-0.2	-0.2	0.0	0.0	0.1	0.0	0.0	0.2
[elvis, brando, stephen, jackson, chris, king, michael]	5	0.0	0.1	0.0	0.0	0.0	0.0	0.2	-0.2	0.2	-0.2	-0.2	-0.2	0.2	0.0	0.0	0.2
[budget, scary, zombie, effects, film, gore, horror]	6	0.0	0.0	0.0	-0.1	-0.2	0.3	-0.2	-0.2	-0.2	0.3	-0.3	0.2	0.3	0.0	0.1	-0.3
[oh, loved, li, totally, oliver, wow, !]	7	0.0	0.1	0.0	-0.2	0.0	0.2	0.0	0.3	0.2	0.1	0.2	-0.3	-0.2	0.0	0.1	-0.1
[cole, british, virus, time, bush, irish, james]	8	-0.1	0.0	0.2	0.0	0.0	-0.2	-0.2	0.0	-0.2	0.2	0.0	0.0	-0.2	0.1	-0.2	-0.1
[film, welles, noir, city, new, joe, york]	9	0.1	0.0	-0.2	-0.3	-0.1	0.2	0.1	0.2	-0.1	-0.2	0.3	-0.2	0.0	0.0	0.2	0.2
[kate, caine, performance, alan, cast, role, peter]	10	0.0	0.1	0.2	-0.2	-0.2	0.1	0.0	-0.2	-0.1	-0.2	0.3	-0.2	0.1	0.1	0.0	-0.2
[script, characters, just, acting, bad, plot, film]	11	-0.5	-0.6	0.0	0.0	-0.2	0.0	-0.1	-0.4	0.0	0.0	-0.4	-0.4	0.2	-0.5	0.2	-0.6
[camp, arts, martial, fight, action, lee, game]	12	0.0	0.2	0.0	0.0	0.3	-0.2	-0.2	0.0	-0.2	0.2	-0.2	0.2	-0.3	0.1	0.2	-0.2
[kid, child, little, disney, family, children, kids]	13	0.0	0.0	0.0	0.1	0.3	0.0	0.0	-0.1	-0.2	0.2	-0.2	0.1	-0.2	0.0	0.0	0.1
[robert, bank, roy, pacino, rob, mary, al]	14	0.0	-0.1	0.2	0.1	0.0	0.0	0.3	0.3	0.2	-0.2	0.0	0.0	0.0	0.0	0.0	0.0
[rose, hardy, sutherland, titanic, steve, jack, george]	15	-0.1	0.2	0.2	0.0	0.1	0.0	-0.4	0.0	0.1	0.2	0.0	-0.2	0.3	0.0	-0.1	0.0
[really, don't, ?, just, like, bad, movie]	16	-0.7	-0.5	0.2	-0.2	-0.2	-0.4	-0.3	-0.3	0.0	0.0	-0.3	0.0	0.1	-0.6	0.2	-0.3
[films, beautiful, love, characters, great, story, film]	17	0.4	0.3	-0.2	-0.2	0.0	-0.1	0.3	0.0	-0.2	0.0	0.0	0.3	0.1	0.6	0.0	0.5
[man, racist, like, film, american, white, black]	18	-0.2	0.0	0.0	0.2	-0.2	0.0	0.0	-0.2	-0.2	-0.3	-0.2	0.0	0.1	0.0	-0.1	0.0
[great, soundtrack, band, songs, song, rock, music]	19	0.1	0.0	0.0	-0.2	0.2	-0.3	0.0	0.0	-0.2	0.3	0.0	0.2	-0.1	0.1	0.0	0.0
[clark, street, africa, nightmare, south, freddy, superman]	20	0.0	0.0	-0.2	0.0	0.3	0.0	0.0	0.3	0.0	0.2	0.0	-0.2	-0.2	0.0	-0.2	0.0
[john, tv, sam, candy, murphy, eddie, night]	21	-0.2	0.0	0.0	0.3	0.0	0.0	0.2	0.0	-0.1	0.0	-0.2	0.0	0.3	-0.1	-0.2	0.2
[sky, ship, trek, richard, captain, star, scott]	22	0.1	0.0	0.2	0.2	0.0	0.1	0.0	0.0	-0.2	0.0	0.0	0.2	0.0	0.0	-0.2	0.0
[maria, new, london, mr, young, movie, ford]	23	0.0	0.2	0.2	-0.2	-0.1	0.0	0.3	0.3	0.3	0.0	0.2	0.2	0.0	0.0	0.0	0.0
[music, astaire, rogers, ted, fred, dancing, dance]	24	0.0	0.0	-0.1	0.2	0.0	0.1	0.0	-0.2	0.2	0.1	0.0	-0.1	0.2	0.0	-0.1	0.0
[think, just, really, good, like, films, film]	25	0.0	0.0	-0.1	0.2	0.2	0.4	-0.1	0.0	0.3	-0.2	0.3	-0.1	-0.2	0.1	0.0	0.2
[seagal, steven, bollywood, jeff, sandler, adam, indian]	26	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	-0.3	0.0	0.0	0.3	0.1	0.0	0.3	0.0
[human, like, world, way, film, life, people]	27	0.2	0.3	-0.2	0.0	-0.2	0.3	0.1	0.0	0.3	-0.3	0.3	0.2	0.0	0.1	0.0	0.0
[mr, hudson, emma, italian, soap, russian, opera]	28	0.0	0.0	-0.1	0.0	0.0	0.1	0.2	0.0	0.2	-0.1	0.0	0.1	-0.1	-0.1	-0.3	0.0
[man, released, video, release, version, film, dvd]	29	0.1	0.1	0.3	0.2	0.2	0.0	0.3	-0.2	0.0	0.3	-0.1	0.3	0.0	0.1	0.2	-0.1
[scene, women, sexual, scenes, violence, nudity, sex]	30	0.0	0.0	0.0	0.0	-0.2	-0.1	0.0	0.0	0.0	0.0	0.2	0.0	0.1	0.0	-0.1	0.0
[charlie, batman, animated, cartoon, original, animation, like]	31	0.1	0.1	0.1	-0.3	-0.1	0.0	-0.2	0.0	0.0	-0.2	0.0	-0.1	0.3	0.0	0.0	0.1
[baseball, team, williams, santa, ben, match, christmas]	32	-0.1	0.0	0.0	0.1	-0.2	0.2	0.0	0.0	0.3	0.0	0.2	0.0	0.3	0.1	-0.1	0.2
[football, city, segment, world, paris, men, women]	33	0.0	0.0	0.0	0.2	-0.1	0.3	0.3	-0.2	-0.3	0.0	0.0	0.2	0.0	0.0	-0.3	0.2
[watch, movies, really, good, like, just, movie]	34	0.0	0.3	-0.1	0.0	-0.2	0.0	0.3	0.3	0.0	0.0	0.0	-0.2	-0.2	0.1	-0.3	0.0
[beautiful, earth, time, film, art, french, tarzan]	35	0.0	0.1	-0.2	0.2	0.0	0.2	0.2	0.0	-0.1	0.2	0.2	0.3	0.0	0.2	0.3	0.0
[wife, gets, murder, horror, man, house, killer]	36	0.1	-0.2	-0.3	0.0	-0.3	0.0	0.0	0.0	0.0	-0.3	0.0	-0.2	0.2	0.0	0.1	-0.2
[question, think, don't, does, know, did, ?]	37	-0.1	0.0	0.2	-0.2	-0.2	0.0	0.2	-0.3	0.1	-0.3	0.2	0.2	-0.1	0.0	0.0	-0.2
[man, young, woman, father, family, life, love]	38	0.0	0.2	0.0	0.3	-0.1	0.1	0.0	0.0	0.1	0.0	0.3	0.0	0.3	0.0	-0.2	0.4
[school, religious, jesus, movie, church, christian, god]	39	0.0	-0.1	-0.2	-0.1	0.0	0.0	-0.2	0.1	0.0	0.0	-0.2	-0.3	-0.1	0.0	0.0	-0.2
[won, award, actor, role, oscar, performance, best]	40	0.0	0.0	0.1	0.0	0.0	-0.2	0.1	0.0	0.3	0.0	0.0	0.2	-0.3	0.1	0.2	0.0
[time, shows, season, episodes, tv, episode, series]	41	0.2	0.1	-0.1	0.0	-0.2	-0.2	-0.2	0.2	0.2	0.0	0.2	0.1	-0.3	0.1	0.0	0.2
[laughs, hilarious, laugh, jokes, humor, funny, comedy]	42	0.2	0.2	0.0	0.0	0.2	0.0	-0.1	-0.1	0.0	0.1	0.0	-0.2	0.2	0.1	0.0	0.0
[best, great, role, hollywood, arthur, kelly, musical]	43	-0.1	0.2	-0.1	0.0	-0.3	0.1	-0.3	0.1	0.0	0.2	-0.2	-0.2	-0.2	0.1	0.2	0.0
[school, girl, teenage, family, dad, house, girls]	44	0.1	0.0	0.2	0.0	0.2	0.0	0.2	-0.3	0.2	0.2	-0.2	-0.2	0.1	0.1	0.2	0.0
[flynn, detective, jim, murder, anne, marie, powell]	45	0.1	0.0	-0.2	0.2	0.0	0.2	0.0	-0.1	0.2	0.0	0.2	0.1	-0.1	0.0	0.0	0.2
[elvira, money, j, cast, danny, alex, tony]	46	0.2	0.0	0.0	-0.1	-0.2	0.0	0.2	0.2	0.0	0.2	0.2	0.2	-0.1	0.0	0.3	-0.1
[van, nancy, check, julia, drew, vampires, vampire]	47	0.0	0.1	0.2	0.3	0.3	-0.2	0.0	0.0	0.0	0.2	-0.1	-0.2	-0.2	-0.1	-0.3	0.0
[action, really, story, like, character, good, movie]	48	0.0	0.0	0.2	0.2	0.0	-0.2	0.0	0.0	0.1	0.1	0.0	-0.2	0.0	0.1	-0.2	0.0
[director, page, shot, new, festival, documentary, film]	49	0.0	0.2	0.2	0.1	-0.2	0.0	0.0	0.2	-0.2	0.0	-0.2	0.3	0.2	0.0	0.0	0.0
[japanese, military, soldiers, history, world, american, war]	50	-0.1	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.2	-0.1	0.0	0.0	-0.3	0.1	0.0	0.0

Figure 14: The full dictionary learned by CEN-`tpc` model: rows correspond to topics and columns correspond to dictionary atoms. Very small values were thresholded for visualization clarity. Different atoms capture different prediction patterns; for example, atom 5 assigns a highly positive weight to the `[kid, child, disney, family]` topic and down-weighs `[sexual, violence, nudity, sex]`, while atom 11 acts in an opposite manner. Given the context of the review, CEN combines just a few atoms to make a prediction.



M1	M2		M1	M2		M1	M2		M1	M2	
-0.7	-0.7	01 Nightlight intensity	0.3	0.5	17 HH type: Hut	-0.6	-0.7	33 Walls: Brick w/ cement	0.4	0.2	49 Water src: Bore-hole
-0.0	-0.6	02 Is urban	-0.4	-0.7	18 HH type: Private apt	-0.3	-0.3	34 Walls: Brick w/ mud	0.7	0.8	50 Water src: Gravity flow
-0.2	-0.8	03 Has electricity	-0.2	-0.3	19 HH type: Private house	-0.7	-1.0	35 Walls: Cement blocks	-0.8	-0.8	51 Water src: Other
-0.1	0.1	04 Has generator	-0.5	-0.6	20 HH type: Other	0.3	0.4	36 Walls: Mud, poles	-1.1	-1.0	52 Water src: Private tap
-0.0	-0.1	05 Avg. temperature	-0.6	-0.7	21 HH type: Shared house	-0.1	0.1	37 Walls: Other	-0.1	-0.1	53 Water src: Protected well
-0.2	0.3	06 Avg. percipitation	-0.6	-0.6	22 HH type: Tenement	-0.2	0.4	38 Walls: Thatch, Straw	-0.6	-1.2	54 Water src: Public tap
-0.1	0.4	07 Vegetation	-0.0	-0.0	23 HH type: Uniport	0.0	-0.9	39 Walls: Timber	-0.8	-0.9	55 Water src: Rain water
-0.4	-0.2	08 Avg. vegetation inc.	-0.1	-0.3	24 Roof: Asbestos	0.3	-0.2	40 Walls: Unburnt bricks	-0.7	0.2	56 Water src: River/lake/pond
0.1	0.2	09 Avg. vegetation dec.	-0.5	-0.3	25 Roof: Concrete	-0.3	0.6	41 Walls: Stone	0.1	0.3	57 Water src: Unprotected well
0.1	0.0	10 Avg. dist. to market	-0.5	-0.4	26 Roof: Iron sheets	-0.0	0.0	42 Floor: Bricks	-0.9	-0.3	58 Water src: Vendor truck
0.4	0.4	11 Avg. dist. to road	0.6	0.5	27 Roof: Mud	-0.8	-0.8	43 Floor: Cement	0.0	0.2	59 Water: Far away
-0.3	-0.1	12 Num. of rooms	0.3	0.5	28 Roof: Other	0.3	-0.1	44 Floor: Earth	-0.6	-0.3	60 Water: Long queues
0.3	0.4	13 Dist. to water src.	0.5	0.2	29 Roof: Thatch, Straw	0.3	0.4	45 Floor: Cow dung	-0.5	0.2	61 Water: Unprotect. OK
-0.1	0.1	14 Water usage p/ day	-0.6	-0.3	30 Roof: Tiles	-0.9	-0.9	46 Floor: Mosaic/tiles	-0.9	-0.4	62 Water: Bad taste
-0.3	-0.3	15 Is water payed	-0.2	0.0	31 Roof: Tin	-0.5	-0.5	47 Floor: Other	0.3	-0.9	63 Water: Contribution
-0.9	-0.7	16 HH type: BQ	-0.7	-0.4	32 Roof: Wood, Planks	-0.9	-1.1	48 Floor: Stone	0.9	-0.4	64 Water: Unreliable

(a) Full visualization of models M1 and M2 learned by CEN on Satellite data.



(b) Correlation between the selected explanation and the value of a particular survey variable.

Figure 15: Additional visualizations for CENs trained on the Satellite data.

Table 6: Top-performing architectures used in our experiments on MNIST and IMDB datasets.

(a) MNIST

Convolutional Encoder		
Convolutional Block	layer	Conv2D
	# filters	32
	kernel size	3×3
	strides	1×1
	padding	valid
	activation	ReLU
	layer	Conv2D
	# filters	32
	kernel size	3×3
	strides	1×1
	padding	valid
	activation	ReLU
	layer	MaxPoo2D
	pooling size	2×2
	dropout	0.25
	layer	Dense
	units	128
	dropout	0.50
# blocks		1
# params		1.2M

Contextual Explanations	
model	Logistic regr.
features	HOG (3, 3)
# features	729
standardized	Yes
dictionary	256
l_1 penalty	$5 \cdot 10^{-5}$
l_2 penalty	$1 \cdot 10^{-6}$
model	Logistic reg.
features	Pixels (20, 20)
# features	400
standardized	Yes
dictionary	64
l_1 penalty	$5 \cdot 10^{-5}$
l_2 penalty	$1 \cdot 10^{-6}$

Contextual VAE	
prior	Dir(0.2)
sampler	LogisticNormal

(b) IMDB

Squential Encoder	
layer	Embedding
vocabulary	20k
dimension	1024
layer	LSTM
bidirectional	Yes
units	256
max length	200
dropout	0.25
rec. dropout	0.25
layer	MaxPool1D
# params	23.1M

Contextual Explanations	
model	Logistic reg.
features	BoW
# features	20k
Dictionary	32
l_1 penalty	$5 \cdot 10^{-5}$
l_2 penalty	$1 \cdot 10^{-6}$
model	Logistic reg.
features	Topics
# features	50
Dictionary	16
l_1 penalty	$1 \cdot 10^{-6}$
l_2 penalty	$1 \cdot 10^{-8}$

Contextual VAE	
Prior	Dir(0.1)
Sampler	LogisticNormal

Table 7: Top-performing architectures used in our experiments on CIFAR10 and Satellite datasets. `VGG-16`architecture for CIFAR10 was taken from https://github.com/szagoruyko/cifar.torch but implemented in Keras with TensorFlow backend. Weights of the pre-trained `VGG-F`model for the Satellite experiments were taken from https://github.com/nealjean/predicting-poverty.

(a) CIFAR10

Convolutional Encoder		
VGG-16	model	`VGG-16`
	pretrained	No
	fixed weights	No
MLP	layer	Dense
	pretrained	No
	fixed weights	No
	units	16
	dropout	0.25
	activation	ReLU
# params		20.0M

Contextual Explanations	
model	Logistic reg.
features	HOG (3, 3)
# features	1024
dictionary	16
l_1 penalty	$1 \cdot 10^{-5}$
l_2 penalty	$1 \cdot 10^{-6}$

Contextual VAE	
prior	Dir(0.2)
sampler	LogisticNormal

(b) Satellite

Convolutional Encoder		
VGG-F	model	`VGG-F`
	pretrained	Yes
	fixed weights	Yes
MLP	layer	Dense
	pretrained	No
	fixed weights	No
	units	128
	dropout	0.25
	activation	ReLU
# trainable params		0.5M

Contextual Explanations	
model	Logistic reg.
features	Survey
# features	64
dictionary	16
l_1 penalty	$1 \cdot 10^{-3}$
l_2 penalty	$1 \cdot 10^{-4}$
# params	

Contextual VAE	
prior	Dir(0.2)
sampler	LogisticNormal

Table 8: Top-performing architectures used in our experiments on SUPPORT2 and PhysioNet.

(a) SUPPORT2

MLP Encoder			**Contextual Explanations**	
MLP	layer	Dense	model	Linear CRF
	pretrained	No	features	Measurements
	fixed weights	No	# features	50
	units	64	dictionary	16
	dropout	0.50	l_1 penalty	$1 \cdot 10^{-3}$
	activation	ReLU	l_2 penalty	$1 \cdot 10^{-4}$

(b) PhysioNet Challenge 2012

Sequential Encoder			**Contextual Explanations**	
LSTM	layer	LSTM	model	Linear CRF
	bidirectional	No	features	Statistics
	units	32	# features	111
	max length	150	dictionary	16
	dropout	0.25	l_1 penalty	$1 \cdot 10^{-3}$
	rec. dropout	0.25	l_2 penalty	$1 \cdot 10^{-4}$

References

O.O. Aalen. A linear regression model for the analysis of life time. *Statistics in Medicine, 8(8):907–925*, 1989.

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

Maruan Al-Shedivat, Andrew Gordon Wilson, Yunus Saatchi, Zhiting Hu, and Eric P Xing. Learning scalable deep kernels with recurrent structure. *Journal of Machine Learning Research*, 18(82):1–37, 2017.

David Belanger and Andrew McCallum. Structured prediction energy networks. In *Proceedings of the International Conference on Machine Learning*, 2016.

Luca Bertinetto, João F Henriques, Jack Valmadre, Philip Torr, and Andrea Vedaldi. Learning feed-forward one-shot learners. In *Advances in Neural Information Processing Systems*, pages 523–531, 2016.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

Rich Caruana, Hooshang Kangarloo, JD Dionisio, Usha Sinha, and David Johnson. Case-based explanation of non-case-based learning methods. In *Proceedings of the AMIA Symposium*, page 212, 1999.

Rich Caruana et al. 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*, pages 1721–1730. ACM, 2015.

Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. *arXiv preprint arXiv:1603.00550*, 2016.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug), 2011.

Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

DR Cox. Regression Models and Life-Tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 187–220, 1972.

George E. Dahl, Ryan P. Adams, and Hugo Larochelle. Training restricted boltzmann machines on word observations. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pages 1163–1170, USA, 2012. Omnipress. ISBN 978-1-4503-1285-1. URL http://dl.acm.org/citation.cfm?id=3042573.3042723.

Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*, pages 3079–3087, 2015.

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

Bert De Brabandere, Xu Jia, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. In *Neural Information Processing Systems (NIPS)*, 2016.

Adji B. Dieng, Chong Wang, Jianfeng Gao, and John William Paisley. Topicrnn: A recurrent neural network with long-range semantic dependency. In *International Conference on Learning Representations*, 2017.

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

Harrison Edwards and Amos Storkey. Towards a neural statistician. *arXiv preprint arXiv:1606.02185*, 2016.

Bradley Efron. Logistic regression, survival analysis, and the kaplan-meier curve. *Journal of the American statistical Association*, 83(402):414–425, 1988.

Kuzman Ganchev, Jennifer Gillenwater, Ben Taskar, et al. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11(Jul):2001–2049, 2010.

Yuanjun Gao, Evan W Archer, Liam Paninski, and John P Cunningham. Linear dynamical neural population models through nonlinear embeddings. In *Advances in Neural Information Processing Systems*, pages 163–171, 2016.

Marta Garnelo, Dan Rosenbaum, Chris J Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo J Rezende, and SM Eslami. Conditional neural processes. *arXiv preprint arXiv:1807.01613*, 2018.

David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

Trevor Hastie and Robert Tibshirani. Varying-coefficient models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 757–796, 1993.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep structured output learning for unconstrained text recognition. *arXiv preprint arXiv:1412.5903*, 2014.

Neal Jean, Marshall Burke, Michael Xie, W Matthew Davis, David B Lobell, and Stefano Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 353(6301):790–794, 2016.

Wenxin Jiang and Martin A Tanner. Hierarchical mixtures-of-experts for exponential family regression models: approximation and maximum likelihood estimation. *Annals of Statistics*, pages 987–1011, 1999.

Matthew Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in Neural Information Processing Systems*, pages 2946–2954, 2016.

Rie Johnson and Tong Zhang. Effective use of word order for text categorization with convolutional neural networks. *arXiv preprint arXiv:1412.1058*, 2014.

Rie Johnson and Tong Zhang. Supervised and semi-supervised text categorization using lstm for region embeddings. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 526–534, 2016.

Andrej Karpathy, Justin Johnson, and Li Fei-Fei. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*, 2015.

Been Kim, Cynthia Rudin, and Julie A Shah. The bayesian case model: A generative approach for case-based reasoning and prototype classification. In *Advances in Neural Information Processing Systems*, pages 1952–1960, 2014.

Been Kim, Oluwasanmi O Koyejo, and Rajiv Khanna. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances In Neural Information Processing Systems*, pages 2280–2288, 2016.

P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning (ICML)*, 2017.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques.* MIT press, 2009.

Rahul G Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *AAAI*, pages 2101–2109, 2017.

John Lafferty, Andrew McCallum, Fernando Pereira, et al. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289, 2001.

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, 2014.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*, 2016.

Jimmy Lei Ba, Kevin Swersky, Sanja Fidler, and Ruslan Salakhutdinov. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4247–4255, 2015.

Bruce G Lindsay. Mixture models: theory, geometry and applications. In *NSF-CBMS regional conference series in probability and statistics*, pages i–163. JSTOR, 1995.

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

Liping Liu, Francisco Ruiz, and David Blei. Context selection for embedding models. In *Advances in Neural Information Processing Systems*, pages 4817–4826, 2017.

Tania Lombrozo. The structure and function of explanations. *Trends in cognitive sciences*, 10(10):464–470, 2006.

Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 142–150, Stroudsburg, PA, USA, 2011a. Association for Computational Linguistics. ISBN 978-1-932432-87-9. URL `http://dl.acm.org/citation.cfm?id=2002472.2002491`.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 142–150. Association for Computational Linguistics, 2011b.

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*, 2016.

Sahand Negahban, Bin Yu, Martin J Wainwright, and Pradeep K Ravikumar. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. In *Advances in Neural Information Processing Systems*, pages 1348–1356, 2009.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 427–436, 2015.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against deep learning systems using adversarial examples. *arXiv preprint arXiv:1602.02697*, 2016.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why Should I Trust You?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

Maja Rudolph, Francisco Ruiz, Stephan Mandt, and David Blei. Exponential family embeddings. In *Advances in Neural Information Processing Systems*, pages 478–486, 2016.

Maja Rudolph, Francisco Ruiz, and David Blei. Structured embedding models for grouped data. In *Advances in Neural Information Processing Systems*, pages 250–260, 2017.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850, 2016.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685*, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Charles Sutton, Andrew McCallum, et al. An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, 4(4):267–373, 2012.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Sebastian Thrun and Lorien Pratt. *Learning to learn.* Springer, 1998.

Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(4):389–434, 2012.

Manasi Vartak, Hugo Larochelle, and Arvind Thiagarajan. A meta-learning perspective on cold-start recommendations for items. In *Advances in Neural Information Processing Systems*, pages 6888–6898, 2017.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016.

Joseph Wang and Venkatesh Saligrama. Local supervised learning through space partitioning. In *NIPS*, 2012.

Sida Wang and Christopher D. Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 90–94, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=2390665.2390688.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 370–378, 2016.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.

Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

Chun-Nam J Yu, Russell Greiner, Hsiu-Chin Lin, and Vickie Baracos. Learning patient-specific cancer survival distributions as a sequence of dependent regressors. In *Advances in Neural Information Processing Systems*, pages 1845–1853, 2011.

Sergey Zagoruyko. 92.45% on CIFAR-10 in Torch. http://torch.ch/blog/2015/07/30/cifar.html, 2015.