Contextual Feature Selection with Conditional Stochastic Gates

Ram Dyuthi Sristi [1] **Ofir Lindenbaum** [2] **Shira Lifshitz** [3] **Maria Lavzin** [3] **Jackie Schiller** [3] **Gal Mishne** [1] **Hadas Benisty** [3]

Abstract

Feature selection is a crucial tool in machine learning and is widely applied across various scientific disciplines. Traditional supervised methods generally identify a universal set of informative features for the entire population. However, feature relevance often varies with context, while the context itself may not directly affect the outcome variable. Here, we propose a novel architecture for contextual feature selection where the subset of selected features is conditioned on the value of *context variables*. Our new approach, Conditional Stochastic Gates (c-STG), models the importance of features using conditional Bernoulli variables whose parameters are predicted based on contextual variables. We introduce a hypernetwork that maps context variables to feature selection parameters to learn the context-dependent gates along with a prediction model. We further present a theoretical analysis of our model, indicating that it can improve performance and flexibility over population-level methods in complex feature selection settings. Finally, we conduct an extensive benchmark using simulated and real-world datasets across multiple domains demonstrating that c-STG can lead to improved feature selection capabilities while enhancing prediction accuracy and interpretability.

1. Introduction

Feature selection techniques are vital in Machine Learning (ML) as they identify informative features from large sets of observed variables. These techniques are increasingly crucial across scientific domains due to the high dimensionality of collected data and complex prediction models. Feature selection simplifies models by removing nuisance features and identifying informative features, ultimately improving generalization (Li et al., 2017; Kumar & Minz, 2014; Islam et al., 2022; Everaert et al., 2022). Feature selection can be broadly categorized into filter, wrapper, and embedded methods.

Filter methods (Battiti, 1994; Peng et al., 2005; Estévez et al., 2009; Song et al., 2007; 2012; Chen et al., 2017; Sristi et al., 2022) use a predefined criterion, independent of the predictive model, to rank and select features mostly based on statistical measures such as correlation or mutual information (Lewis, 1992). Wrapper methods (Kohavi & John, 1997; Stein et al., 2005; Zhu et al., 2007; Reunanen, 2003; Allen, 2013), on the other hand, use the predictive model itself as a criterion for feature selection. These methods select subsets of features and evaluate the prediction quality based only on those features to identify the subset with the best performance. This can be prohibitively expensive for complex models. Embedded methods (Tibshirani, 1996; Hans, 2009; Li et al., 2011; 2006) incorporate feature selection into the training process of the predictive model, for example, regularization-based techniques like LASSO and its variants (Daubechies et al., 2010; Bertsimas et al., 2017). Here we present a novel embedded method to learn the prediction model and informative features in an end-to-end fashion.

The methods above are global in the sense that they identify a single set of informative features for the entire data set. However, in certain cases, the importance of features varies depending on contextual information, while the context itself does not encode the outcome variable on its own. For example, in healthcare, feature selection reveals which explanatory features directly predict disease risk, but feature relevance and importance can vary based on context, such as age or gender. Thus, considering context adds depth to medical analysis and offers clinicians valuable insights. Similarly, product recommendation systems need to tailor feature selection to the user's location, time, and device (Baltrunas & Ricci, 2009).

Many feature selection methods fail to fully address the intricate relationship between features, context, and outcomes, often neglecting contextual variables or merely concatenating them with explanatory features. This approach hinders interpretability, as it does not clearly show the dependency

[1]University of California San Diego, La Jolla, California, USA [2]Bar-Ilan University , Ramat Gan, Israel [3]Technion - Israel Institute of Technology, Haifa, Israel . Correspondence to: Ram Dyuthi Sristi <rsristi@ucsd.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

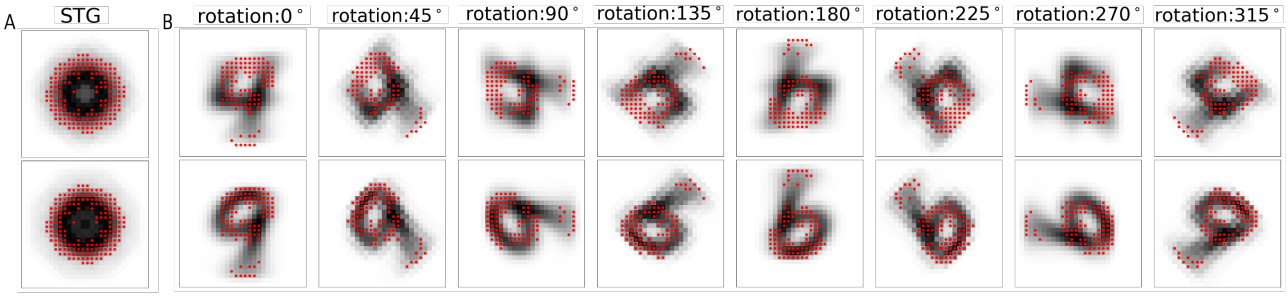

Figure 1. Illustration with "rotating MNIST". We perform a binary classification between rotated versions of the digits 4 (top row) and 9 (bottom row) from the MNIST dataset. We compare the features selected by the global STG (Yamada et al., 2020) model (A) and our proposed c-STG (B), which selects features conditioned on the rotation angle. Each image depicts the mean pixel values across all rotated images (A) or all images at a given rotation angle (B). Red dots indicate the features selected using STG (A) or c-STG (B) for each rotation angle. c-STG can learn to dynamically change its prediction of the most informative features given the context (rotation angle).

of features on the context. An alternative is to train a separate model for different values of categorical contextual variables, e.g., gender, by dividing the data accordingly. However, this strategy significantly increases computational demands with multiple categorical contexts and decreases the available training data for each model, potentially affecting performance. Furthermore, this method necessitates arbitrary binning for continuous contextual variables (e.g., age, location), which coarsens the interpretability of feature importance across contexts. This issue becomes more pronounced when dealing with multiple contextual variables, making it challenging to maintain nuanced interpretability.

For illustration purposes, we consider a binary classification task between digits 4 and 9 observed at various rotations (see Fig. 1). Suppose the goal is to select a subset of informative pixels (features) for classifying the two digits. In the original orientation (zero rotation), the pixels on the top of the image are more significant than those at the bottom for distinguishing between 4 and 9. When we rotate the digits by intervals of 45 degrees, global feature selection models will select almost all features as informative. However, given a specific rotation angle, only a subset of these features are informative. Moreover, the rotation angle is unrelated to the response (the identity of the digit). Using rotation as a contextual variable to select the informative pixels for the classification can alleviate overfitting and improve the interpretability of the model. In Figure 1 (B), we present features (pixels) selected by our proposed approach, conditioned on a rotation variable. The significant features rotate as the context changes, and our model identifies the features that best distinguish between digits 4 and 9 for each context. In contrast, the features selected by the global STG (Yamada et al., 2020) (Fig. 1 A), trained across all rotated images, mainly recover the action of rotation, without providing meaningful interpretable features for class separation.

In this paper, we propose c-STG, a method for context-specific feature selection in a supervised learning setup. We posit that the informativeness of a feature is governed by a conditional Bernoulli distribution, aiming to learn its parameters to enhance feature selection. To enable parameter learning via backpropagation, we reparametrize the discrete Bernoulli variables using a truncated Gaussian (Yamada et al., 2020) whose mean is predicted using a hypernetwork that receives the context variables as input. Furthermore, we propose *weighted* c-STG, which uses a hypernetwork to learn both a score for the significance of the selected features and the parameters of the probabilistic feature selection. In both c-STG and weighted c-STG, we learn the weights of the hypernetwork and the weights of a prediction model by empirical risk minimization. Since the hypernetwork is parametric, we can generalize the contextual feature selection to unseen contexts. Our framework enables studying the explainability of the model along two axes: 1) given a context, we identify the importance of the different features, which is essential for performing an accurate prediction task; and 2) given a feature, we identify how its importance varies across different contexts as well as interpolate or extrapolate to unseen contexts. Overall, our contributions are as follows:

1. We develop an end-to-end machine learning framework comprising two distinct but interconnected networks: a hypernetwork and a prediction network. In our c-STG, the hypernetwork maps contextual variables to feature selection parameters, and the prediction network links these selected features to the outcome. c-STG can handle categorical, continuous, and/or multi-dimensional contextual variables.
2. In weighted c-STG, we augment the c-STG hypernetwork to further output a context-dependent weight vector, evaluating each feature's impact. Thus, the prediction network leverages the selected and weighted features, improving the prediction by combining feature selection and significance.
3. We analyze the optimal risk of c-STG compared with

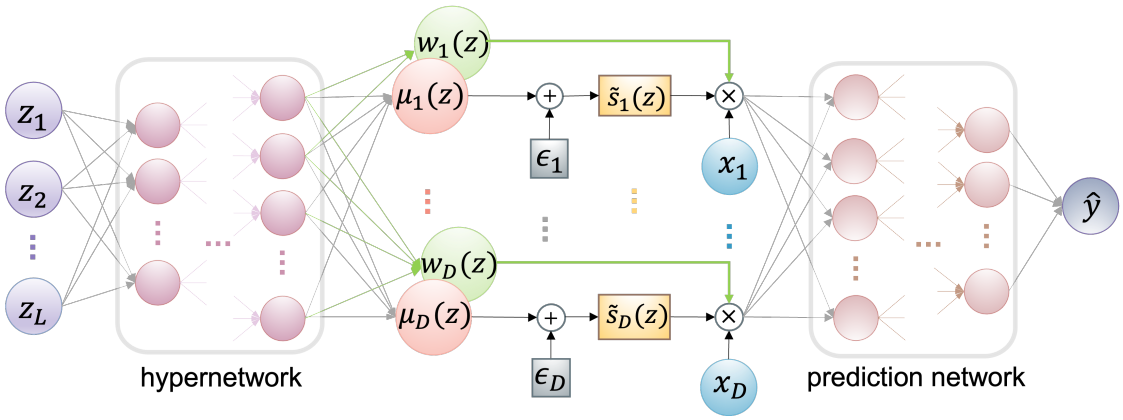

Figure 2. Contextual feature selection framework. Contextual variables z (in purple) feed into the hypernetwork. The hypernetwork outputs the parameters of the gates, $\mu(z)$, which are combined with ϵ to determine if each gate is open or close $\tilde{s}_d$ (yellow) for each feature x_d (blue). For weighted c-STG, the hypernetwork also outputs weight vectors (green), indicating the importance of the selected explanatory features. The selected and weighted features are fed into the prediction model, thus enhancing its ability to process feature significance in predictions.

STG and study the optimal solutions under a linear regression prediction model.

4. We conduct comprehensive empirical evaluations on simulated and real-world datasets across healthcare, housing, and neuroscience, demonstrating the effectiveness and adaptability of our proposed methods compared to existing techniques.

Related Work: Several solutions have been proposed for the problem of context-specific feature selection, including Contextual Explanation Networks (CEN) (Al-Shedivat et al., 2020), Contextualized ML (Lengerich et al., 2023), and Contextual LASSO (Thompson et al., 2023). These methods usually consider the prediction task as a linear function of explanatory features and determine the model's parameters based on contextual variables. In contrast, our method accommodates both linear and non-linear models, thus extending the applicability beyond the constraints of traditional methods. While CEN and Contextualized ML do not employ sparsity constraints, Contextual LASSO utilizes an ℓ_1 regularization that leads to shrinkage of the model's coefficients. In contrast, our c-STG approach uses stochastic gates regularization, which effectively approximates the ℓ_0 sparsity constraint in a context-conditioned manner, therefore achieving sparser solutions.

Recently, several dynamic feature selection approaches have been proposed. In sample-wise methods (Chen et al., 2018; Yoon et al., 2018; Yang et al., 2022a) feature selection is tailored to each sample based on its feature values, requiring all feature values to be known in advance. This approach could be limiting in areas like healthcare, where not all possible test results are available for the entire population. Active learning-based methods (Shim et al., 2018; Covert et al., 2023) focus on real-time feature acquisition and iteratively select features for each sample. These methods, though not originally designed for contextual feature selection, can incorporate context by concatenating contextual and explanatory features. Both sample-wise and active learning methods can be used in context-dependent setups. In sample-specific selection, one can ensure contextual variables are automatically deemed significant from the start, subsequently determining the significance of the remaining explanatory features. In active selection methods, contextual variables can be prioritized before choosing significant explanatory features, ensuring a context-informed selection process. Both approaches empower the model to identify essential features within the specific context, ensuring context-driven, tailored analysis per sample.

The interpretability of these models, however, is not straightforward. Identifying which features are selected as a function of context would require aggregating the selected features across all samples per context value, thus resorting to coarse binning. In contrast, our c-STG discerns crucial features based on context and can adapt to new, unseen context values, e.g., an age value not encountered in training.

2. Problem Setup and Background

Let $X \subseteq \mathbb{R}^D$, $Z \subseteq \mathbb{R}^L$ and $Y \subseteq \mathbb{R}$ be the random variables corresponding to the input explanatory features, input contextual variables, and output prediction value, respectively. Given the realizations of these random variables from some unknown joint data distribution $P_{X,Z,Y}$, the goal of embedded contextual feature selection methods is to achieve the following simultaneously: 1) construct a hyper-model $h_{\boldsymbol{\phi}} : \mathbb{R}^L \to \{0,1\}^D$ that selects a subset of explanatory features $X_{\mathcal{S}}$ as a function of context $\boldsymbol{z}$; and 2) construct a

model $f_{\boldsymbol{\theta}} : \mathbb{R}^D \to \mathbb{R}$ that predicts the response Y based on these selected features. Given a loss function L, we solve for the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ of the risk minimization problem

$$R(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{X,Z,Y}[L(f_\theta(\boldsymbol{x} \odot s(\boldsymbol{z})), y)], \tag{1}$$

where $s(\boldsymbol{z}) = h_{\boldsymbol{\phi}}(\boldsymbol{z})$ and $\boldsymbol{x}, \boldsymbol{z}$ and $\boldsymbol{y}$ represent a realization of the random variables X, Z and Y following a data distribution $P_{X,Z,Y}$, the feature selection vector, given by a vector of indicator variables for the set $\mathcal{S}$, is the output of the hypernetwork $s(\boldsymbol{z}) = h_{\boldsymbol{\phi}}(\boldsymbol{z}) = \{0,1\}^D$, and $\odot$ denotes the point-wise product.

Throughout this paper, bold lowercase letters denote vectors, e.g., $\boldsymbol{x}$, while scalars are represented by unbolded lowercase letters. Elements within a vector are indicated by subscripts, such as $x_1, x_2, \ldots, x_n$. For a function f that maps a vector $\boldsymbol{x}$ to another vector, $f(\boldsymbol{x})$, the ith element of the resultant vector is denoted as $f_i(\boldsymbol{x})$. $\mathbf{0}$ and $\mathbf{1}$ indicate a vector of all zeros and all ones, respectively. $\boldsymbol{I}$ represents identity matrix. $d \in [D]$ is shorthand for d ranging from 1 to D, inclusive.

3. Conditional STG

The risk minimization in (1) often includes an additional constraint to induce sparsity on the feature selection, for example, $\|\boldsymbol{s}\|_0 \leq D$, which reduces the number of selected features and enhances interpretability. Model interpretability is crucial for understanding complex relations between input features and the predicted target in applications such as health care (Liu et al., 2019), shopping (Baltrunas & Ricci, 2009), and psychology (Everaert et al., 2022).

In practice, ℓ_0 regularization is computationally challenging, especially for high-dimensional data, and cannot be integrated into gradient descent-based optimization commonly used in deep networks. Our proposed solution is a probabilistic and computationally efficient method that facilitates contextual feature selection. This is achieved by applying ℓ_0 regularization to stochastic gates, which mask the input feature conditioned on the context.

First, we introduce conditional Bernoulli gates corresponding to each of the D features as a probabilistic extension of contextual feature selection. Expressly, we assume that the probability of an individual explanatory feature being selected, given contextual features $\boldsymbol{z}$, follows a Bernoulli distribution independent of the probability of selecting other explanatory features. Let $S'|Z$ be a conditional random vector that represents these independent Bernoulli gates, with $\mathbb{P}(S'_d = 1|Z = \boldsymbol{z}) = \pi_d(\boldsymbol{z})$ for $d \in [D]$, and let $s'(\boldsymbol{z}) = s'|\boldsymbol{z}$ be a realization of the random vector $S'|Z$. We term these conditional-Stochastic Gates (c-STG), as the parameter of the distribution of each gate is conditional on the context variables $\boldsymbol{z}$.

The hypernetwork $h'_{\boldsymbol{\phi}}$ learns a parametric function mapping

Algorithm 1 Weighted c-STG

Input: $\boldsymbol{x}^{(k)} \in \mathbb{R}^D, \boldsymbol{z}^{(k)} \in \mathbb{R}^L, y^{(k)} \in \mathbb{R}$, for $k = [K]$
Output: Trained models $f_{\boldsymbol{\theta}}$ and $\tilde{h}_{\boldsymbol{\phi}}$.

Initialize: $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ using Xavier initialization.
while model not converged **do**
  Forward Pass:
  for $k = 1$ to K **do**
    $\mu(\boldsymbol{z}^{(k)}), w(\boldsymbol{z}^{(k)}) = \widetilde{h}_{\boldsymbol{\phi}}(\boldsymbol{z}^{(k)})$
    $\epsilon \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$
    $\widetilde{s}(\boldsymbol{z}^{(k)}) = \max(\mathbf{0}, \min(\mathbf{1}, \mu(\boldsymbol{z}^{(k)}) + \epsilon))$.
    where min and max are applied elementwise.
    $\hat{y}^{(k)} = f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(k)} \odot \widetilde{s}(\boldsymbol{z}^{(k)}) \odot w(\boldsymbol{z}^{(k)}))$
  end for
  $\hat{R}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \sum_{k=1}^{K} \left[L(\hat{y}^{(k)}, y^{(k)}) + \lambda \sum_{d=1}^{D} \Phi\left(\frac{\mu_d(\boldsymbol{z}^{(k)})}{\sigma}\right)\right]$
  Back Propagation:
  Update $\boldsymbol{\theta}$: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \frac{\partial \hat{R}(\boldsymbol{\theta}, \boldsymbol{\phi})}{\partial \boldsymbol{\theta}}$
  Update $\boldsymbol{\phi}$: $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} - \eta \frac{\partial \hat{R}(\boldsymbol{\theta}, \boldsymbol{\phi})}{\partial \boldsymbol{\phi}}$
end while

the contextual variable $\boldsymbol{z}$ to the conditional probability, i.e., the parameter of the Bernoulli distribution. The task of contextual variable selection then boils down to learning the parameters $\pi(\boldsymbol{z}) = h'_{\boldsymbol{\phi}}(\boldsymbol{z})$ of this conditional distribution, which spans over a continuous space $[0,1]^D$, instead of a discrete set $\{0,1\}^D$, as in Eq. (1). Let $h'_{\boldsymbol{\phi}} : \mathbb{R}^L \to [0,1]^D$ be the function that maps the contextual information to the parameters of the probability distribution. This reformulates the regularized version of the risk in Eq. (1) to

$$\begin{aligned}\hat{R}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \hat{\mathbb{E}}_{X,Z,Y} \mathbb{E}_{S'|Z}[L(f_{\boldsymbol{\theta}}(\boldsymbol{x} \odot s'(\boldsymbol{z})), y) \\ + \lambda ||s'(\boldsymbol{z})||_0],\end{aligned} \tag{2}$$

where the parameters of the distribution $S'|Z$, $\pi(\boldsymbol{z})$, are learnt by the hypernetwork, $h'_{\boldsymbol{\phi}}(\boldsymbol{z})$. $\hat{\mathbb{E}}_{X,Z,Y}$ represents the empirical expectation over the observations X, Z and Y and $\mathbb{E}_{S'|Z}||s'(\boldsymbol{z})||_0 = \sum_{d=1}^{D} \pi_d(\boldsymbol{z})$, the sum of Bernoulli parameters. Constraining $\pi_d(\boldsymbol{z})$ to $\{0,1\}$ makes this equivalent to the cardinality-constrained version of Eq. (1) , with a regularized penalty on cardinality rather than an explicit constraint. Moreover, this probabilistic formulation converts the combinatorial search to a search over the space of Bernoulli distribution parameters. Thus, the problem of feature selection translates to finding $\boldsymbol{\theta}^*$ and $\boldsymbol{\phi}^*$ that minimize the empirical risk based on the formulation in Eq. (2).

3.1. Bernoulli Continuous Relaxation for Contextual Feature Selection

Incorporating discrete random variables into a differentiable loss function to retain informative data features is appealing, but discrete variable gradient estimates often have high variance (He & Niyogi, 2004). Consequently, continuous

approximations of discrete variables have been proposed (Maddison et al., 2017; Jang et al., 2017). A stable approach for continuous relaxation uses the Gaussian distribution, more consistent in feature selection than Gumbel-softmax techniques like concrete and hard concrete (Jang et al., 2017), which can lead to high variance in approximating Bernoulli variables (Yamada et al., 2020; Jana et al., 2023). Such relaxations (Louizos et al., 2018) are applied in various areas, including discrete softmax activations (Jang et al., 2017), feature selection (Yamada et al., 2020; Lindenbaum et al., 2021b; Shaham et al., 2022), and sparsification (Lindenbaum et al., 2021a; 2024). We utilize a Gaussian-based relaxation for Bernoulli variables, termed Stochastic Gates (STG) (Yamada et al., 2020), differentiated using the reparameterization trick (Miller et al., 2017; Figurnov et al., 2018).

The Gaussian-based continuous relaxation for the Bernoulli variable is defined as $\widetilde{s}_d(\boldsymbol{z}) = \max(0, \min(1, \mu_d(\boldsymbol{z}) + \epsilon_d)$, where ϵ_d is drawn from a normal distribution $\mathcal{N}(0, \sigma^2)$, with σ fixed throughout training. Unlike the ReLU function, which only clips the negative values to zero, the mean-shifted Gaussian variable clips both the positive and negative values; therefore, it accounts for the binary nature of the original random variable. Here, we learn μ_d as a parametric function of the contextual variables $\boldsymbol{z}$. Thus, the hypernetwork $\widetilde{h}_{\boldsymbol{\phi}}$ aims to learn the parameters of the relaxed-continuous distribution as a function of the context variables $\boldsymbol{z}$ instead of learning the original discrete distribution. Our objective as a minimization of the empirical risk $\hat{R}(\boldsymbol{\theta}, \boldsymbol{\phi})$ is as follows:

$$\min_{\boldsymbol{\theta},\boldsymbol{\phi}} \hat{\mathbb{E}}_{X,Z,Y} \mathbb{E}_{\widetilde{S}|Z} \left[L(f_{\boldsymbol{\theta}}(\boldsymbol{x} \odot \widetilde{s}(\boldsymbol{z})), y) + \lambda ||\widetilde{s}(\boldsymbol{z})||_0 \right], \quad (3)$$

where $\widetilde{S}|Z$ is a random vector with D independent variables $\widetilde{\boldsymbol{s}}_d(\boldsymbol{z}) = \widetilde{\boldsymbol{s}}|\boldsymbol{z}$ for $d \in [D]$ and the parameters of the distribution $\widetilde{S}|Z$, $\pi(\boldsymbol{z})$, are learnt by the hypernetwork, $\widetilde{h}_{\boldsymbol{\phi}}(\boldsymbol{z})$. The regularization term can be further simplified to

$$\mathbb{E}_{\widetilde{S}|Z} ||\widetilde{s}(\boldsymbol{z})||_0 = \sum_{i=1}^{D} \mathbb{P}(\widetilde{s}_d(\boldsymbol{z}) > 0) = \sum_{d=1}^{D} \Phi(\frac{\mu_d(\boldsymbol{z})}{\sigma}), \quad (4)$$

where Φ is the standard Gaussian CDF. The term (4) penalizes open gates so that gates corresponding to features that are not useful for prediction are encouraged to transition into a closed state (which is the case for small $\mu_d(\boldsymbol{z})$). Hence, we perform a context-specific feature selection strategy by inducing sparsity through the empirical mean of the regularization term (4) over multiple realizations of Z. This enables us to select distinct informative features for different contexts while maintaining the sparsity.

In practice, we consider the function class for $\widetilde{h}_{\boldsymbol{\phi}}$ and $f_{\boldsymbol{\theta}}$ to be a class of neural networks parameterized by $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$, respectively. To optimize for these parameters, we use a Monte Carlo sampling gradient estimator of (3), which gives

$$\begin{aligned} \frac{\partial \hat{R}(\boldsymbol{\theta}, \boldsymbol{\phi})}{\partial \boldsymbol{\theta}} &= \frac{1}{K} \sum_{k=1}^{K} \frac{\partial}{\partial \boldsymbol{\theta}} L(f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(k)} \odot \widetilde{s}(\boldsymbol{z}^{(k)}), y^{(k)}) \\ \frac{\partial \hat{R}(\boldsymbol{\theta}, \boldsymbol{\phi})}{\partial \boldsymbol{\phi}} &= \frac{1}{K} \sum_{k=1}^{K} \frac{\partial}{\partial \phi} \mathcal{L}(f_{\theta}(x^{(k)} \odot \widetilde{s}(z^{(k)})), y^{(k)}) \quad (5) \\ &\quad + \lambda \frac{\partial}{\partial \phi} \Phi \left(\frac{\widetilde{h}_{\phi}(z^{(k)})}{\sigma} \right) \end{aligned}$$

where K is the number of Monte Carlo samples (corresponds to the batch size). Our methodology is summarized in Alg. 1 and illustrated in Fig. 2. Note for c-STG, the explanatory features are masked by the feature gates, $\widetilde{s}(\boldsymbol{z})$, and fed into the prediction model $\hat{y}^{(k)} = f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(k)} \odot \widetilde{s}(\boldsymbol{z}^{(k)}))$.

In the initial training phase $\boldsymbol{\phi}$, all gates should have an equal probability of being open or closed. We set $\mu_d(\boldsymbol{z}) = 0.5$ $\forall\, d \in [D]$ so that all gates approximate a "fair" Bernoulli variable. The initialization of the hyper-model's $\boldsymbol{\phi}$ using Xavier initialization (Glorot & Bengio, 2010) and a Sigmoid activation function in the final layer of $\widetilde{h}_{\boldsymbol{\phi}}$ ensures that the means ($\mu_d(\boldsymbol{z})$) are centered around 0.5, $\forall\, \boldsymbol{z}$, early in the training phase. It is worth noting that we need the noise term only during the training phase.

3.2. Theoretical Analysis

We conduct a thorough theoretical analysis to establish the equivalence between our probabilistic formulation, as represented by Eq.2, and the original NP-hard contextual variable selection problem defined in Eq.1. Additionally, we provide proof demonstrating that c-STG achieves a lower risk than STG. Furthermore, we extend our analysis to a linear regression scenario, where we demonstrate the main advantage of c-STG over STG. While STG selects features with consistent significance across contextual variables on average, c-STG adapts the feature selection process according to the contextual variables, thus effectively learning the optimal feature selection as a function of these variables.

Theorem 1. Let $s^*(\boldsymbol{z})$ and $s'^*(\boldsymbol{z})$ represent the optimal feature selection functions in Eq. (1) and its corresponding probabilistic formulation in Eq. (2) respectively. Then $s^*(\boldsymbol{z}) = s'^*(\boldsymbol{z})$.

This theorem suggests that a deterministic search for feature selection can be transitioned to a probabilistic approach. We use the universal function approximators, deep networks, to learn the function $s'^*(\boldsymbol{z})$. In subsequent theorems, we contrast c-STG's and STG's performance and feature selection abilities. In (Yamada et al., 2020), the optimization problem

of the global feature selection using STG is given by

$$\min \hat{\mathbb{E}}_{X,Y}\mathbb{E}_{S'}\left[L(f_\theta(\boldsymbol{x}\odot\boldsymbol{s'}),y)+\lambda||\boldsymbol{s'}||_0\right], \quad (6)$$

where S' is a random vector that represents independent Bernoulli gates, $\mathbb{P}(S'_d=1)=\pi_d$ for $d\in[D]$, and s'_d denotes the realization of the random vector S'_d. Note that S' is similar to $S'(\boldsymbol{z})$ except that the latter is a function of $\boldsymbol{z}$, and the former is constant for all $\boldsymbol{z}$. Thus, their approach optimizes for a fixed feature selection independent of contextual information by promoting sparsity in feature selection through the regularization term $\mathbb{E}_{\widetilde{S}}\left[||\widetilde{\boldsymbol{s}}||_0\right]$. In contrast, we perform context-specific feature selection by maximizing the sparsity through the empirical mean of Eq. (4) across various realizations of Z. The following theorem draws a connection between their empirical risk minimizations.

Theorem 2. c-STG attains an optimal risk lower or equal to the risk attained by STG.

Through the following Theorems 3 and 4, we further emphasize the advantage of c-STG over STG.

Theorem 3. In a linear regression setup, the relationship between the optimal parameters of the conditional Bernoulli stochastic gates, $\pi^*(\boldsymbol{z})$, and the optimal parameters of the non-conditional Bernoulli stochastic gates $\boldsymbol{\pi}^*_{\text{stg}}$ is given by

$$\boldsymbol{\pi}^*_{\text{stg}}=E_Z[\pi^*(\boldsymbol{z})]. \quad (7)$$

Theorem 4. In a linear regression setup, the relationship between the optimal parameters of the conditional Gaussian stochastic gates, $\mu^*(\boldsymbol{z})$, and the optimal parameters of the non-conditional Gaussian stochastic gates $\boldsymbol{\mu}^*_{\text{stg}}$ is given by

$$\boldsymbol{\mu}^*_{\text{stg}}=E_Z[\mu^*(\boldsymbol{z})]. \quad (8)$$

The theorems illustrate that c-STG offers an enhanced feature selection resolution by pinpointing features crucial to specific contexts. This stands in contrast to STG, which tends to select features based on their average importance across various contexts. Theorem 3 delineates this difference in the realm of discrete probability spaces, whereas Theorem 4 addresses the continuous probability spaces. The proofs of all theorems are provided in the appendix.

3.3. Weighted Conditional STG

We extend c-STG to weighted conditional-STG (weighted c-STG) to determine and quantify the significance of the features identified by the conditional stochastic gates. To achieve this, we integrate an additional layer in our model that maps the hypernetwork's penultimate layer, $h_L(\boldsymbol{z})$, to a weight vector $w(\boldsymbol{z})=Wh_L(\boldsymbol{z})+b$, as depicted in Figure 2 (green circles). Explanatory features $\boldsymbol{x}$ are masked by the feature selection output $\widetilde{s}(\boldsymbol{z})$ and weighted by $w(\boldsymbol{z})$, then fed into the prediction model for task execution.

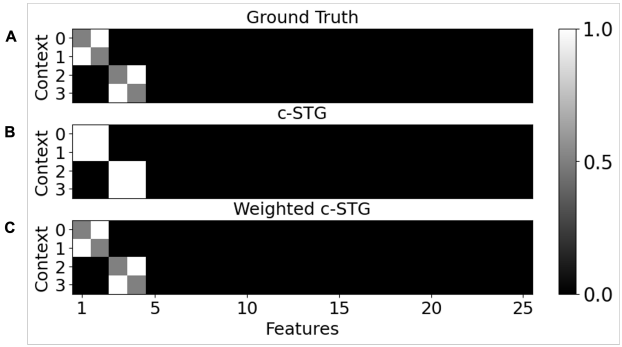

Figure 3. XOR2. (A) Ground truth feature significance as a function of context, z. Feature gates for c-STG (B) and weighted c-STG (C).

The hypernetwork and prediction network parameters are learned using back propagation, as detailed in Algorithm 1.

Theorem 5. Weighted c-STG attains an optimal risk lower or equal to the risk attained by c-STG.

4. Experiments[1]

We evaluate c-STG and weighted c-STG against multiple benchmarks: 1) context-specific techniques Contextual LASSO and CEN; 2) the sample-specific method INVASE (Yoon et al., 2018); 3) an active feature selection approach: AFS (Covert et al., 2023); 4) population-level methods like LASSO and STG; and 5) a prediction model without any feature selection. We train the population-level methods and prediction model without any feature selection using either 1) only the explanatory features as input or 2) concatenating these features with the context variables (referred to as 'with context'). See supplementary material (Appendix C) for implementation details on the hyper-parameters of all methods. We report the performance of all methods in Table 1 and their corresponding number of selected features in (Table 2) for four datasets, demonstrating c-STG and weighted c-STG outperform competing methods.

Simulated Datasets: First, we use synthetic data to validate that our method can identify context-specific informative features while learning non-linear prediction functions. Following synthetic examples in (Yamada et al., 2020; Yang et al., 2022b), we design a nonlinear moving XOR dataset (XOR1), described in the appendix, and a nonlinear weighted moving XOR (XOR2) as follows. We generate a data matrix X with 1000 samples and 25 features, where each entry is sampled from a normal distribution. The contextual variable $\boldsymbol{z}$ is sampled uniformly from $\{0,1,2,3\}$.

[1]Code for c-STG is available in https://github.com/Mishne-Lab/Conditional-STG

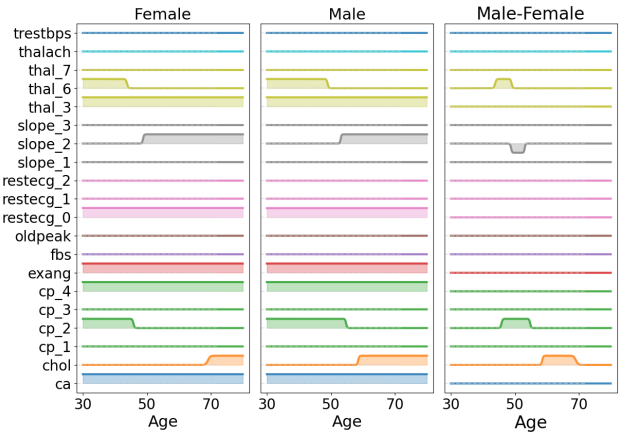

Figure 4. Heart disease. Feature selection gates $\mu(z)$ for each input feature as a function of context—age and gender (left - females and center - males)—produced by weighted c-STG. The difference in the c-STG values between males and females indicates gender-specific informative features as a function of age.

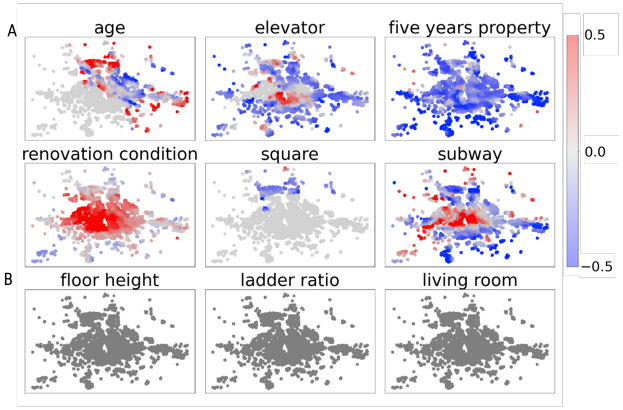

Figure 5. Housing. Geographic significance of nine housing features according to weighted c-STG analysis ($w(z) \times \widetilde{s}(z)$). The red to blue gradient denotes positive to negative impact, respectively, with gray as neutral. Weighted significant features (A) and unselected features (B).

The prediction variable y is generated as

$$\textbf{XOR2:}\ \ \mathrm{y}(\boldsymbol{x};\boldsymbol{z}) = \begin{cases} ReLU(0.5x_1 + x_2), & \text{if } z = 0, \\ ReLU(x_1 + 0.5x_2), & \text{if } z = 1, \\ ReLU(0.5x_3 + x_4), & \text{if } z = 2, \\ ReLU(x_3 + 0.5x_4), & \text{if } z = 3. \end{cases}.$$

Based on the value of $\boldsymbol{z}$, the response variable y for different samples will have different feature significance. Fig. 3 shows that c-STG correctly recovers which features are significant for prediction while weighted c-STG recovers context-dependent importance. Furthermore, we introduce a third synthetic example (XOR3) in the appendix, which demonstrates c-STG's advantage over contextual LASSO, particularly by showcasing how the latter is prone to shrinkage problems in a linear context.

MNIST: We apply our proposed method for image classification using a subset of the MNIST dataset, including digit images of 4 and 9. To create a contextually diverse set of input images, we rotate each original image by intervals of 45 degrees, resulting in eight distinct images per original image. In this example, the goal is to identify which pixels most effectively differentiate between the digits per rotation angle. Applying c-STG or weighted c-STG where rotation serves as context results in a test accuracy of 98%. In Fig.1, we contrast STG with c-STG, illustrating the superiority of context-specific feature selection over population-level analysis. By comparing separate STG models for distinct categorical contexts against a unified c-STG model, we find that c-STG leads to higher accuracy with fewer samples (Fig.12 in the appendix).

Heart disease dataset: We now focus on medical data, specifically, the heart disease dataset from UCI ML repository (Janosi et al., 1988). Given features including chest pain type, resting blood pressure, serum cholesterol levels, and more, our goal is to understand how age and gender influence these biometrics in relation to the risk of heart disease. Age and gender are set as a two-dimensional contextual input to our hypernetwork in this binary classification problem. Table 1 shows the 5-fold cross-validation accuracy where we surpass other methods.

Weighted c-STG analysis (Fig. 4) reveals age and gender-specific feature significance for heart disease. For example, cholesterol ('chol') monitoring becomes crucial with increased age as a risk factor for cardiovascular disease. In males, cholesterol's relevance arises around age 50, likely due to the absence of hormone-induced cholesterol regulation. In females, a significant cholesterol risk arises later, past age 70, when they are post-menopause, and estrogen's protective effect has faded. Similarly, a flat ST segment slope ('slope_2') indicates heart disease risk in females starting at age 45, intensifying post-menopause. For males, 'slope_2' becomes significant after 55, showcasing distinct cardiovascular risk patterns between genders.

Housing dataset:

In the domain of housing price prediction, understanding how various features, such as the renovation condition or floor height, influence its cost based on the house's location in the city requires context-specific feature selection. The Housing dataset (Lianjia, 2017) includes a set of features such as age, elevator, floor height, renovation condition, and the latitude and longitude of the house. We used our proposed approach to predict the price of the house using all

Table 1. Comparison of feature selection. **Bold** indicates best performance and underline indicates second-best.

	XOR1	**XOR2**	**MNIST**	**Heart disease**	**Housing**
	Accuracy (%)	R^2 score	Accuracy (%)	Accuracy (%)	R^2 score
LASSO	50.09 (1.23)	0.3109(0.0175)	83.87 (0.05)	83.50 (3.69)	0.2161 (0.0014)
with context	50.05 (1.09)	0.3105(0.0170)	83.88 (0.04)	83.67 (3.48)	0.2272 (0.0016)
Prediction model	70.74 (17.72)	0.3409 (0.0192)	98.11 (0.05)	86.53 (1.51)	0.2022 (0.0361)
with context	71.93 (11.88)	0.3382 (0.0167)	98.21 (0.07)	83.92 (5.52)	0.2132 (0.0381)
STG	73.98 (0.69)	0.3516 (0.0287)	98.55 (0.08)	86.53 (2.11)	0.2139 (0.0023)
with context	74.22 (1.25)	0.3487 (0.0269)	98.34 (0.07)	87.88 (1.95)	0.2234 (0.0022)
CEN	50.08 (0.67)	0.5186 (0.0126)	87.66 (0.24)	84.17 (5.69)	0.2561 (0.0002)
Contextual LASSO	50.60 (0.61)	0.7316 (0.0090)	97.17 (0.04)	83.19 (5.32)	0.4616 (0.0033)
AFS	78.95 (19.91)	0.7572 (0.0187)	94.83 (0.84)	86.44 (1.07)	0.3963 (0.0092)
INVASE	52.36(1.70)	0.4627(0.0194)	88.06(2.30)	87.22(2.20)	0.3335 (0.0203)
c-STG (Ours)	**100** (0)	0.8739 (0.0107)	98.66 (0.05)	87.55 (3.41)	0.3976 (0.0082)
Weighted c-STG (Ours)	**100** (0)	**0.9956** (0.0008)	**98.69** (0.06)	**89.23** (1.72)	**0.5308** (0.0052)

Table 2. Comparison of a number of features selected on XOR, MNIST, Housing, and heart disease datasets. **Bold** indicates least number of selected features and underline indicates second-best.

	XOR1	**XOR2**	**MNIST**	**Heart disease**	**Housing**
LASSO	18.80 (0.87)	14.20 (2.04)	621.70 (4.77)	13.6 (1.11)	8.00 (0.00)
with context	21.50 (0.92)	14.60 (1.85)	627.5 (3.26)	16.90 (0.70)	10.00 (0.00)
STG	6.60 (1.50)	5.60 (1.02)	585.40 (3.77)	16.60 (2.65)	8.00 (0.00)
with context	4.20 (0.40)	4.20 (0.40)	**148.80** (2.64)	18.00 (4.15)	8.00 (0.00)
Contextual LASSO	4.60 ± 1.93	2.75 (0.93)	357.72 (8.88)	11.79 (3.66)	5.43 (0.29)
AFS	5.40 (4.27)	**2.00** (0.00)	290.00 (20.00)	8.00 (5.48)	**2.70** (0.64)
INVASE	9.85(1.47)	13.04 (0.09)	376.49 (12.30)	9.12 (1.85)	4.80 (0.16)
c-STG (Ours)	**3.00** (0.00)	**2.00** (0.00)	247.25 (10.84)	9.69 (0.81)	**2.70** (0.42)
Weighted c-STG (Ours)	**3.00** (0.00)	**2.00** (0.00)	217.02 (2.87)	**6.50** (0.50)	4.36 (0.30)

the features while considering the location (latitude and longitude) as two-dimensional contextual data. For this dataset, our weighted c-STG outperforms all other models. Visualizing weighted c-STG feature significance in Fig. 5 offers key insights into the Housing dataset. First, feature importance shows a localized pattern, with neighboring locations displaying similar feature significance. Secondly, subway presence in the city center positively impacts house pricing, possibly due to enhanced accessibility and convenience. In contrast, subway presence negatively affects prices in the outskirts, potentially due to increased noise and disruption. Additionally, renovated properties in the city center positively influence pricing, likely because renovations in high-demand urban areas add significant value to properties. These findings provide valuable insights for real estate companies and policymakers. By understanding which features are more important in which locations, they can make informed decisions regarding housing prices, regulations, and development.

Neuroscience: In studying the way the brain encodes behavior, perception, and memory, machine learning models are trained to predict measures of behavior from neuronal activity recordings. Feature selection can provide valuable insights by revealing individual (or subsets of) neurons encoding a particular behavior as a function of task-relevant timing, sensory input, or arousal state. In (Levy et al., 2020), the authors study a motor task where mice are trained to perform a hand-reach task of a food pellet, given an auditory cue (tone). They recorded the activity of neurons in layers 2-3 of the primary motor cortex, where each trial was labeled as successful if the animal managed to grab and consume the food pellet. Cellular networks in the motor cortex are expected to communicate error signals while acquiring a new motor task to achieve improvement across attempts. To test this hypothesis, (Levy et al., 2020) trained a separate SVM binary model to classify trials as success or failure per neuron based on the activity during a short time window. This resulted in training 7866 models (342 neurons $\times$ 23 sliding time windows). The analysis showed that 1) outcome can be consistently decoded from neuronal activity starting 2 seconds after the tone till the end of the trial; 2) 12% of neurons exhibited prolonged activity starting 2 seconds after the tone, on either success or failure trials, thus "reporting" trial outcome.

Using a single c-STG model, we reveal the same biological findings. We set the prediction model to be a binary classifier (success or failure); the explanatory features are the neuronal activity of all neurons in a given time window, and

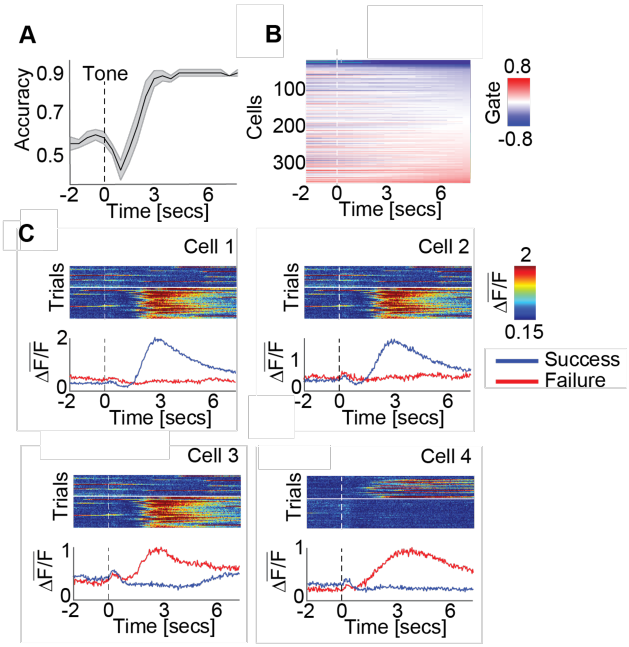

Figure 6. Neuroscience: Context is time. (A) Prediction accuracy vs. time. (B) Weighted gate values for all cells vs. time. (C) Examples of cells selected were for each cell: top - activity vs. time for all successful and failed trials (separated by white horizontal line), bottom - traces of average activity for successful (red) and failed trials (blue).

the contextual variable is *time*. Fig. 6 (A) presents the test accuracy of weighted c-STG as a function of time, indicating that the trial outcome was successfully classified starting from 2 seconds after the tone. The learned weights are presented in (B), where 12% of cells have absolute weight values that are larger than 0.25. Four example neurons with high absolute weights (C) exhibit prolonged activity only during either success or failure. Overall, weighted c-STG captured the complex dynamics of how outcome is encoded within the ensemble neuronal activity; it was able to detect individual neurons reporting outcome as a function of time using a single model, a biological finding which (Levy et al., 2020) used thousands of SVM models to reveal.

Finally, we apply c-STG to another hand-reach study where a mouse was given flavored food pellets: bitter, sweet, or regular (unflavored). We hypothesize that outcome encoding depends on flavor. We train c-STG to classify the outcome based on the activity in the last 4 seconds of each trial, where the contextual variable is *flavor*. Our results in Fig. 7 reveal that the cellular network encodes the outcome differently across flavors. We quantify how similar the encoding is by correlating the gate values across flavors and find they are most different for sweet and bitter (c=0.4), and they are more similar for regular and sweet (c=0.7) than for regular

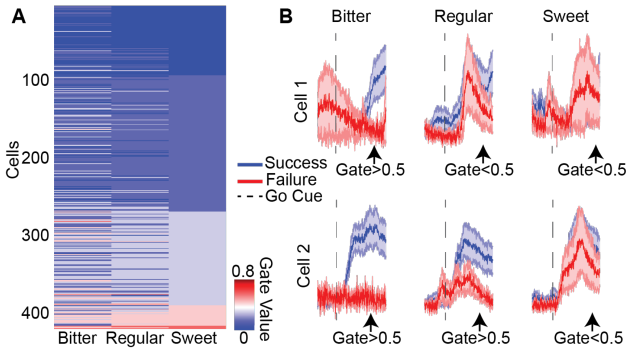

Figure 7. Neuroscience: Context is flavor. (A) c-STG gate values for all cells per flavor. (B) Examples of cells selected by c-STG (gate > 0.5); Averaged activity across trials (mean$\pm$ Standard Error of the Mean) for successful (blue) and failed (red) trials demonstrating differences in outcome encoding across flavors.

and bitter (c=0.6). Overall, c-STG is able to capture the complexity of outcome encoding by the neuronal network across different contexts.

5. Conclusion

We presented an embedded method for context-specific feature selection. We developed conditional stochastic gates based on a hypernetwork to map between contextual variables and the parameters of the conditional distributionOur c-STG leads to improved accuracy and interpretability by efficiently determining which input features are relevant for prediction using a single trained model for categorical, continuous, and/or multi-dimensional context. We demonstrate that our method outperforms embedded feature selection methods and reveals insights across multiple domains. A limitation of our work is the challenge of parameter tuning, specifically the regularization and cross-validation, which is often necessary to find optimal value. Notably, LASSO has a well-established theory for tuning its λ parameter. Developing automated procedures for parameter tuning for c-STG would be an exciting avenue for future research.

Acknowledgements

This research is partially supported by a Simons Foundation Pilot Extension Award - 00003245 (G.M. and R.S.) and Israel Science Foundation 1432/19 (J.S.).

Impact Statement

The potential societal impact of our work is broad and multifaceted. On a positive note, c-STG's ability to incorporate contextual information into feature selection could impact fields such as healthcare by enabling more accurate pa-

tient diagnostics and tailored treatments or urban planning through improved housing and infrastructure development models. However, as with any technological advancement, the potential for misuse exists. For instance, using contextual data like sentiment or mood in social networks could lead to manipulative practices, such as exploiting users' emotional states for targeted content delivery that may not be in their best interest. Similarly, in targeted advertising, c-STG could be employed to differentiate offerings based on sensitive context variables (e.g., location, race), leading to inequitable practices like dynamic pricing or content filtering that exacerbate societal divides.

As researchers, we acknowledge these potential ethical dilemmas and stress the importance of developing and implementing safeguards against misuse. This includes promoting transparency in how models are trained and used, ensuring diverse and inclusive data to prevent bias, and advocating for regulations that protect individuals' rights and privacy. Moreover, it is vital to foster interdisciplinary dialogue between technologists, ethicists, policymakers, and stakeholders to navigate the ethical complexities and societal implications of advanced machine learning technologies.

In conclusion, while c-STG represents a significant leap forward in context-aware machine learning, it is incumbent upon the research community and society at large to carefully consider and address the ethical and societal ramifications of such technologies. By doing so, we can harness the full potential of c-STG and similar innovations for the greater good, ensuring they serve to unite rather than divide, empower rather than exploit, and foster equity rather than exclusion.

References

Al-Shedivat, M., Dubey, A., and Xing, E. Contextual explanation networks. *The Journal of Machine Learning Research*, 21(1):7950–7993, 2020.

Allen, G. I. Automatic feature selection via weighted kernels and regularization. *Journal of Computational and Graphical Statistics*, 22(2):284–299, 2013.

Baltrunas, L. and Ricci, F. Context-based splitting of item ratings in collaborative filtering. In *Proceedings of the third ACM conference on Recommender systems*, pp. 245–248, 2009.

Battiti, R. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on neural networks*, 5(4):537–550, 1994.

Bertsimas, D., Copenhaver, M. S., and Mazumder, R. The trimmed lasso: Sparsity and robustness, 2017.

Chen, J., Stern, M., Wainwright, M. J., and Jordan, M. I. Kernel feature selection via conditional covariance minimization. In *Advances in Neural Information Processing Systems*, pp. 6946–6955, 2017.

Chen, J., Song, L., Wainwright, M., and Jordan, M. Learning to explain: An information-theoretic perspective on model interpretation. In *International conference on machine learning*, pp. 883–892. PMLR, 2018.

Covert, I. C., Qiu, W., Lu, M., Kim, N. Y., White, N. J., and Lee, S.-I. Learning to maximize mutual information for dynamic feature selection. In *International Conference on Machine Learning*, pp. 6424–6447. PMLR, 2023.

Daubechies, I., DeVore, R., Fornasier, M., and Güntürk, C. S. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 63(1):1–38, 2010.

Estévez, P. A., Tesmer, M., Perez, C. A., and Zurada, J. M. Normalized mutual information feature selection. *IEEE Transactions on Neural Networks*, 20(2):189–201, 2009.

Everaert, J., Benisty, H., Gadassi Polack, R., Joormann, J., and Mishne, G. Which features of repetitive negative thinking and positive reappraisal predict depression? an in-depth investigation using artificial neural networks with feature selection. *Journal of Psychopathology and Clinical Science*, 2022.

Figurnov, M., Mohamed, S., and Mnih, A. Implicit reparameterization gradients. In *Advances in Neural Information Processing Systems*, pp. 441–452, 2018.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Hans, C. Bayesian Lasso regression. *Biometrika*, 96(4): 835–845, 2009.

He, X. and Niyogi, P. Locality preserving projections. In *Advances in neural information processing systems*, pp. 153–160, 2004.

Islam, M. R., Lima, A. A., Das, S. C., Mridha, M., Prodeep, A. R., and Watanobe, Y. A comprehensive survey on the process, methods, evaluation, and challenges of feature selection. *IEEE Access*, 2022.

Jana, S., Li, H., Yamada, Y., and Lindenbaum, O. Support recovery with projected stochastic gates: Theory and application for linear models. *Signal Processing*, 213: 109193, 2023.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rkE3y85ee.

Janosi, A., Steinbrunn, W., Pfisterer, M., and Detrano, R. Heart Disease. UCI Machine Learning Repository, 1988. DOI: 10.24432/C52P4X.

Kohavi, R. and John, G. H. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.

Kumar, V. and Minz, S. Feature selection: A literature review. *Smart Comput. Rev.*, 4:211–229, 2014.

Lengerich, B., Ellington, C. N., Rubbi, A., Kellis, M., and Xing, E. P. Contextualized machine learning. *arXiv preprint arXiv:2310.11340*, 2023.

Levy, S., Lavzin, M., Benisty, H., Ghanayim, A., Dubin, U., Achvat, S., Brosh, Z., Aeed, F., Mensh, B. D., Schiller, Y., et al. Cell-type-specific outcome representation in the primary motor cortex. *Neuron*, 107(5):954–971, 2020.

Lewis, D. D. Feature selection and feature extraction for text categorization. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*, 1992.

Li, F., Yang, Y., and Xing, E. P. From lasso regression to feature vector machine. In *Advances in Neural Information Processing Systems*, pp. 779–786, 2006.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., and Liu, H. Feature selection. *ACM Computing Surveys*, 50(6):1–45, dec 2017. doi: 10.1145/3136625. URL https://doi.org/10.1145%2F3136625.

Li, W., Feng, J., and Jiang, T. IsoLasso: a LASSO regression approach to RNA-Seq based transcriptome assembly. In *International Conference on Research in Computational Molecular Biology*, pp. 168–188. Springer, 2011.

Lianjia. Housing price in Beijing, 2017. URL https://www.kaggle.com/datasets/ruiqurm/lianjia.

Lindenbaum, O., Salhov, M., Averbuch, A., and Kluger, Y. L0-sparse canonical correlation analysis. In *International Conference on Learning Representations*, 2021a.

Lindenbaum, O., Shaham, U., Peterfreund, E., Svirsky, J., Casey, N., and Kluger, Y. Differentiable unsupervised feature selection based on a gated laplacian. *Advances in Neural Information Processing Systems*, 34:1530–1542, 2021b.

Lindenbaum, O., Aizenbud, Y., and Kluger, Y. Probabilistic robust autoencoders for outlier detection. *The Conference on Uncertainty in Artificial Intelligence (UAI)*, 2024.

Liu, Y., Chen, P.-H. C., Krause, J., and Peng, L. How to read articles that use machine learning: users' guides to the medical literature. *Jama*, 322(18):1806–1816, 2019.

Louizos, C., Welling, M., and Kingma, D. P. Learning sparse neural networks through l_0 regularization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1Y8hhg0b.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=S1jE5L5gl.

Miller, A., Foti, N., D'Amour, A., and Adams, R. P. Reducing reparameterization gradient variance. In *Advances in Neural Information Processing Systems*, pp. 3708–3718, 2017.

Peng, H., Long, F., and Ding, C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

Reunanen, J. Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research*, 3(Mar):1371–1382, 2003.

Shaham, U., Lindenbaum, O., Svirsky, J., and Kluger, Y. Deep unsupervised feature selection by discarding nuisance and correlated features. *Neural Networks*, 152: 34–43, 2022.

Shim, H., Hwang, S. J., and Yang, E. Joint active feature acquisition and classification with variable-size set encoding. *Advances in neural information processing systems*, 31, 2018.

Song, L., Smola, A., Gretton, A., Borgwardt, K. M., and Bedo, J. Supervised feature selection via dependence estimation. In *Proceedings of the 24th international conference on Machine learning*, pp. 823–830. ACM, 2007.

Song, L., Smola, A., Gretton, A., Bedo, J., and Borgwardt, K. Feature selection via dependence maximization. *Journal of Machine Learning Research*, 13(May):1393–1434, 2012.

Sristi, R. D., Mishne, G., and Jaffe, A. Disc: Differential spectral clustering of features. In *Advances in Neural Information Processing Systems*, 2022.

Stein, G., Chen, B., Wu, A. S., and Hua, K. A. Decision tree classifier for network intrusion detection with ga-based feature selection. In *Proceedings of the 43rd annual Southeast regional conference-Volume 2*, pp. 136–141. ACM, 2005.

Thompson, R., Dezfouli, A., and Kohn, R. The contextual lasso: Sparse linear models via deep neural networks, 2023.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

Yamada, Y., Lindenbaum, O., Negahban, S., and Kluger, Y. Feature selection using stochastic gates. In *International Conference on Machine Learning*, pp. 10648–10659. PMLR, 2020.

Yang, J., Lindenbaum, O., and Kluger, Y. Locally sparse neural networks for tabular biomedical data. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 25123–25153. PMLR, 17–23 Jul 2022a. URL https://proceedings.mlr.press/v162/yang22i.html.

Yang, J., Lindenbaum, O., and Kluger, Y. Locally sparse neural networks for tabular biomedical data. In *International Conference on Machine Learning*, pp. 25123–25153. PMLR, 2022b.

Yin, M., Ho, N., Yan, B., Qian, X., and Zhou, M. Probabilistic best subset selection via gradient-based optimization, 2022.

Yoon, J., Jordon, J., and van der Schaar, M. Invase: Instance-wise variable selection using neural networks. In *International Conference on Learning Representations*, 2018.

Zhu, Z., Ong, Y.-S., and Dash, M. Wrapper–filter feature selection algorithm using a memetic framework. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(1):70–76, 2007.

A. Theoretical Proofs

Theorem 1. Let $s^*(\boldsymbol{z})$ and $s'^*(\boldsymbol{z})$ represent the optimal feature selection functions in Eq. (1) and its corresponding probabilistic formulation in Eq. (2) respectively. Then $s^*(\boldsymbol{z}) = s'^*(\boldsymbol{z})$.

Proof: Firstly, the optimal solution for Eq. (1) lies in the feasible solution space for Eq. (2). Secondly, Eq. (2) attains its optimal value at the extreme values of the Bernoulli parameters $\pi(\boldsymbol{z})$. We refer the reader to (Yin et al., 2022) for detailed proof.

Theorem 2. c-STG attains an optimal risk lower or equal to the risk attained by STG.

Proof: The feasible solution space of STG is contained in the feasible solution space of c-STG as the parameters of the stochastic gates in c-STG can be any function of contextual variables $\boldsymbol{z}$, which includes a constant function, which is the case in STG. Therefore, the optimal risk attained by c-STG must be less than or equal to the optimal risk value attained by STG.

Theorem 3. In a linear regression setup, the relationship between the optimal parameters of the conditional Bernoulli stochastic gates, $\pi^*(\boldsymbol{z})$, and the optimal parameters of the non-conditional Bernoulli stochastic gates $\boldsymbol{\pi}^*_{\text{stg}}$ is given by

$$\boldsymbol{\pi}^*_{\text{stg}} = E_Z[\pi^*(\boldsymbol{z})]. \tag{7}$$

Proof: The proof proceeds in five steps.

Step 1: When we consider $L(\cdot,\cdot)$ to be a mean squared loss, the solution to the optimization problem in Eq. (6) and Eq. (2) will be given by their respective MMSE (minimum mean squared error) estimates. These are given by,

$$f_{\boldsymbol{\theta}^*}(\boldsymbol{x} \odot \boldsymbol{s'}) = E[Y|X = \boldsymbol{x}, S' = \boldsymbol{s'}] \tag{9}$$

$$f_{\boldsymbol{\theta}^*}(\boldsymbol{x} \odot s'(\boldsymbol{z})) = E[Y|X = \boldsymbol{x}, Z = \boldsymbol{z}, S'(\boldsymbol{z}) = s'(\boldsymbol{z})] \tag{10}$$

where the feature selection vector $\boldsymbol{s'}$ and contextual feature selection vector $s'(\boldsymbol{z})$ will be sampled from a constrained space.

Step 2: Relaxing $f_{\boldsymbol{\theta}}$ to be a linear function $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^T\boldsymbol{x}$, the above MMSE estimates are further reduced to

$$\begin{aligned} E[Y|X = \boldsymbol{x}, S' = \boldsymbol{s'}] &= f_{\boldsymbol{\theta}^*}(\boldsymbol{x} \odot \boldsymbol{s'}) \\ &= \boldsymbol{\theta}^{*T}(\boldsymbol{x} \odot \boldsymbol{s'}) \\ &= (\boldsymbol{x} \odot \boldsymbol{\theta}^*)^T \boldsymbol{s'} \end{aligned} \tag{11}$$

$$\begin{aligned} E[Y|X = \boldsymbol{x}, Z = \boldsymbol{z}, S'(\boldsymbol{z}) = s'(\boldsymbol{z})] &= f_{\boldsymbol{\theta}^*}(\boldsymbol{x} \odot s'(\boldsymbol{z})) \\ &= \boldsymbol{\theta}^{*T}(\boldsymbol{x} \odot s'(\boldsymbol{z})) \\ &= (\boldsymbol{x} \odot \boldsymbol{\theta}^*)^T s'(\boldsymbol{z}) \end{aligned} \tag{12}$$

Step 3: We now average the estimates given in Eq. (11) and Eq. (12) across S' and $S'|Z = \boldsymbol{z}$ respectively.

$$E[Y|X = \boldsymbol{x}] = E_{S'}(\boldsymbol{x} \odot \boldsymbol{\theta}^*)^T \boldsymbol{s'} = (\boldsymbol{x} \odot \boldsymbol{\theta}^*)^T \boldsymbol{\pi}^*_{\text{stg}} \tag{13}$$

$$\begin{aligned} E[Y|X = \boldsymbol{x}, Z = \boldsymbol{z}] &= E_{S'(\boldsymbol{z})}(\boldsymbol{x} \odot \boldsymbol{\theta}^*)^T s'(\boldsymbol{z}) \\ &= (\boldsymbol{x} \odot \boldsymbol{\theta}^*)^T \pi^*(\boldsymbol{z}) \end{aligned} \tag{14}$$

Step 4: We now project the MMSE estimate of c-STG onto the space of contextual-feature selection vectors that remain constant across contextual variables. This is given by

$$\begin{aligned} E_Z E[Y|X = \boldsymbol{x}, Z = \boldsymbol{z}] &= E_Z(\boldsymbol{x} \odot \boldsymbol{\theta}^*)^T \pi(\boldsymbol{z}) \\ &= (\boldsymbol{x} \odot \boldsymbol{\theta}^*)^T E_Z \pi^*(\boldsymbol{z}) \end{aligned} \tag{15}$$

Step 5: This solution should be equivalent to solving the STG problem as it restricts the stochastic gates to be constant over z. Comparing the two equations Eq. (13) and Eq. (15), we get $\boldsymbol{\pi}^*_{\text{stg}} = E_Z[\pi^*(\boldsymbol{z})]$.

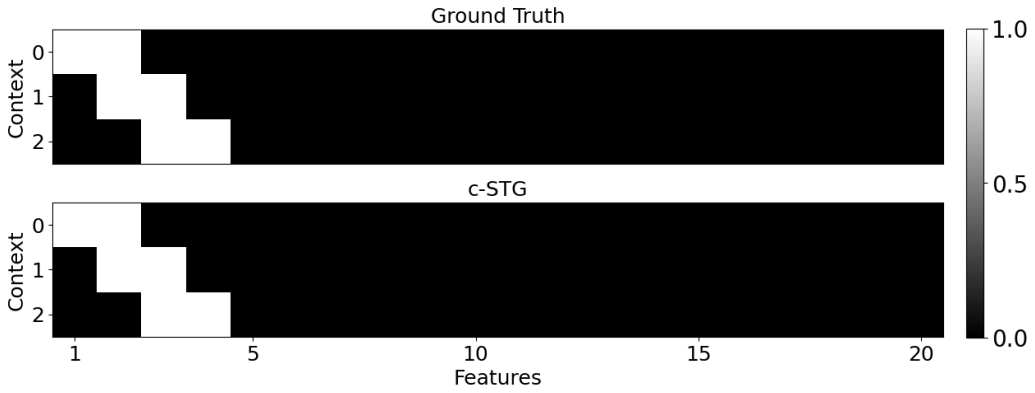

Figure 8. XOR1. The ground truth (top) feature importance as a function of z. Features identified by c-STG (bottom).

Theorem 4. In a linear regression setup, the relationship between the optimal parameters of the conditional Gaussian stochastic gates, $\mu^*(\boldsymbol{z})$, and the optimal parameters of the non-conditional Gaussian stochastic gates $\boldsymbol{\mu}^*_{\text{stg}}$ is given by

$$\boldsymbol{\mu}^*_{\text{stg}} = E_Z[\mu^*(\boldsymbol{z})]. \tag{8}$$

Proof: This follows from the proof of Theorem 3.3 by replacing Eq. (13) and Eq. (14) with the following two equations respectively:

$$E[Y|X = \boldsymbol{x}] = E_{S'}(\boldsymbol{x} \odot \boldsymbol{\theta}^*)^T \boldsymbol{s'} = (\boldsymbol{x} \odot \boldsymbol{\theta}^*)^T \boldsymbol{\mu}^*_{\text{stg}} \tag{16}$$

$$\begin{aligned} E[Y|X = \boldsymbol{x}, Z = \boldsymbol{z}] &= E_{S'(\boldsymbol{z})}(\boldsymbol{x} \odot \boldsymbol{\theta}^*)^T s'(\boldsymbol{z}) \\ &= (\boldsymbol{x} \odot \boldsymbol{\theta}^*)^T \mu^*(\boldsymbol{z}) \end{aligned} \tag{17}$$

Theorem 5. Weighted c-STG attains an optimal risk lower or equal to the risk attained by c-STG.

Proof: This follows similar to the proofs of the theorem 2. The feasible solution space of c-STG is contained in the feasible solution space of weighted c-STG as the weights of the weighted c-STG can be any function of contextual variables $\boldsymbol{z}$, which includes a constant function, which is the case in c-STG. Therefore, the optimal risk attained by weighted c-STG must be less than or equal to the optimal risk value attained by c-STG.

B. Additional Simulated Datasets

B.1. Moving XOR

Here, we validate that the c-STG method can identify context-specific informative features while learning complex non-linear prediction functions. Following synthetic examples in (Yamada et al., 2020; Yang et al., 2022b), we design a moving XOR dataset as

$$\textbf{XOR1:} \qquad \text{y}(\boldsymbol{x}, \boldsymbol{z}) = \begin{cases} x_1 \times x_2, & \text{if } \boldsymbol{z} = 0, \\ x_2 \times x_3, & \text{if } \boldsymbol{z} = 1, \\ x_3 \times x_4, & \text{if } \boldsymbol{z} = 2, \end{cases} . \tag{18}$$

We generate a data matrix X with 1500 samples and 20 features, where each entry is sampled from a fair Bernoulli distribution ($P(x_{ij} = 1) = P(x_{ij} = -1) = 0.5$). The contextual variable $\boldsymbol{z}$ is sampled uniformly from $\{0, 1, 2\}$ and the response variable y for different samples will have different subsets of significant features. In Fig. 8, we demonstrate that c-STG correctly recovers the dependence of y on the features as a function of $\boldsymbol{z}$.

Additionally, to illustrate c-STG's superiority over contextual LASSO, a context-specific feature selection technique using ℓ_1 regularization, we create a third synthetic example (XOR3), designed for linear prediction models where contextual LASSO

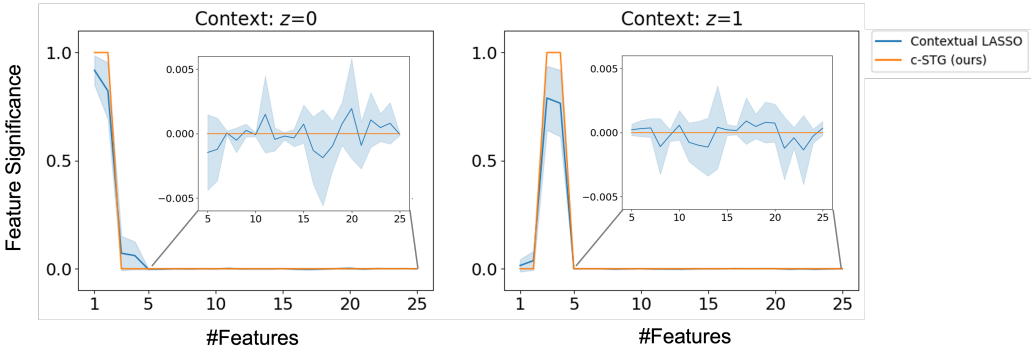

Figure 9. XOR3. Comparative feature significance between Contextual LASSO (blue) and c-STG (orange) for contexts $z = 0$ (left) and $z = 1$ (right). c-STG precisely isolates features 1 and 2 as significant for $z = 0$ and features 3 and 4 for $z = 1$, assigning no significance to other features. In contrast, Contextual LASSO indicates non-zero significance to all features, though it correctly identifies higher significance for features 1 and 2 for $z = 0$ and similarly for $z = 1$. This disparity showcases the shrinkage challenge of Contextual LASSO, which c-STG overcomes, ensuring accurate feature sparsity.

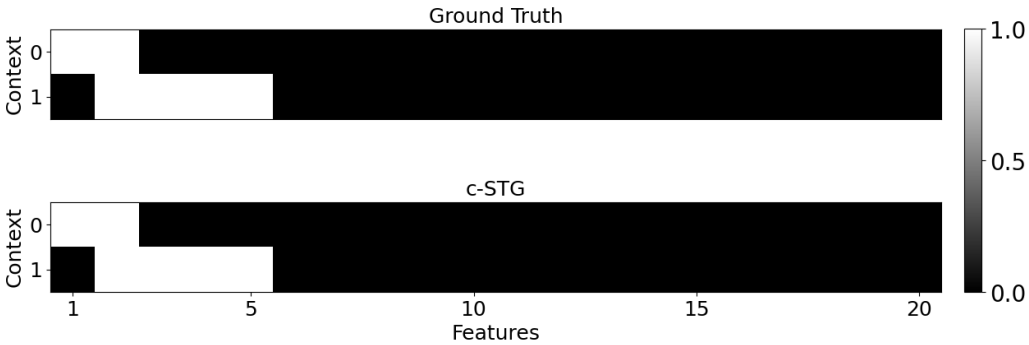

Figure 10. XOR4. The ground truth (top) feature importance as a function of z. Features identified by c-STG (bottom).

is applicable. This example aims to showcase c-STG's advantage, particularly its resistance to the shrinkage problem that plagues contextual LASSO. We construct a dataset X with 1000 samples and 25 features, drawing each entry from a normal distribution. The contextual variable $\boldsymbol{z}$ is uniformly sampled from $0, 1$. The prediction variable y is defined as follows, where different subsets of features become significant depending on $\boldsymbol{z}$'s value:

$$\textbf{XOR3:} \qquad \mathrm{y}(\boldsymbol{x}, \boldsymbol{z}) = \begin{cases} x_1 + x_2 + \eta, & \text{if } \boldsymbol{z} = 0, \\ x_3 + x_4 + \eta, & \text{if } \boldsymbol{z} = 1, \end{cases}.$$

with $\eta \sim \mathcal{N}(0, 0.25I)$. In Fig. 9, c-STG accurately identifies the significant features for each context, demonstrating its effectiveness compared to the contextual LASSO, which is hindered by coefficient shrinkage.

Furthermore, to show that c-STG can identify different important features for each context, we create another synthetic example named XOR4. Similar to XOR3, we construct a dataset X with 1000 samples and 25 features, drawing each entry from a normal distribution. The contextual variable $\boldsymbol{z}$ is uniformly sampled from $0, 1$. The prediction variable y is defined as follows, where different number of features become significant depending on $\boldsymbol{z}$'s value:

$$\textbf{XOR4:} \qquad \mathrm{y}(\boldsymbol{x}, \boldsymbol{z}) = \begin{cases} x_1 + x_2 + \eta, & \text{if } \boldsymbol{z} = 0, \\ x_3 + x_4 + x_5 + x_6 + \eta, & \text{if } \boldsymbol{z} = 1, \end{cases}.$$

with $\eta \sim \mathcal{N}(0, 0.25I)$. In Fig. 10, we demonstrate that c-STG accurately recovers the dependence of y on the features as a function of $\boldsymbol{z}$.

C. Implementation Details

In selecting model architecture, we aim to maintain simplicity, interpretability, and linearity whenever feasible. We retained the same hypernetwork and prediction model architecture across various methods when possible. For the hypernetwork, we conducted experiments with one and two hidden layers, while for the prediction networks, we explored options ranging from zero (linear) to two hidden layers. The final model architectures for each of the datasets are mentioned below in their respective paragraphs. To determine the best hyperparameters, namely the learning rate (η) and regularization coefficient (λ), we performed a grid search over the following values: $\eta \in \{1e^{-1}, 5e^{-2}, 1e^{-2}, 5e^{-3}, 1e^{-3}, 5e^{-4}, 1e^{-4}\}$ and $\lambda \in \{1, 5e^{-1}, 1e^{-1}, 5e^{-2}, 1e^{-2}, 5e^{-3}, 1e^{-3}\}$. The same set of values was used for the grid search across all the datasets. The selection of model parameters/hyperparameters was based on preventing issues like underfitting and overfitting and ensuring optimal 5-fold cross-validated performance.

XOR1: The hypernetwork $\widetilde{h}_\phi$ has two fully connected layers, with 100 and 10 neurons, respectively. We employed ReLU as the activation function after the first layer and Sigmoid activation after the last layer. For the prediction model, we used two fully connected layers, with 10 and 10 neurons, respectively, and their corresponding nonlinearities, ReLU and sigmoid.

XOR2: Both the hypernetwork $\widetilde{h}_\phi$ and the prediction model have no hidden layers with inputs directly projecting to outputs. However, the hypernetwork has a Sigmoid activation, and the prediction model has a ReLU activation function in their output layers.

XOR3: For the XOR3 dataset, the architecture of hypernetwork and prediction model is same as that of XOR2 except that the prediction model has no activation function.

XOR4: For the XOR4 dataset, the architecture of hypernetwork and prediction model is same as that of XOR3.

MNIST: For the MNIST dataset, the configuration of the hypernetwork $\widetilde{h}_\phi$ differs, possessing two layers with 64 and 128 neurons. ReLU and Sigmoid activation functions are used for these two layers, respectively. The prediction model consists of layers with 128 and 64 neurons, coupled with the ReLU followed by Sigmoid activations.

Housing: We employed a hypernetwork architecture similar to the one used in the MNIST example. However, we used a linear model to predict house prices in this case. We divided the data into 10 train, validation, and test splits and conducted a grid search on η and λ for each split. We then selected the hyperparameters with the best validation performance.

Heart disease The full list of features includes: chest pain type (cp), resting blood pressure (trestbps), serum cholesterol levels (chol), fasting blood sugar (fbs), resting electrocardiographic results (restecg), maximum heart rate achieved (thalach), exercise-induced angina (exang), ST depression induced by exercise (oldpeak), the slope of the peak exercise ST segment (slope), the number of major vessels colored by fluoroscopy (ca), and thalassemia (thal). Among these features, cp, restecg, and slope are categorical and, hence, are converted to one hot encoding vector for feeding them into the prediction model. The hypernetwork and the prediction model have one hidden layer with 1000 neurons and 10 neurons, respectively, with sigmoid activation on their last layer. We conducted a grid search on η and λ from the range of values mentioned in the MNIST example.

Neuroscience: For the first data set (context is time), a weighted c-stg model was trained using 5-fold cross-validation with separate sets of trials for training, parameter tuning, and testing. We conducted a grid search on different learning rates and regularization parameters ranging from 0.0005 to 0.1 and for the number of hidden units of the hypernetwork ranging from 10 to 1000 units. Overall, for the hypernetwork, we used a single hidden layer of 10 units, a learning rate of 0.0005, and a regularization weight of 0.05, where the prediction network consisted of six hidden layers of 500, 300, 100, 50, and 2 units. For the second data set (flavor is context), a c-stg model was trained using 5-fold cross-validation with the same mechanism for parameter tuning. Overall, for the hypernetwork, we used a single layer with 1000 units, a learning rate of 0.05, and a regularization weight of 0.001, where the prediction network consisted of six hidden layers of 400, 300, 100, 50, and 2 units.

Table 3. Comparison of feature selection techniques with a fixed number of significant features across all the models. The number of significant features selected for each dataset is reported in the second row. **Bold** indicates best performance and underline indicates second-best.

	XOR1	**XOR2**	**MNIST**	**Heart disease**	**Housing**
	3	2	218	7	5
	Accuracy (%)	R^2 score	Accuracy (%)	Accuracy (%)	R^2 score
LASSO	50.07 (1.43)	0.1760 (0.0156)	73.70 (0.48)	73.70 (0.48)	0.1979 (0.0016)
with context	50.02 (1.09)	0.1761 (0.0156)	73.79 (0.48)	82.00 (3.48)	0.2251(0.0015)
STG	65.73 (1.46)	0.1838 (0.0248)	91.47 (0.67)	84.18 (2.92)	0.2136 (0.0026)
with context	65.77 (0.87)	0.1873 (0.0320)	98.34 (0.07)	83.84 (1.31)	0.2231 (0.0022)
Contextual LASSO	50.53 (0.60)	0.7311 (0.0090)	97.08 (0.06)	82.85 (4.83)	0.4535 (0.0034)
AFS	70.83 (19.06)	0.7572 (0.0187)	93.91 (0.99)	83.39 (1.98)	0.3482 (0.0224)
INVASE	50.07 (0.70)	0.3547 (0.0118)	65.27 (4.69)	84.18 (1.72)	0.2502 (0.0060)
c-STG (Ours)	**100** (0)	0.8739 (0.0107)	98.65 (0.04)	86.89 (3.18)	0.3970 (0.0074)
Weighted c-STG (Ours)	**100** (0)	**0.9956** (0.0008)	**98.69** (0.06)	**89.23** (1.72)	**0.5308** (0.0052)

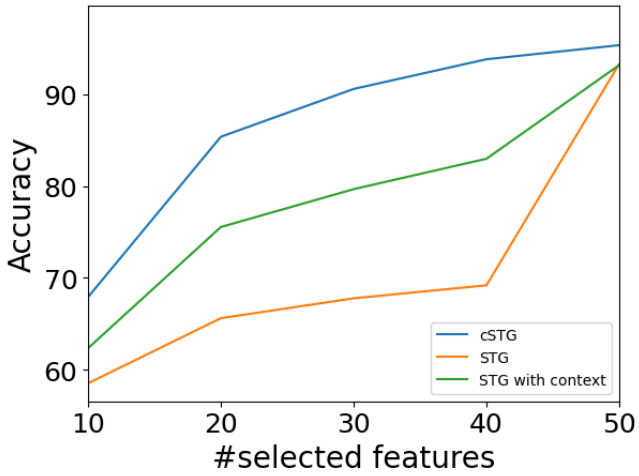

Figure 11. Comparing the performance variations of cSTG, STG, and STG with context with respect to different sparsity levels on the MNIST dataset.

D. Additional Experiments

Sparsity-Driven Performance Analysis: We evaluate the performance of c-STG as a function of the number of selected features in the model, i.e., sparsity. Our analysis uses the MNIST example, as described in section 4. To perform this analysis, we varied the regularization coefficient, λ, in a way that controlled the sparsity level within the range of 10 to 50. It is worth noting that for STG with context, we only counted the explanatory features corresponding to pixels and not the eight additional contextual features (one-hot encoding) representing the angle. Figure 11 illustrates the performance of three models, c-STG, STG, and STG, with context as the sparsity level increases. This demonstrates that c-STG has superior performance compared to both STG with context and STG alone, indicating that c-STG, with its contextual information and sparse representation, outperforms the other models for varying numbers of selected features.

Comparison of Discrete Contextual Model Training: When focusing on deriving insights from specific contextual features, training separate models for each specific context is a viable strategy. However, it is crucial to understand that this strategy is only applicable when the contextual variables are categorical and not continuous. Although straightforward, this method brings its own challenges, particularly when the dataset at hand is limited in size. In such cases, each model only gets to train on a subset of data corresponding to its designated context. We shed light on the consequences of this approach using the modified MNIST dataset, as elaborated in Section 4. Our exploration involved comparing two methodologies: 1) individually training eight models for each context using STG and 2) employing our c-STG to create a unified model for all

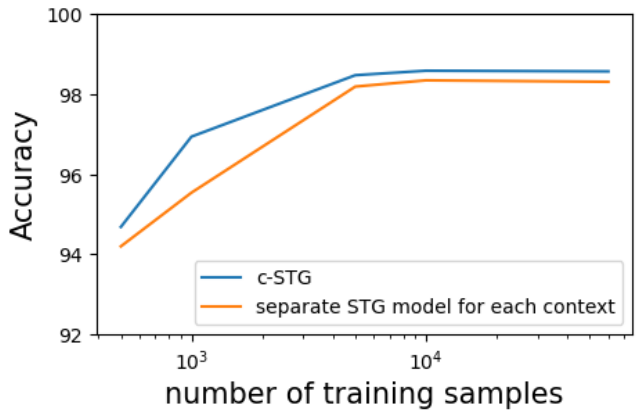

Figure 12. Comparison between training separate models for different values of categorical context variables and a single c-STG model on the modified MNIST dataset in section 4, as the number of training samples varies.

contexts. To assess the efficiency of these approaches, we varied the number of training samples and captured the results in Fig. 12.

Comparatively, our c-STG model offers a compelling advantage. Instead of segregating data by context, it utilizes the entirety of the dataset, benefiting from the diverse range of contexts. A direct contrast in terms of model parameters reveals that while training separate models for each discrete context demands 876,688 parameters, the c-STG model operates with a lean set of 218,834 parameters. The reduced parameter count in c-STG does not compromise performance; in fact, it underscores c-STG's prowess in delivering enhanced outcomes with fewer parameters.

Feature Selection Across Models: We present a comparative analysis detailing the number of features chosen by each method whose prediction performance is reported in Table 1. A notable observation is that c-STG has fewer selected features and ranks either as the best or second-best in performance evaluation, as demonstrated in Table 1. The prediction model (with context) and CEN are excluded, as no sparsity constraint is imposed on the explanatory features to perform the prediction task.

E. Compute details

We trained all networks using CUDA-accelerated PyTorch implementations on a NVIDIA Quadro RTX8000 GPU.