Evidential Softmax for Sparse Multimodal Distributions in Deep Generative Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Many applications of generative models rely on the marginalization of their high-dimensional output probability distributions. Normalization functions that yield sparse probability distributions can make exact marginalization more computationally tractable. However, sparse normalization functions usually require alternative loss functions for training because the log-likelihood can be undefined for sparse probability distributions. Furthermore, many sparse normalization functions often collapse the multimodality of distributions. In this work, we present *ev-softmax*, a sparse normalization function that preserves the multimodality of probability distributions. We derive its properties, including its gradient in closed-form, and introduce a continuous family of approximations to *ev-softmax* that have full support and can thus be trained with probabilistic loss functions such as negative log-likelihood and Kullback-Leibler divergence. We evaluate our method on a variety of generative models, including variational autoencoders and auto-regressive models. Our method outperforms existing dense and sparse normalization techniques in distributional accuracy and classification performance. We demonstrate that *ev-softmax* successfully reduces the dimensionality of output probability distributions while maintaining multimodality.

1 Introduction

Learning deep generative models over discrete probability spaces has enabled state-of-the-art performance across tasks in computer vision [1], natural language processing [2]–[4], and robotics [5]–[7]. The output probability distributions of deep generative models are often obtained from a neural network by the *softmax* transformation as a natural extension of multinomial logistic regression. Since the *softmax* function has full support, the logarithm of the output probabilities is well-defined, which is important for common probabilistic loss functions such as the negative log likelihood, cross entropy, and Kullback-Leibler (KL) divergence.

Several applications highlight the importance of achieving sparsity in high dimensional dense distributions to make downstream tasks computationally feasible and perhaps even interpretable [8]–[11]. In discrete latent variable models, for instance, calculating the posterior probabilities exactly requires marginalization over all possible latent variable assignments, which is intractable for large latent spaces [9], [10]. Stochastic approaches to circumvent this include the score function estimator [12], [13], which suffers from high variance, and continuous relaxations of the latent variable, such as straight-through or Gumbel-Softmax [14], [15], which introduce bias.

Alternatively, high-dimensional latent spaces can be tractably marginalized using normalization functions that produce sparse probability distributions. Several sparse alternatives to *softmax* have been proposed, including *sparsemax* [8], α-*entmax* [16], and *sparsehourglass* [17]. However, these approaches often collapse the multimodality of distributions, resulting in unimodal probability

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

distributions [10]. When marginalizing over discrete latent spaces, maintaining this multimodality is crucial for downstream applications such as image generation, where the latent space is multimodal in nature [1], and machine translation, where multiple words can be valid translations and require context to distinguish between the optimal word choice [9]. Furthermore, these approaches either have to construct alternative loss functions to avoid the undefined log-likelihood of zero-valued output probabilities [8], [16]–[18], or they are applied as a post-hoc transformation at test time [10]. As a result, the latent distributions obtained through these existing sparse normalization functions and their resulting posteriors cannot be trained directly.

The post-hoc sparsification procedure introduced by Itkina *et al.* [10] provides a method for obtaining sparse yet multimodal distributions at test time. We generalize this method to a sparse normalization function termed *evidential softmax* (*ev-softmax*) which can be applied during both training and test time. We define continuous approximations of this function that have full support and are thus compatible with typical loss terms, such as negative log-likelihood and KL divergence. By using this full-support approximation of *ev-softmax* to train generative models over discrete probability spaces, we optimize these models directly for objective functions with probabilistic interpretations instead of optimizing a proxy model in the case of post-hoc sparsification or nonprobabilistic objective functions, such as the *sparsemax* loss function [8]. We evaluate these discrete generative models at test time with the *ev-softmax* function to obtain sparse yet multimodal latent distributions.

Contributions: This paper proposes a strategy for training neural networks with sparse output probability distributions that is compatible with negative log-likelihood and KL-divergence computations. We generalize the evidential sparsification procedure developed by Itkina *et al.* [10] to the *ev-softmax* function, and prove properties of *ev-softmax* and its continuous approximation. We facilitate exact marginalization over the discrete latent space of VAE models during training. Our approach outperforms existing dense and sparse normalization approaches in distributional accuracy and classification performance across tasks in image generation and machine translation.

2 Background

Notation: We denote scalars, vectors, matrices, and sets as a, $\mathbf{a}$, $\mathbf{A}$, and $\mathcal{A}$, respectively. The vectorized Kronecker delta $\boldsymbol{\delta}_i$ is defined to be 1 at the ith index and 0 at all the other indices. The K-dimensional simplex is written as $\Delta^K := \{\boldsymbol{\xi} \in \mathbb{R}^{K+1} : \mathbf{1}^T\boldsymbol{\xi} = 1\}$. The KL divergence of probability distribution $\mathbf{p}$ from distribution $\mathbf{q}$ is denoted $\mathrm{KL}(\mathbf{p} \mid\mid \mathbf{q})$.

2.1 Discrete Variational Autoencoders

To model a data generating distribution, we consider data $\mathbf{x} \in \mathbb{R}^d$, categorical input variable $\mathbf{y} \in \mathcal{Y}$, and categorical latent variable $\mathbf{z} \in \mathcal{Z}$. In the VAE model, we condition $\mathbf{x}$ on $\mathbf{z}$ as $p_\theta(\mathbf{x} \mid \mathbf{z})$. The conditional variational autoencoder (CVAE) [19] extends this framework by allowing for a generative framework controlled by input $\mathbf{y}$. The latent variable $\mathbf{z}$ is conditioned on the input $\mathbf{y}$ as $p_\theta(\mathbf{z} \mid \mathbf{y})$.

The VAE estimates parameters ϕ and an auxiliary distribution $q_\phi(\mathbf{z} \mid \mathbf{x})$ to calculate the evidence lower bound (ELBO) of the log-likelihood for the observed data [20]. Likewise, the CVAE estimates parameters ψ of an auxiliary distribution $q_\psi(\mathbf{z} \mid \mathbf{x}, \mathbf{y})$ to obtain the ELBO [19]. The lower bounds are

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x} \mid \mathbf{z}) - \mathrm{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}) \mid\mid p_\theta(\mathbf{z})) \tag{1}$$

$$\log p_\theta(\mathbf{x} \mid \mathbf{y}) \geq \mathbb{E}_{q_\psi(\mathbf{z}|\mathbf{x},\mathbf{y})} \log p_\theta(\mathbf{x} \mid \mathbf{z}) - \mathrm{KL}(q_\psi(\mathbf{z} \mid \mathbf{x}, \mathbf{y}) \mid\mid p_\theta(\mathbf{z} \mid \mathbf{y})). \tag{2}$$

Maximizing the VAE ELBO using gradient-based methods involves computing the term $\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \log p_\phi(\mathbf{x} \mid \mathbf{z})$ in Eq. (1), which requires marginalization over the auxiliary distribution $q_\phi(\mathbf{z} \mid \mathbf{x})$. Many Monte Carlo approaches estimate this quantity without explicit marginalization over all possible latent classes $\mathbf{z} \in \mathcal{Z}$. The score function estimator (SFE) is an unbiased estimator which uses the identity $\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x} \mid \mathbf{z}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} p_\theta(\mathbf{x} \mid \mathbf{z}) \nabla_\phi \log q_\phi(\mathbf{z} \mid \mathbf{x})$ to estimate the gradient through samples drawn from $q_\phi(\mathbf{z}|\mathbf{x})$. However, the SFE is prone to high variance, and approaches in practice generally require variance reduction methods such as baselines, control variates, and Rao-Blackwellization techniques [21]. Alternatively, a low-variance gradient estimate can be obtained by the *reparameterization trick*, which separates the parameters of the distribution from the source of stochasticity [20]. The Gumbel-Softmax and Straight-Through reparameterizations rely on a continuous relaxation of the discrete latent space [14], [15]. In contrast with these stochastic

approaches, our method, along with *sparsemax* [8], α-*entmax* [16], and sparsehourglass [17], focuses on obtaining sparse auxiliary distributions $q_\phi(\mathbf{z} \mid \mathbf{x})$ and $q_\psi(\mathbf{z} \mid \mathbf{x}, \mathbf{y})$ to enable exact marginalization over all latent classes $\mathbf{z} \in \mathcal{Z}$ for which the auxiliary distribution has nonzero probability mass.

2.2 Normalization Functions

A common normalization function that maps vectors $\mathbf{v} \in \mathbb{R}^K$ to probability distributions in Δ^{K-1} is the *softmax* function:

$$\textsc{Softmax}(\mathbf{v})_k \propto e^{v_k}. \tag{3}$$

One limitation of this function is the resulting probability distribution always has full support, *i.e.* $\textsc{Softmax}(\mathbf{v})_k \neq 0$ for all $\mathbf{v}$ and k. Thus, if the distributions q_ϕ in Eq. (1) and Eq. (2) are outputs of the *softmax* function, then calculating the expectations over q_ϕ exactly requires marginalizing over all one-hot vectors $\mathbf{z} \in \mathbb{R}^K$, which can be computationally intractable when K is large [9], [10].

Recent interest in sparse alternatives has led to the development of many new sparse normalization functions [8], [10], [16]–[18]. The *sparsemax* function projects inputs to the simplex to achieve a sparse distribution [8], and the α-*entmax* family of functions generalizes the *softmax* and *sparsemax* functions through a learnable parameter α which tunes the level of sparsity [16]. *Sparsehourglass* is another generalization of *sparsemax* with a controllable level of sparsity [17]. Despite the controls, we find that these methods achieve sparsity at the expense of collapsing valid modes.

2.3 Post-Hoc Evidential Sparsification

Of particular interest is a sparse normalization function that maintains multimodality in the latent distribution, which is relevant to models where the output distribution of $p_\theta(\mathbf{z} \mid \mathbf{y})$ is possibly multimodal, such as in image generation or translation tasks. The post-hoc sparsification method introduced by Itkina *et al.* [10] calculates evidential weights as affine transformations of an input feature vector and interprets these weights as evidence either for a singleton latent class $\{z_k\}$ or its complement $\overline{\{z_k\}}$ depending on the sign of the weights. This interpretation, with origins in evidential theory [22], [23], allows for the distinction between conflicting evidence and lack of evidence, which is normally not possible with ordinary multinomial logistic regression. Conflicting evidence can be interpreted as multimodality whereas lack of evidence can correspond to sparsity [10].

For a given dataset and model, the resulting post-hoc sparsification is as follows. A CVAE is trained on a dataset using *softmax* normalization and the ELBO in Eq. (2). At test time, for a given query $\mathbf{y}$ from the dataset, the learned weights of the last layer of the encoder $\hat{\boldsymbol{\beta}} \in \mathbb{R}^{(J+1)\times K}$, and the corresponding output of the last hidden layer $\phi(\mathbf{y}) \in \mathbb{R}^J$, the evidential weights are:

$$w_k = \hat{\boldsymbol{\beta}}_{0k} - \frac{1}{K}\sum_{\ell=1}^{K}\hat{\boldsymbol{\beta}}_{0\ell} + \sum_{j=1}^{J}\left[\phi_j\left(\hat{\boldsymbol{\beta}}_{jk} - \frac{1}{K}\sum_{\ell=1}^{K}\hat{\boldsymbol{\beta}}_{j\ell}\right)\right] \tag{4}$$

for each $k \in \{1, \ldots, K\}$. Evidence in support of a latent class z_k is then defined as $w_k^+ = \max(0, w_k)$ and evidence against it as $w_k^- = \max(0, -w_k)$. The resulting sparse distribution is constructed as:

$$p(z_k) \propto \mathbb{1}\{m_k \neq 0\}e^{w_k}, \tag{5}$$

where

$$m_k = e^{-w_k^-}\left(e^{w_k^+} - 1 + \prod_{\ell \neq k}\left(1 - e^{-w_\ell^-}\right)\right) \tag{6}$$

for each $k \in \{1, \ldots, K\}$. Further information can be found in Appendix A.

3 The Evidential Softmax Transformation

Although the evidential sparsification procedure is successful at providing sparse yet multimodal normalizations of scores, it can only be performed post-hoc, without the possibility of being used at training time. Furthermore, it requires $O(JK^2)$ computations to calculate all the weights for each input, with J hidden features in the last layer and K latent classes.

We propose a transformation, which we call *evidential softmax* (*ev-softmax*), on inputs $\mathbf{v} \in \mathbb{R}^K$. We claim that this transformation is equivalent to that of the post-hoc evidential sparsification procedure with the advantages of having a simple closed form and requiring only $O(K)$ computations.

Let $\overline{v} = \frac{1}{K}\sum_{i=1}^{K} v_i$ be the arithmetic mean of the elements of $\mathbf{v}$. We define our transformation as,

$$\textsc{EvSoftmax}(\mathbf{v})_k \propto \mathbb{1}\{v_k \geq \overline{v}\}e^{v_k}. \tag{7}$$

Note that the post-hoc sparsification procedure in Section 2.3 takes input vectors $\phi \in \mathbb{R}^J$ and the weights $\hat{\beta} \in \mathbb{R}^{J\times K}$ of the neural network's last layer whereas the *ev-softmax* function takes input $\mathbf{v} = \hat{\beta}\phi \in \mathbb{R}^K$. We formalize the equivalence of *ev-softmax* with the post-hoc sparsification procedure as follows:

Theorem 3.1. *For any input vector $\phi \in \mathbb{R}^J$, weight matrix $\hat{\boldsymbol{\beta}} \in \mathbb{R}^{J\times K}$, define the score vector $\boldsymbol{v} = \hat{\boldsymbol{\beta}}\phi$ and variables $w_k, w_k^{\pm}, m_k$, and $p(z_k)$ as in Eqs.* (4) *to* (6) *for all $k \in \{1, \ldots, K\}$. If the probability distribution of $\boldsymbol{v}$ is non-atomic, then* $\textsc{EvSoftmax}(\boldsymbol{v})_k = p(z_k)$ *for each k.*

The proof of this theorem uses the following two lemmas:

Lemma 3.2. *Let $\overline{v} = \frac{1}{K}\sum_{i=1}^{K} v_i$. Then $v_k - \overline{v} = w_k$ for all $k \in \{1, \ldots, K\}$.*

Proof. The result follows from algebraic manipulation of Eq. (4). Full details are in Appendix B. □

Lemma 3.3. *For m_k as defined in Eq.* (6)*, $\mathbb{1}\{m_k \neq 0\} = \mathbb{1}\{v_k > \overline{v}\}$ for all $k \in \{1, \ldots, K\}$.*

Proof. Observe that the condition $\mathbb{1}\{m_k = 0\} = 1 - \{m_k \neq 0\}$ is equivalent to the conjunction of the following two conditions: 1) $w_k \leq 0$ and 2) there exists some $\ell \neq k$ such that $w_\ell \geq 0$. From Lemma 3.2, these statements are equivalent to 1) $v_k \leq \overline{v}$ and 2) there exists some $\ell \neq k$ such that $v_\ell \geq \overline{v}$. Note that if $v_k \leq \overline{v}$, then either $v_\ell = \overline{v}$ or there must exist some $\ell \neq k$ such that $v_\ell > \overline{v}$. In both cases, there exists $\ell \neq k$ such that $v_\ell \geq \overline{v}$. Thus, we see that $v_k \leq \overline{v}$ implies there exists some $\ell \neq k$ such that $v_\ell \geq \overline{v}$. Then we conclude that $\mathbb{1}\{m_k = 0\} = \mathbb{1}\{v_k \leq \overline{v}\}$ and the result follows. □

We briefly sketch the proof of Theorem 3.1 and refer the reader to Appendix B for the full derivation. Lemma 3.2 shows that $e^{w_k} \propto e^{v_k}$. Combining Lemma 3.3 with the non-atomicity of the distribution of $\mathbf{v}$ yields that $\mathbb{1}\{m_k \neq 0\} = \mathbb{1}\{v_k \geq \overline{v}\}$. Eq. (7) follows from these two results.

3.1 Properties

Table 1 gives a list of *softmax* alternatives, desirable properties of a normalization function, and which properties are satisfied by each function as inspired by the analysis of Laha *et al.* [17]. Refer to Appendix C for proofs of these properties. These results show that *ev-softmax* satisfies the same properties as the *softmax* transformation. While idempotence can be useful, these normalizations are generally only applied once, and in this regime the idempotence does not provide any further guarantees. Likewise, scale invariance is less applicable in practice because the scale of the inputs to normalization functions is often one of the learned features of data, and the scale generally provides a measure of confidence in the predictions.

The Jacobians of *softmax* and *ev-softmax* have analogous forms (Appendix C):

$$\frac{\partial \textsc{Softmax}(\mathbf{v})_i}{\partial v_j} = \textsc{Softmax}(\mathbf{v})_i(\delta_{ij} - \textsc{Softmax}(\mathbf{v})_j) \tag{8}$$

$$\frac{\partial \textsc{EvSoftmax}(\mathbf{v})_i}{\partial v_j} = \textsc{EvSoftmax}(\mathbf{v})_i(\delta_{ij} - \textsc{EvSoftmax}(\mathbf{v})_j). \tag{9}$$

For the Jacobian of the proposed *ev-softmax* function, there are two cases to consider. First, when either or both of $\textsc{EvSoftmax}(\mathbf{v})_i$ and $\textsc{EvSoftmax}(\mathbf{v})_j$ are exactly 0, we have $\frac{\partial \textsc{EvSoftmax}(\mathbf{v})_i}{\partial v_j} = 0$. Intuitively, this occurs because classes that have a probability of 0 do not contribute to the gradient and also remain 0 under small local changes in the input. Second, when both $\textsc{EvSoftmax}(\mathbf{v})_i$ and $\textsc{EvSoftmax}(\mathbf{v})_j$ are nonzero, the partial derivative is equivalent to that of *softmax* over the classes that have nonzero probabilities. Hence, optimizing models using the *ev-softmax* function should yield similar behaviors to those using the *softmax* function.

Table 1: Summary of properties satisfied by normalization functions. Here, ✓ denotes that the property is satisfied. The $*$ denotes that the property is satisfied conditional on either the input or some parameter that is independent of the input. Other properties satisfied by all considered normalization functions include invariance with respect to permutation of the coordinates and existence of the Jacobian. These properties are proven in Appendix C.

Property	Softmax	Sparsemax [8]	Sparsehourglass [17]	**Ev-Softmax (Ours)**
Idempotence		✓	✓	
Monotonic	✓	✓	✓	✓
Translation Invariance	✓	✓	✓*	✓
Scale Invariance			✓*	
Full Domain	✓	✓	✓	✓
Lipschitz	1	1	$1 + \frac{1}{Kq}$	1^*

3.2 Training-time Modification

Since *ev-softmax* outputs a sparse probability distribution, the negative log likelihood loss term cannot be directly computed. To address this, we modify *ev-softmax* during training time as follows:

$$\textsc{EvSoftmax}_{\text{train},\epsilon}(\mathbf{v})_i \propto (\mathbb{1}\{v_i \geq \overline{v}\} + \epsilon)e^{v_k} \tag{10}$$

for some small ϵ. In practice, we set ϵ to 10^{-6}.

The KL divergence with the above training modification takes the same form as the ϵ-KL used for overcoming sparsity in language modeling [24], [25]. As ϵ approaches 0, the gradient of the negative log likelihood loss term approaches a form that mirrors that of the *softmax* function:

$$\nabla_{\mathbf{v}} \log\left[\textsc{Softmax}(\mathbf{v})_i\right] = \boldsymbol{\delta}_i - \textsc{Softmax}(\mathbf{v}) \tag{11}$$

$$\lim_{\epsilon \to 0} \nabla_{\mathbf{v}} \log\left[\textsc{EvSoftmax}_{\text{train},\epsilon}(\mathbf{v})_i\right] = \boldsymbol{\delta}_i - \textsc{EvSoftmax}(\mathbf{v}), \tag{12}$$

for any finite-dimensional, real-valued vector $\mathbf{v}$. A derivation of Eq. 12 is given in Appendix D.

While the Gumbel-Softmax relaxation also enables gradient computations through discrete latent spaces, Eq. (12) establishes the convergence of the gradient as well as the closed-form solution to the limit of the gradient as the relaxation goes to 0. This convergence applies uniquely to the *ev-softmax* function. Note that the limit of the log-likelihood of a distribution generated by the training-time modification (e.g. $\lim_{\epsilon \to 0} \log[\textsc{EvSoftmax}_{\text{train},\epsilon}(\mathbf{v})_i]$) is undefined, but Eq. (12) shows that the limit of the *gradient* of the log-likelihood is defined.

4 Experiments

We demonstrate the applicability of our proposed strategies on the tasks of image generation and machine translation. We compare our method to the *softmax* function as well as sparse normalization functions including *sparsemax* [8] and *entmax*-1.5 [16]. Since post-hoc evidential sparsification procedure [10] is functionally equivalent but can only be used at test time, we do not compare against it. Experiments indicate that *ev-softmax* achieves higher distributional accuracy and classification performance than *softmax*, *sparsemax* and *entmax*-1.5 by better balancing the objectives of sparsity and multimodality. All experiments were performed on a single local NVIDIA GeForce GTX 1070 or Tesla K40c GPU.

4.1 Conditional Variational Autoencoder

To gain intuition for the *ev-softmax* function, we consider the multimodal task of generating digit images for *even* and *odd* queries ($y \in \{\text{even}, \text{odd}\}$) on MNIST. We use a CVAE with $|\mathcal{Z}| = 10$ latent classes. Since the true conditional prior distribution given $y \in \{\text{even}, \text{odd}\}$ is uniform over 5 digits, this task benefits from both sparsity and multimodality of a normalization function.

Model: We follow the CVAE architecture and training procedure of Itkina *et al.* [10] for this task. For each of the four normalization functions we consider (*ev-softmax*, *sparsemax*, *entmax*-1.5, and *softmax*), we train an encoder, which consists of two multi-layer perceptrons (MLPs), and a decoder. The

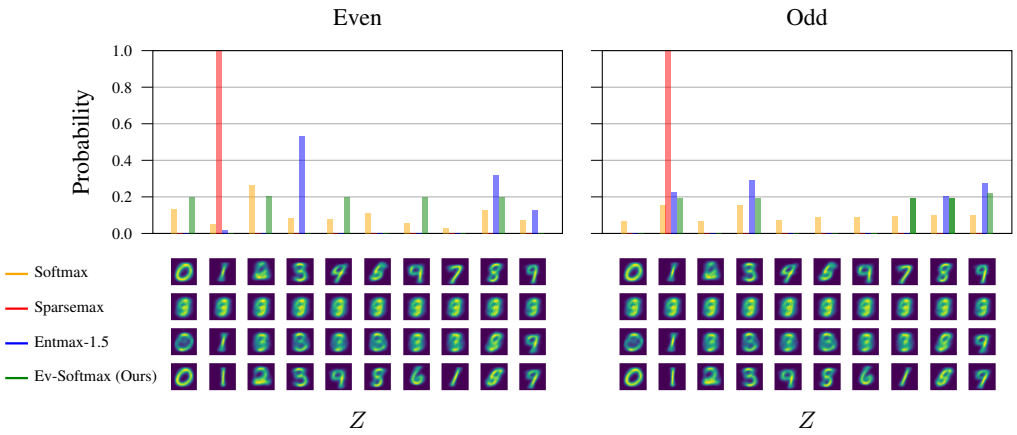

Figure 1: Using the proposed normalization function *ev-softmax* at training time yields a more accurate distribution on the MNIST dataset than *softmax*, *sparsemax*, and *entmax*-1.5 baselines. The horizontal axis shows the decoded image for each latent class and vertical axis shows the probability mass.

first MLP of the encoder receives an input query y and outputs a categorical prior distribution $p_\theta(\mathbf{z} \mid y)$ over the latent variable $\mathbf{z}$ generated by the normalization function. The second MLP takes as input a feature vector $\mathbf{x}$ in addition to the query y, and outputs the probability distribution for the posterior $q_\psi(\mathbf{z} \mid \mathbf{x}, y)$, also generated using the same normalization function. The decoder consists of a single MLP that takes as input the latent variable $\mathbf{z}$ and generates an output image $\mathbf{x}$. Each model was trained for 20 epochs to minimize the ELBO of Eq. (2). Further experimental details are in Appendix E.

Results and discussion: Since $|\mathcal{Z}| = 10$, the ground truth prior $p_\theta(\mathbf{z} \mid y)$ is a uniform distribution over 5 of the 10 latent classes. In Fig. 1, we show the decoded images of the latent classes trained using each of the normalization functions, as well as the corresponding prior distributions over these latent classes for even and odd inputs. Using *ev-softmax* at training yields the most sensible images for both even and odd inputs. The CVAE trained with *ev-softmax* learns a distribution with 5 non-zero probability modes for both even and odd inputs. All of the digits are represented by the latent classes of the *ev-softmax* distribution, and they are generated by the even and odd priors almost perfectly. In contrast, the *softmax* distribution results in nonzero probability mass assigned to all latent classes for both inputs, producing a distributional accuracy that is qualitatively worse than that of *ev-softmax*. The decoded images of *entmax*-1.5 show that *entmax*-1.5 the posterior latent space to fewer than 10 dimensions, reducing the representational power of the generative model. Over 10 random seeds and a hyperparameter grid-search, *sparsemax* collapsed both the prior and posterior distributions to a single mode.

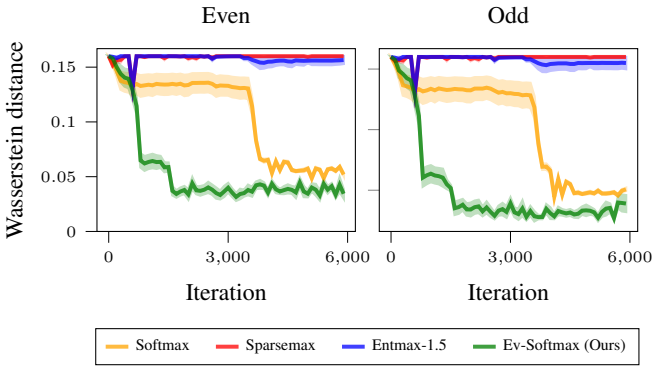

Figure 2: Wasserstein distance between the predicted categorical prior and the ground truth prior distributions. Our method outperforms *softmax*, *sparsemax*, and *entmax*-1.5 over 10 random seeds. Lower is better.

To quantify the distributional accuracy of each CVAE model, we compute the Wasserstein distance (Appendix E) between the model's predicted categorical prior $p(\mathbf{z} \mid y)$ and the ground truth prior, shown in Fig. 2. We see that *ev-softmax* converges the fastest and obtains the lowest distance from the ground truth distribution. *Sparsemax* and *entmax*-1.5, on the other hand, do not allow the model to converge toward the true prior as they collapse the prior and posterior distributions. The model trained with *softmax* converges more slowly and to a higher distance from the ground truth distribution because it lacks the sparsity that *ev-softmax* offers, resulting in worse distributional accuracy.

4.2 Semi-supervised Variational Autoencoder

To evaluate our method in the semi-supervised setting, we consider the semi-supervised VAE of Kingma *et al.* [26]. We evaluate the model on the MNIST dataset, using 10% of the labeled data and treating the remaining data as unlabeled. In this regime, we expect a higher level of distributional accuracy where sparsity remains a key ingredient and multimodality reflects uncertainty in the generative model.

Table 2: The results of the semi-supervised VAE on MNIST show that *ev-softmax* yields the best semi-supervised performance. We report the mean and standard error of the accuracy and number of decoder calls for 10 random seeds.

Method	Accuracy (%)	Decoder calls
Monte Carlo		
SFE	$94.75 \pm .002$	1
SFE+	$96.53 \pm .001$	2
NVIL	$96.01 \pm .002$	1
Sum&Sample	$96.73 \pm .001$	2
Gumbel	$95.46 \pm .001$	1
ST Gumbel	$86.35 \pm .006$	1
Marginalization		
Softmax	$96.93 \pm .001$	10
Sparsemax	$96.87 \pm .001$	1.01 ± 0.01
Entmax-1.5	$97.20 \pm .001$	1.01 ± 0.01
Ev-softmax	$\mathbf{97.30 \pm .001}$	1.64 ± 0.01

Model: We follow the architecture and training procedure of Correia *et al.* [9]. We model the joint probability of an MNIST image $\mathbf{x} \in \mathbb{R}^d$, a continuous latent variable $\mathbf{h} \in \mathbb{R}^n$, and a discrete random variable $\mathbf{z} \in \{\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_{10}\}$ as $p_\theta(\mathbf{x}) = p_\theta(\mathbf{x} \mid \mathbf{z}, \mathbf{h})p(\mathbf{z})p(\mathbf{h})$. We assume $\mathbf{h}$ has an n-dimensional standard Gaussian prior and $\mathbf{z}$ a uniform prior.

We compare *ev-softmax* to other sparse normalization functions, including *sparsemax* and *entmax*-1.5. Since we follow the procedure of Correia *et al.* [9], we also report the results they obtained for stochastic methods. Unbiased gradient estimators that they report include SFE with a moving average baseline; SFE with a self-critic baseline (SFE+) [27]; neural variational inference and learning (NVIL) with a learned baseline [28]; and sum-and-sample, a Rao-Blackwellized version of SFE [21]. They also evaluate biased gradient estimators including Gumbel-Softmax and straight-through Gumbel-Softmax (ST Gumbel). Each model was trained for 200 epochs.

Results and discussion: Table 2 demonstrates that using *ev-softmax* yields the highest classification accuracy of models using either Monte Carlo methods or marginalization. Compared to other sparse normalization functions, *ev-softmax* maintains higher multimodality in the output distribution, reducing the latent space by 84% on average. In particular, both *sparsemax* and *entmax*-1.5 reduce the latent space by almost 90%, but at the cost of decreased accuracy. This suggests that in this semi-supervised setting, maintaining multimodality in the output distribution is important for distributional accuracy, which benefits overall performance.

4.3 Vector-Quantized Variational Autoencoder (VQ-VAE)

We demonstrate the performance of our approach on a much larger latent space by training a Vector Quantized-Variational Autoencoder (VQ-VAE) model on *tiny*ImageNet for image generation. The VQ-VAE learns a discrete latent variable embedded within a continuous vector space [1]. We use the smaller *tiny*ImageNet dataset due to computational constraints.

Table 3: Downstream classification performance (top-5 accuracy and top-10 accuracy) and average number of nonzero latent classes on 5000 images (25 samples $\times$ 200 classes) sampled from VQ-VAE models shows that our method yields images with the highest top-5 and top-10 accuracy scores.

Method	Acc-5	Acc-10	K
Softmax	38.4	48.8	512
Sparsemax	40.0	52.0	46
Entmax-1.5	38.4	49.2	90
Ev-softmax	**43.6**	**55.6**	77

Model: We train the VQ-VAE architecture in two stages. First the VQ-VAE model is trained assuming a uniform prior over the discrete latent space. For each normalization method, we train an autoregressive prior over the latent space from which we sample. We consider a latent space of 32×32 discrete latent variables with $K = 512$ classes each. We use the same PixelCNN as in Itkina *et al.* [10] for the prior. For each normalization function, we train a PixelCNN prior using the normalization function to generate the output probability over the latent space. To

evaluate the accuracy of our VQ-VAE models, we train a Wide Residual Network classifier on *tiny*ImageNet and calculate its classification accuracy on the images generated by the VQ-VAE models. Hyperparameters and further experimental details are given in Appendix E.

Results and discussion: Table 3 shows that *ev-softmax* is able to generate images with the most resemblance to the ground truth. Our method reduces the number of latent classes by 85% on average, more than the 83% of *entmax*-1.5 but less than the 91% of *sparsemax*. The sparse methods all outperform *softmax*, suggesting that sparsity improves the downstream task of image generation by discarding irrelevant latent classes. *Ev-softmax* outperforms other sparse methods in accuracy. Since the VQ-VAE samples directly from the output distribution of the PixelCNN prior, the probabilistic nature of this downstream task suggests that optimizing the prior for log-likelihood allows the model to learn a prior distribution that achieves better performance on the downstream task.

4.4 Machine Translation

We evaluate our method in the setting of machine translation models. Attention-based translation models generally employ dense attention weights over the entire source sequence while traditional statistical machine translation systems are usually based on a lattice of sparse phrase hypotheses [9]. Thus, sparse attention offers the possibility of more interpretable attention distributions which avoid wasting probability mass on irrelevant source words. Furthermore, language modeling often requires the context of multiple words to correctly identify dependencies within an input sequence [2]. Hence, we investigate the impact of the multimodality of our proposed normalization function on the quality of translations.

Table 4: BLEU score and number of nonzero elements in the attention on the EN→DE corpus of IWSLT 2014. We report the mean and standard error over 5 random seeds. Our method outperforms all baselines in BLEU score and maintains a higher number of nonzero attended elements than other sparse normalization functions.

Method	BLEU	# attended
Softmax	29.2 $\pm$0.06	39.5 $\pm$11.5
Sparsemax	29.0 $\pm$0.05	2.3 $\pm$0.54
Entmax-1.5	29.2 $\pm$0.07	4.1 $\pm$1.3
Ev-softmax	**29.4 $\pm$0.05**	8.2 $\pm$1.3

Model: We train transformer models for each of the sparse normalization functions using the OpenNMT Neural Machine Translation Toolkit [29]. Each normalization function is used in the self-attention layers. We use byte pair encoding (BPE) to encode rare and unknown words and ensure an open vocabulary [30]. Due to computational constraints, we finetune transformer models on the English-German translation dataset from IWSLT 2014 [31]. These models were pretrained on the EN-DE corpus from WMT 2016 [32]. We finetune for 40000 iterations. Hyperparameters and further experimental details are given in Appendix E.

Results and discussion: As shown in Table 4, our method obtains the highest BLEU score of the considered normalization functions while reducing the average number of words attended to 8.2. In comparison with the other normalization functions, ours maintains the highest number of attended source words. This supports the notion that attending over a subset of the source words benefits performance, but the performance of *sparsemax* and *entmax*-1.5 could be hurt by attending over too few words. For short sentences, *ev-softmax* maintains nonzero attention over the entire sentence, as Fig. 3 demonstrates. This allows for the model to identify relationships between words in the entire source sequence, such as tense and conjugation, enabling more accurate translation.

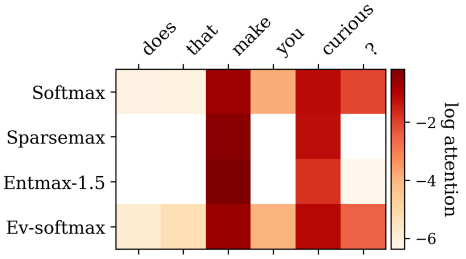

Figure 3: Attention weights produced by each normalization method for the first word of the translated sentence. *Sparsemax* and *entmax*-1.5 both attend over two words while *ev-softmax* and *softmax* attend over the entire source. The resulting predictions are *macht* (*ev-softmax*), *Ist* (*sparsemax*), and *Machen* (*entmax*-1.5, *softmax*). The ground truth translation is *macht sie das neugierig?*. Although the attention distributions of *ev-softmax* and *softmax* are similar in this example, *ev-softmax* still significantly sparsifies attention away from superfluous inputs.

5 Related Work

Softmax Alternatives: There is considerable interest in alternatives to the *softmax* function that provide greater representational capacity [33], [34] and improved discrimination of features [35]. *Sparsemax* [9], its generalization α-*entmax* [16], and its *maximum a posteriori* adaptation SparseMAP [36]

introduce sparse normalization functions and corresponding loss functions. While the α-*entmax* family of functions contains a parameter α to control the level of sparsity, the functions are all sparse for $\alpha > 1$ with no known continuous approximation with full support. Therefore, these methods are all trained with alternative loss functions that do not directly represent the likelihood of the data. Another controllable sparse alternative to *softmax* is *sparsehourglass* [17]. We baseline against only *sparsemax* and *entmax*-1.5 as more research has been conducted on these two methods and *entmax*-1.5 is semantically similar to *sparsehourglass*. In contrast, we develop a normalization function which has a continuous family of approximations that can be used during training time with the negative log-likelihood and KL divergence loss terms. This flexibility enables our method to substitute the *softmax* transformation in any context and model.

Discrete Variational Autoencoders: Recent advancements in VAEs with discrete latent spaces have introduced strategies for learning discrete latent spaces to tackle a wide-ranging set of tasks including image generation [1], [37], video prediction [38], and speech processing [39]. For discrete VAEs, the gradient of the evidence lower bound (ELBO) usually cannot be calculated directly [14]. While the SFE [12], [13] enables Monte Carlo-based approaches to estimate the stochastic gradient of the ELBO, it suffers from high variance and is consequently slow to converge. Continuous relaxation via the Gumbel-Softmax distribution allows for an efficient and convenient gradient calculation through discrete latent spaces [14], [15], but the gradient estimation is biased and requires sampling during training. Our work, in contrast, enables direct marginalization over a categorical space without sampling. Similar to our work, Correia *et al.* [9] apply *sparsemax* to marginalize discrete VAEs efficiently. While we also generate a sparse distribution over the latent space as in Correia *et al.* [9], we balance sparsity with the multimodality of the distribution. Prior work on evidential sparsification [10] focused on a post-hoc method for discrete CVAEs, while we extend this method to more classes of generative models and develop a modification to apply the method at training time.

Sparse Attention: In machine translation, sparse attention reflects the intuition that only a few source words are expected to be relevant for each translated word. The global attention model considers all words on the source side for each target word, which is potentially prohibitively expensive for translating longer sequences [40]. The *softmax* transformation has been replaced with sparse normalization functions, such as the constrained *sparsemax* operation [11] and its generalization, α-*entmax* [16], [41]. Like our work, both *sparsemax* and *entmax* can substitute *softmax* in the attention layers of machine translation models. Prior work on sparse normalization functions for attention layers has focused on global attention layers in recurrent neural network models. In contrast, we study the effect of incorporating these sparse normalization functions in transformers, where they are used in both global attention and self-attention layers.

6 Conclusion

We present a novel method to train deep generative models with sparse output and latent probability distributions. We show that our method satisfies many of the same theoretical properties as softmax and other sparse alternatives to softmax. By balancing the goals of maintaining multimodality and sparsifying the distributions, our method yields better distributional accuracy and classification performance than baseline methods. Future work can involve training large computer vision and machine translation models from scratch with our method and examining its efficacy at a larger scale on more complex downstream tasks.

7 Broader Impact

Our work focuses on a method to train generative models that output sparse probability distributions. We apply our method in the domains of image generation and machine translation. Our work has the potential to improve the explainability and introspection of generative models. Furthermore, our work aims to reduce the computational cost of training generative models, which are used in unsupervised and semi-supervised approaches to learning. Although our empirical validation shows that our method can improve the accuracy of generated distributions, the use of sparse probability distributions can also amplify any inherent bias present in the data or modeling assumptions. One limitation of our work is that we do not provide explicit theoretical guarantees on the conditions for yielding an output probability of zero for a class. We hope that our work motivates further research into improving the accuracy of generative modeling and reducing bias of both models and data.

References

[1] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 6306–6315.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[3] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[4] Z. Yang, W. Chen, F. Wang, and B. Xu, "Improving neural machine translation with conditional sequence generative adversarial nets," *arXiv preprint arXiv:1703.04887*, 2017.

[5] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *European Conference on Computer Vision (ECCV)*, 2020.

[6] J. Co-Reyes, Y. Liu, A. Gupta, B. Eysenbach, P. Abbeel, and S. Levine, "Self-consistent trajectory autoencoder: Hierarchical reinforcement learning with trajectory embeddings," in *International Conference on Machine Learning (ICML)*, PMLR, 2018, pp. 1009–1018.

[7] V. Fortuin, M. Hüser, F. Locatello, H. Strathmann, and G. Rätsch, "SOM-VAE: Interpretable discrete representation learning on time series," in *International Conference on Learning Representations (ICLR)*, 2019.

[8] A. Martins and R. Astudillo, "From softmax to sparsemax: A sparse model of attention and multi-label classification," in *International Conference on Machine Learning (ICML)*, 2016, pp. 1614–1623.

[9] G. Correia, V. Niculae, W. Aziz, and A. Martins, "Efficient marginalization of discrete and structured latent variables via sparsity," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020.

[10] M. Itkina, B. Ivanovic, R. Senanayake, M. J. Kochenderfer, and M. Pavone, "Evidential sparsification of multimodal latent spaces in conditional variational autoencoders," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020.

[11] C. Malaviya, P. Ferreira, and A. F. Martins, "Sparse and constrained attention for neural machine translation," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018, pp. 370–376.

[12] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[13] P. W. Glynn, "Likelihood ratio gradient estimation for stochastic systems," *Communications of the ACM*, vol. 33, no. 10, pp. 75–84, 1990.

[14] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-Softmax," in *International Conference on Learning Representations (ICLR)*, 2017.

[15] C. J. Maddison, A. Mnih, and Y. W. Teh, "The Concrete distribution: A continuous relaxation of discrete random variables," in *International Conference on Learning Representations (ICLR)*, 2017.

[16] B. Peters, V. Niculae, and A. F. Martins, "Sparse sequence-to-sequence models," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019, pp. 1504–1519.

[17] A. Laha, S. A. Chemmengath, P. Agrawal, M. Khapra, K. Sankaranarayanan, and H. G. Ramaswamy, "On controllable sparse alternatives to softmax," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018, pp. 6422–6432.

[18] V. Niculae, A. Martins, M. Blondel, and C. Cardie, "SparseMAP: Differentiable sparse structured inference," in *International Conference on Machine Learning (ICML)*, PMLR, 2018, pp. 3799–3808.

[19] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015, pp. 3483–3491.

[20] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *International Conference on Learning Representations (ICLR)*, 2014.

[21] R. Liu, J. Regier, N. Tripuraneni, M. Jordan, and J. Mcauliffe, "Rao-Blackwellized stochastic gradients for discrete distributions," in *International Conference on Machine Learning (ICML)*, PMLR, 2019, pp. 4023–4031.

[22] A. P. Dempster, "A generalization of Bayesian inference," *Classic works of the Dempster-Shafer Theory of Belief Functions*, pp. 73–104, 2008.

[23] T. Denoeux, "Logistic regression, neural networks and dempster–shafer theory: A new perspective," *Knowledge-Based Systems*, vol. 176, pp. 54–67, 2019.

[24] B. Bigi, "Using Kullback-Leibler distance for text categorization," in *European Conference on Information Retrieval (ECIR)*, Springer, 2003, pp. 305–319.

[25] J. Hullman, N. Diakopoulos, and E. Adar, "Contextifier: Automatic generation of annotated stock visualizations," in *SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2013, pp. 2707–2716.

[26] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2014, pp. 3581–3589.

[27] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7008–7024.

[28] A. Mnih and K. Gregor, "Neural variational inference and learning in belief networks," in *International Conference on Machine Learning (ICML)*, PMLR, 2014, pp. 1791–1799.

[29] G. Klein, F. Hernandez, V. Nguyen, and J. Senellart, "The OpenNMT neural machine translation toolkit: 2020 edition," in *Conference of the Association for Machine Translation in the Americas (AMTA)*, 2020, pp. 102–109.

[30] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016, pp. 1715–1725.

[31] M. Cettolo, M. Federico, L. Bentivogli, N. Jan, S. Sebastian, S. Katsuitho, Y. Koichiro, and F. Christian, "Overview of the IWSLT 2017 evaluation campaign," in *International Workshop on Spoken Language Translation*, 2017, pp. 2–14.

[32] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, A. J. Yepes, P. Koehn, V. Logacheva, C. Monz, *et al.*, "Findings of the 2016 Conference on Machine Translation," in *Conference on Machine Translation (WMT)*, 2016, pp. 131–198.

[33] S. Kanai, Y. Yamanaka, Y. Fujiwara, and S. Adachi, "Sigsoftmax: Reanalysis of the softmax bottleneck," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 2018, pp. 286–296, 2018.

[34] R. Memisevic, C. Zach, M. Pollefeys, and G. E. Hinton, "Gated softmax classification," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 23, pp. 1603–1611, 2010.

[35] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks.," in *International Conference on Machine Learning (ICML)*, PMLR, vol. 2, 2016, p. 7.

[36] V. Niculae and M. Blondel, "A regularized framework for sparse and structured neural attention," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 3340–3350.

[37] A. Razavi, A. van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with VQ-VAE-2," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 14 837–14 847.

[38] J. Walker, A. Razavi, and A. v. d. Oord, "Predicting video with VQVAE," *arXiv*, 2021.

[39] C. Gârbacea, A. van den Oord, Y. Li, F. S. Lim, A. Luebs, O. Vinyals, and T. C. Walters, "Low bit-rate speech coding with vq-vae and a wavenet decoder," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019, pp. 735–739.

[40] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015, pp. 1412–1421.

[41] B. Peters, V. Niculae, and A. F. Martins, "Sparse sequence-to-sequence models," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019, pp. 1504–1519.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Sections 3 and 4
   (b) Did you describe the limitations of your work? [Yes] See Section 7
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 7
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 3
   (b) Did you include complete proofs of all theoretical results? [Yes] They are attached in supplemental materials.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplemental materials.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See supplemental materials.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 4
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]