Evidential Disambiguation of Latent Multimodality in Conditional Variational Autoencoders

Masha Itkina, Boris Ivanovic, Ransalu Senanayake, Mykel J. Kochenderfer, Marco Pavone
Department of Aeronautics and Astronautics
Stanford University
{mitkina, borisi, ransalu, mykel, pavone}@stanford.edu

1 Introduction

An important obstacle to establishing trust in autonomous systems such as self-driving cars is the interpretability of the learned neural network representations. Deep generative models, such as variational autoencoders [1], learn a distillation of the underlying data distribution to generate new samples. Understanding this encoding and its structure can aid in interpreting the reasoning behind the neural network output. Adding to the complexity of the interpretability task, many real-world problems such as video frame prediction [2] or human behavior prediction [3, 4, 5] are represented by multimodal distributions. Multimodality arises in these scenarios from multiple accurate possibilities for the future (e.g., a pedestrian may turn right or left given the same trajectory history). To address prediction multimodality, the conditional variational autoencoder (CVAE) was developed [6]. At test time, the CVAE samples from a multimodal latent encoding of a query label to generate diverse data. The choice of a discrete latent encoding over a continuous one has been shown to encourage multimodal predictions [3, 4, 7] as well as interpretability [8]. However, prohibitively large discrete latent spaces are often required to accurately learn complex data distributions, causing difficulties in interpretability.

We propose a methodology that effectively reduces the latent space, while maintaining the multimodality in the latent distribution. We refer to this dimensionality reduction without loss of information as *disambiguating the latent space*. At test time, the encoder distribution for a CVAE is commonly parameterized by the softmax function. A drawback to the softmax transformation is that uncertainty is distributed across all the available discrete classes since, by definition, the softmax function cannot set a probability to zero (though it can become negligible).

Evidential theory, also known as Dempster-Shafer Theory (DST), can be considered as a generalization of Bayesian probability theory for modeling uncertainty [9]. DST defines a belief mass function over the power set of possible hypotheses, allowing DST to differentiate lack of information (e.g., an uninformative prior) from conflicting information (e.g., multimodality). We use conflicting information to identify multimodal discrete latent classes in trained CVAEs, thus, performing post hoc analysis on the network.

Our contributions are as follows:

1. We introduce a methodology for disambiguating the discrete latent space within a CVAE using evidential theory by removing the latent classes that do not directly receive evidence from the input features.
2. We present a proof-of-concept of our algorithm that shows improved CVAE performance using the reduced latent space distribution on MNIST and Fashion MNIST [10].
3. We experimentally show that our method provides a more accurate distribution over the latent encoding, not only with fewer training iterations but also with fewer training samples.

4th workshop on Bayesian Deep Learning (NeurIPS 2019), Vancouver, Canada.

2 Neural Networks as Evidential Classifiers

Evidential theory distinguishes lack of information from conflicting information [9] making it appealing for handling uncertainty in machine learning tasks. Denoeux [11] recently showed that under a set of assumptions, the softmax transformation is equivalent to the Dempster-Shafer fusion of belief masses. This insight facilitates the use of evidential theory in multi-class machine learning classification paradigms. We include a short overview of evidential theory in Appendix A drawing from the summary by Denoeux [11].

Evidential Classifiers DST considers a discrete set of hypotheses or classes. Let the set of K allowable classes be $Z = \{z_1, \ldots, z_K\}$. A belief mass function, m, is defined over the power set of classes, such that $m(A) \in [0, 1]$ for $A \subseteq Z$ and $\sum_{A \subseteq Z} m(A) = 1$. We also enforce that $m(\emptyset) = 0$, since the allowable classes are exhaustive.

Denoeux shows that all classifiers that transform a linear combination of features through the softmax function can be formulated as evidential classifiers. Each feature can be represented as an elementary piece of evidence in support of a class or its complement. The softmax function then fuses these pieces of evidence to form class probabilities given the input. The weights and features that serve as arguments to the softmax function can be used to compute the corresponding DST belief mass function. The belief mass function provides an additional degree of freedom compared to Bayesian probabilities in separating lack of evidence from conflicting evidence.

Following the notation of Denoeux, we define a feature as the last hidden layer output in a neural network, denoted by $\phi(y_i) \in \mathbb{R}^J$ where y_i is the input training data sample to the network. The evidential weights are assumed to be affine transformations of each feature ϕ_j by construction,

$$w_{jk} = \beta_{jk}\phi_j(y_i) + \alpha_{jk}, \tag{1}$$

where $\alpha_{jk})$ and β_{jk} are parameters [11]. An assumption is made that the evidence supports either a singleton class $\{z_k\}$ when $w_{jk} > 0$ ($w_{jk}^+ = \max(0, w_{jk})$) or its complement $\overline{\{z_k\}}$ when $w_{jk} < 0$ ($w_{jk}^- = \max(0, -w_{jk})$), resulting in $w_{jk}^+ - w_{jk}^- = w_{jk}$. Denoeux arrives at the corresponding mass function at the output of the softmax layer by assuming a simple mass construction (defined in Appendix A) and using Dempster's rule for belief mass fusion,

$$m(\{z_k\}) = Ce^{-w_k^-}\left(e^{w_k^+} - 1 + \prod_{l \neq k}\left(1 - e^{-w_l^-}\right)\right) \tag{2}$$

$$m(A) = C\left(\prod_{z_k \notin A}\left(1 - e^{-w_k^-}\right)\right)\left(\prod_{z_k \in A} e^{-w_k^-}\right), \quad A \subseteq Z, |A| > 1, \tag{3}$$

where C is a normalization constant, $w_k^- = \sum_{j=1}^J w_{jk}^-$, and $w_k^+ = \sum_{j=1}^J w_{jk}^+$. It can be shown that this mass function is equivalent to the softmax distribution in Bayesian probability space [11].

Evidential Disambiguation The α_{jk} and β_{jk} parameters are not uniquely defined due to the extra degree of freedom provided by the belief mass. Denoeux selects the parameters that maximize the Least Commitment Principle, which is analogous to maximum entropy in information theory [12].

In contrast, we choose our parameters with the goal of having the singleton mass function in Eq. (2) identify only the classes that receive direct evidence towards them. We observe that if $w_k^+ = 0$ and at least for one other class $l \neq k$, $w_l^- = 0$, then $m(\{z_k\}) = 0$. Intuitively, this insight means that if a class k has no direct evidence in its support and another class l does have supporting evidence, then the singleton mass is set to zero. This result only occurs if at least one of w_k^+ and w_k^- is zero such that $w_k^+ - w_k^- = w_k$, which does not hold in the original formulation by Denoeux [11]. Thus, we construct an evidential weight w_{jk} that does not depend on j, enforcing this requirement:

$$w_{jk} = \frac{1}{J}\left(\beta_{0k} + \sum_{j=1}^J \beta_{jk}\phi_j(y_i)\right). \tag{4}$$

We provide further details in Appendix B and show that the corresponding α_{jk} and β_{jk}parameters

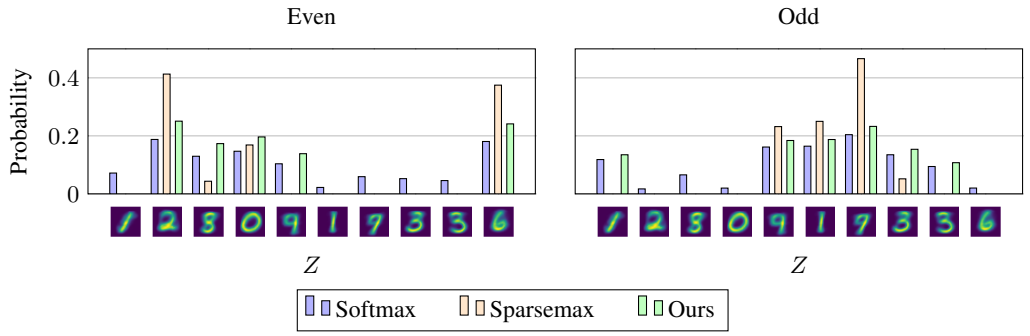


Figure 1: Our proposed filtered distribution (green) is compared to the softmax (blue) and sparsemax (orange) distributions on the MNIST dataset. The horizontal axis depicts decoded latent classes. Our method reduces the dimensionality of the relevant latent space without removing valid latent classes.

satisfy the equivalency constraints with the softmax transformation in Appendix C. We posit that filtering out the classes with zero singleton mass values does not result in loss of information, and imposes a more concentrated distribution output.

3 Experiments

The following section outlines the experiments performed to validate our approach and draws insights from the results. All experiments were performed on an NVIDIA GeForce GTX 1070 GPU.

CVAE Model We demonstrate our methodology for disambiguating multimodal discrete latent spaces in a naive CVAE architecture illustrated in Appendix D. The encoder and decoder networks consist of two-layer MLPs. During training, the Gumbel-Softmax distribution was used to backpropagate ELBO [6] loss gradients through the discrete latent space [8, 13].

At test time, only the target conditioning class y serves as input to the prior encoder MLP. The latent space is sampled from the learned $p(z \mid y)$, and passed through the decoder to generate a corresponding feature x'. Since the CVAE parameterizes the latent distribution through a softmax function at test time, we can directly use the theory of evidential classifiers. The evidence allocated to multiple singleton latent classes indicates *conflict* between them. In the context of a CVAE, we posit that conflict is directly correlated with latent space multimodality. High conflict between a subset of latent classes would indicate distinct, multimodal latent features encoded by the network.

The softmax probability distribution from the encoder is filtered by removing the probabilities for the latent classes with zero singleton mass values $m(\{z_k\})$ in Eq. (2) and then renormalized. Thus, we reduce the dimensionality of the latent space by providing a more concentrated latent class distribution to the decoder, improving neural network performance at test time.

Data For demonstration, we consider the multimodal task of generating digit images for 'even' or 'odd' input target labels. Thus, the labels y consist of either even or odd. We choose $K = 10$ for the dimensionality of the latent space. In the ideally learned network, this label choice results in a 5-modal distribution when conditioned on one of the binary labels. For instance, conditioning on the even class should produce a uniform distribution over the encoded digits: 0, 2, 4, 6, and 8. Parallel experimental results on the Fashion MNIST dataset are included in Appendix G.

Results Qualitative and quantitative experimental results show improved CVAE performance for the latent spaces filtered using the proposed method over standard softmax distributions and the popular class-reduction technique termed sparsemax [14].

Fig. 1 shows a comparison of our proposed method, softmax, and sparsemax on MNIST. The horizontal axis depicts the decoded image for each latent class. Although the CVAE successfully learns a multimodal latent encoding, the softmax distribution has non-negligible probabilities associated with the incorrect latent classes z_k for each input y, resulting in imperfect sampling from the CVAE at test time as shown in Appendix E. To remedy this problem, we consider both sparsemax [14] and

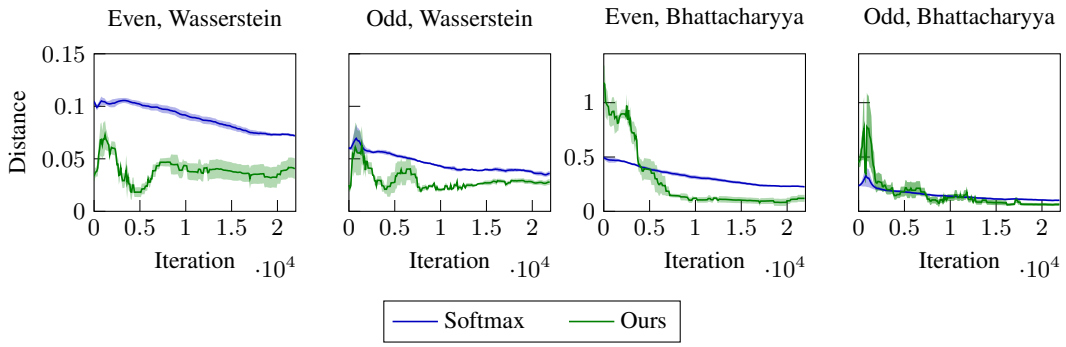


Figure 2: Results for the softmax and filtered distributions across training iterations on MNIST. The Wasserstein and Bhattacharyya distances are measured against a heuristic ground truth probability distribution defined in Appendix F.

our proposed evidential filtration technique. Our filtered distribution selects an almost perfect set of the correct latent classes given an input label. The only error made by the filtered distribution is the selection of the 9 image (5th latent class) for the even input. This error can be explained by the relatively high softmax probability assigned to the 9 image for both the 'even' and 'odd' input classes. The key insight to our approach is that it performs only as well as the quality of the representation learned by the neural network, with the benefit of extracting richer information than softmax. In contrast, sparsemax results in undesirably more aggressive filtering than our method. It removes the correct latent classes of 1 and 3 for the 'odd' input. Thus, our more conservative, yet effective filtration is a compelling latent space disambiguation technique, particularly when considering safety-critical applications.

Furthermore, we quantitatively validate that our filtered distribution achieves better test time performance with fewer training iterations and with less data. Fig. 2 shows the test set performance of the filtered distribution as the network is trained on MNIST. We demonstrate the robustness of the filtered distribution to fewer training iterations in its ability to extract more accurate distributional information from the neural network earlier in the training process than the softmax distribution. Our methodology outperforms the softmax baseline outside of standard error when the softmax distribution assigns non-negligible probability mass towards incorrect latent classes (e.g., for the even input class). We note that for the Bhattacharyya distance metric on the MNIST dataset, the filtered distribution underperforms the baseline up to the first 5,000 iterations. However, our filtered distribution can only perform as well as the learned network. During the beginning of the training process, the latent classes have not been learned well, and the neural network weights are close to random. The randomness in the weights causes the filtered latent classes to fluctuate, resulting in spikes in performance at early iterations. We provide details on our evaluations metrics and further experiments on MNIST and Fashion MNIST in Appendix F and Appendix G.

4 Conclusions

We present a fully analytical methodology for post hoc latent space reduction in CVAEs. The proposed filtered distribution outperforms the ubiquitous softmax distribution in experiments, extracting richer information with fewer training iterations and less training data. Experiments show that our distribution remains conservative by not filtering out viable latent classes, making it a compelling disambiguation technique, particularly when considering safety-critical applications. As future work, we plan to investigate the performance improvement obtained by using the proposed distribution on high-dimensional latent spaces in more complex tasks, such as behavior and video frame prediction.

Acknowledgments

The authors would like to thank Dr. Boris Kirshtein for his advice and assistance. Toyota Research Institute ("TRI") provided funds to assist the authors with their research but this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

References

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

[2] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine, "Stochastic adversarial video prediction," *arXiv*, 2018.

[3] E. Schmerling, K. Leung, W. Vollprecht, and M. Pavone, "Multimodal probabilistic model-based planning for human-robot interaction," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–9.

[4] B. Ivanovic, E. Schmerling, K. Leung, and M. Pavone, "Generative modeling of multimodal multi-human behavior," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE/RSJ, 2018.

[5] B. Ivanovic and M. Pavone, "The Trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs," *arXiv*, 2018.

[6] 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.

[7] T. M. Moerland, J. Broekens, and C. M. Jonker, "Learning multimodal transition dynamics for model-based reinforcement learning," in *29th Benelux Conference on Artificial Intelligence*, 2017, p. 362.

[8] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-Softmax," *arXiv*, 2016.

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

[10] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," *arXiv*, 2017.

[11] T. Denoeux, "Logistic regression, neural networks and Dempster-Shafer theory: A new perspective," *Knowledge-Based Systems*, 2019.

[12] P. Smets, "Belief functions: the disjunctive rule of combination and the generalized Bayesian theorem," *International Journal of Approximate Reasoning*, vol. 9, no. 1, pp. 1–35, 1993.

[13] 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.

[14] 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.

[15] G. Shafer, *A mathematical theory of evidence*. Princeton University Press, 1976, vol. 42.

[16] P. Smets, "Decision making in a context where uncertainty is represented by belief functions," in *Belief Functions in Business Decisions*. Springer, 2002, pp. 17–61.

[17] B. R. Cobb and P. P. Shenoy, "On the plausibility transformation method for translating belief function models to probability models," *International Journal of Approximate Reasoning*, vol. 41, no. 3, pp. 314–330, 2006.

[18] T. Denoeux, "Introduction to belief functions," 4th School on Belief Functions and their Applications, 2017.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[20] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," in *22nd SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1135–1144.

[21] D. A. Melis and T. Jaakkola, "Towards robust interpretability with self-explaining neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018, pp. 7775–7784.

[22] A. Kosiorek, H. Kim, Y. W. Teh, and I. Posner, "Sequential attend, infer, repeat: Generative modelling of moving objects," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018, pp. 8606–8616.

[23] S. A. Eslami, N. Heess, T. Weber, Y. Tassa, D. Szepesvari, G. E. Hinton *et al.*, "Attend, infer, repeat: Fast scene understanding with generative models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016, pp. 3225–3233.

[24] M. Tsang, H. Liu, S. Purushotham, P. Murali, and Y. Liu, "Neural interaction transparency (NIT): Disentangling learned interactions for improved interpretability," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018, pp. 5804–5813.

[25] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016, pp. 2172–2180.

[26] J. D. Olden and D. A. Jackson, "Illuminating the "black box": A randomization approach for understanding variable contributions in artificial neural networks," *Ecological Modelling*, vol. 154, no. 1-2, pp. 135–150, 2002.

[27] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018, pp. 3179–3189.

[28] W. Duch and Ł. Itert, "A posteriori corrections to classification methods," in *Neural Networks and Soft Computing*. Springer, 2003, pp. 406–411.

A Evidential Theory

Mass Functions DST [15] considers a discrete set of hypotheses or, equivalently, classes. Let the complete finite set of K allowable classes be denoted as $Z = \{z_1, \ldots, z_K\}$. A mass function, m, is defined over the power set of classes, such that $m(A) \in [0, 1]$ for $A \subseteq Z$ and $\sum_{A \subseteq Z} m(A) = 1$. In the classification setting, $m(\emptyset) = 0$ is enforced, since the allowable classes are assumed to be exhaustive. The mass function quantifies a piece of evidence in support of or against some subset of classes. A vacuous mass function encodes complete lack of information pertaining to the exhaustive set of classes, that is $m(Z) = 1$. When the only non-zero mass values are over singleton sets, the belief mass is an approximation to the Bayesian probability mass function. Two independent sources of evidence represented by belief masses can be combined through Dempster's rule to generate a fused mass function as follows,

$$(m_1 \oplus m_2)(A) = \frac{1}{1-\kappa} \sum_{B \cap C = A} m_1(B) m_2(C), \qquad \forall A \subseteq Z, A \neq \emptyset \text{ and } (m_1 \oplus m_2)(\emptyset) = 0 \tag{5}$$

where $\kappa = \sum_{B \cap C = \emptyset} m_1(B) m_2(C)$ is the degree of conflict between two mass functions.

Denoeux [11] makes the assumption that individual input features form a simple mass function for each allowable class to later show the equivalence with machine learning classifiers. The simple mass function and the corresponding weight of evidence are defined such that [15],

$$m(A) = s, \quad m(Z) = 1 - s, \quad w = -\ln(1 - s) \tag{6}$$

where A and Z are the only non-zero elements of the mass function m such that $A \subset Z$ with $A \neq \emptyset$. $s \in [0, 1]$ is referred to as the degree of support in A.

Plausibility Transformation DST masses can be reduced to estimated probabilities through either the pignistic [16] or plausibility [17] transformations. The equivalence of DST mass fusion and the softmax function [11] is shown under the plausibility transformation, defined as,

$$p(z_k) = \frac{pl(z_k)}{\sum_{l=1}^{K} pl(z_l)}, \quad \text{where} \quad pl(z_k) = \sum_{B : z_k \in B} m(B), \quad \forall B \subseteq Z \text{ and } k = \{1, \ldots, K\}. \tag{7}$$

The plausibility function $pl(\cdot)$ represents evidence that does not contradict the z_k class [18].

B Evidential Weight Parameter Choice

Under the assumptions outlined in Section 2 and using the plausibility transformation, the equivalence of Dempster's rule and the softmax function is shown as follows,

$$p(z_k) = \frac{\exp\left(\sum_{j=1}^{J} \hat{\beta}_{jk} \phi_j(y) + \hat{\beta}_{0k}\right)}{\sum_{\ell=1}^{K} \exp\left(\sum_{j=1}^{J} \hat{\beta}_{j\ell} \phi_j(y) + \hat{\beta}_{0l}\right)}, \qquad p(z_k) = \frac{\exp\left(\sum_{j=1}^{J} \beta_{jk} \phi_j(y) + \alpha_{jk}\right)}{\sum_{\ell=1}^{K} \exp\left(\sum_{j=1}^{J} \beta_{j\ell} \phi_j(y) + \alpha_{j\ell}\right)}, \tag{8}$$

where on the left is the softmax function, on the right is the plausibility transformation of the output mass function in Eq. (3), and $\hat{\beta}$ are the parameters learned by the neural network [11]. By inspection, the two expressions are equal under the constraint $\hat{\beta}_{0k} = \sum_{j=1}^{J} \alpha_{jk}$. The mass function parameters α_{jk} and β_{jk} are not uniquely defined; Denoeux selects the parameters using the Least Commitment Principle, which is analogous to maximum entropy in probability theory [12]. The β_{jk} parameter may differ from $\hat{\beta}_{jk}$ by a constant c_j while still maintaining the equality between the DST result and the softmax transformation as in Eq. (8). The least commitment mass occurs when the evidential weight w_{jk} approaches zero corresponding to a vacuous mass by Eq. (6), resulting in the optimal parameters:

$$\beta^*_{jk} = \hat{\beta}_{jk} - \frac{1}{K} \sum_{l=1}^{K} \hat{\beta}_{j\ell} \quad \text{and} \quad \alpha^*_{jk} = \frac{1}{J} \left(\beta^*_{0k} + \sum_{j=1}^{J} \beta^*_{jk} \mu_j \right) - \beta^*_{jk} \mu_j \tag{9}$$

where $\mu_j = \frac{1}{N} \sum_{i=1}^{N} \phi(y)$ [11]. Instead of considering the full training set, we propose treating each input y individually at test time, forming new α^*_{jk} parameters:

$$\alpha^*_{jk} = \frac{1}{J} \left(\beta^*_{0k} + \sum_{j=1}^{J} \beta^*_{jk} \phi_j(y) \right) - \beta^*_{jk} \phi_j(y) \tag{10}$$

We show that the new parameters maintain the constraints required by Denoeux for DST-softmax equivalence in Appendix C. We note that the α_{jk} bias term is now a function of the test sample y. By substituting these parameters into Eq. (1), we obtain:

$$w_{jk} = \frac{1}{J} \left(\beta^*_{0k} + \sum_{j=1}^{J} \beta^*_{jk} \phi_j(y_i) \right) \tag{11}$$

Thus, the evidential weight w_{jk} is a function of the linear combination of the J features. We have that $w_k^+ - w_k^- = w_k$, which does not hold in the original formulation by Denoeux [11]. This fact ensures that at least one of w_k^+ or w_k^- is zero. By Eq. (3), the singleton mass $m(\{z_k\})$ is zero when $w_k^- > 0$ and at least one other class has $w_\ell^- = 0$ where $\ell \neq k$. Intuitively, this result indicates that there is no direct evidence for a class when there is aggregate evidence against this class, and there exists at least one other class that has aggregate evidence supporting it. We posit that filtering out the classes with zero singleton mass values does not result in loss of information, and imposes a more concentrated distribution output.

C DST-Softmax Equivalence Satisfaction

The constraint required to ensure that the DST combination is equivalent to the softmax transformation is: $\sum_{j=1}^{J} \alpha_{jk} = \hat{\beta}_{0k} + c_0$ for some constant c_0. Computing, we have:

$$\sum_{j=1}^{J} \alpha_{jk} = \sum_{j=1}^{J} \left[\frac{1}{J} \left(\beta_{0k}^* + \sum_{j=1}^{J} \beta_{jk}^* \phi_j(x_i) \right) - \beta_{jk}^* \phi_j(x_i) \right] \tag{12}$$

$$= \sum_{j=1}^{J} \left[\frac{1}{J} \left(\beta_{0k}^* + \sum_{j=1}^{J} \beta_{jk}^* \phi_j(x_i) \right) \right] - \sum_{j=1}^{J} \beta_{jk}^* \phi_j(x_i) \tag{13}$$

$$= \frac{1}{J} \left(\beta_{0k}^* + \sum_{j=1}^{J} \beta_{jk}^* \phi_j(x_i) \right) \sum_{j=1}^{J} 1 - \sum_{j=1}^{J} \beta_{jk}^* \phi_j(x_i) \tag{14}$$

$$= \frac{1}{J} \left(\beta_{0k}^* + \sum_{j=1}^{J} \beta_{jk}^* \phi_j(x_i) \right) J - \sum_{j=1}^{J} \beta_{jk}^* \phi_j(x_i) \tag{15}$$

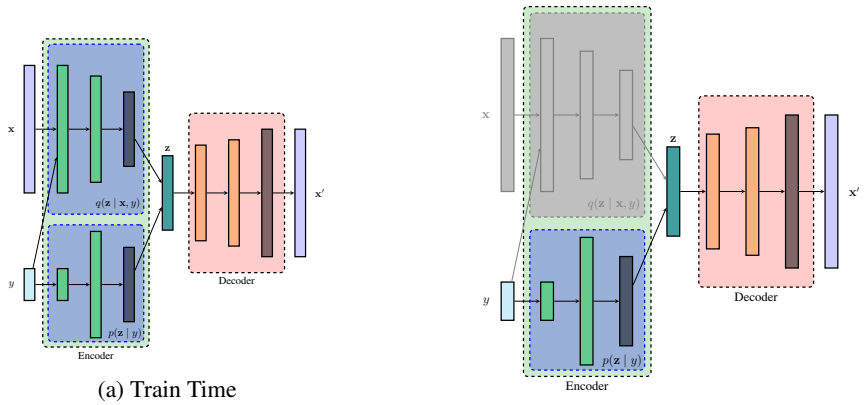
$$= \left(\beta_{0k}^* + \sum_{j=1}^{J} \beta_{jk}^* \phi_j(x_i) \right) - \sum_{j=1}^{J} \beta_{jk}^* \phi_j(x_i) \tag{16}$$

$$= \beta_{0k}^* \tag{17}$$

$$= \hat{\beta}_{0k} + \frac{1}{K} \hat{\beta}_{0k}. \tag{18}$$

The last line follows from the result in Eq. (9). Therefore, we have shown that the new α_{jk}^* parameters meet the required constraint for DST-softmax equivalence as posed by Denoeux [11].

D CVAE Architecture



(a) Train Time

(b) Test Time

Figure 3: The CVAE architecture used to demonstrate our approach.

During training, the encoder consists of two multi-layer perceptrons (MLP). One MLP takes as input the target class label y, and outputs a softmax probability distribution that parameterizes the prior distribution $p(z \mid y)$, where z is a discrete latent variable that can take on $K = 10$ values. The second encoder MLP takes as input a feature vector x and the target label y, and outputs the softmax distribution for $q(z \mid x, y)$. The z value is sampled from the q distribution and passed through another MLP to predict the decoded feature x'. This architecture is illustrated in Fig. 3a. The Gumbel-Softmax distribution was used to backpropagate loss gradients through the discrete latent space [8, 13]. The standard ELBO loss was maximized to train the model [6].

At test time, only the target class y serves as input to the prior encoder MLP. The latent space is sampled from $p(z \mid y)$, and passed through the decoder to generate a feature x'. This process is shown in Fig. 3b.

We use a hidden unit dimensionality of 30 for $p(z_k \mid y)$ and 256 for remaining two MLPs. We chose the 256 dimension following the example from: `https://github.com/timbmg/VAE-CVAE-MNIST`. within the architecture in Fig. 3. We use the ReLU nonlinearity with the original stochastic gradient descent and Adam [19] optimizers with a learning rate of 0.001 for the MNIST and Fashion MNIST datasets respectively.

E CVAE Test-Time Performance Comparison on MNIST

Fig. 4 shows a comparison of the generated images sampled from the softmax discrete latent distribution versus our proposed filtered distribution for the 'even' input. For instance, the softmax distribution often samples 'odd' labels given the even input class. Our distribution provides test time improvement on the generated sample correctness.

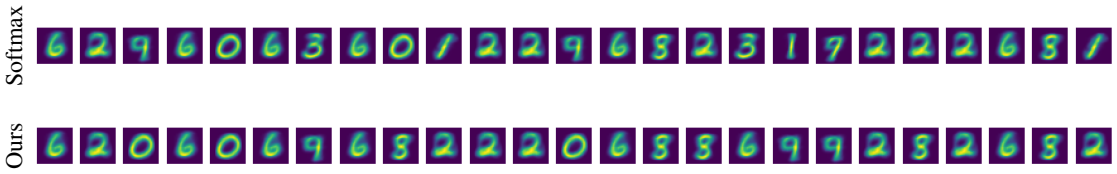


Figure 4: Samples from the network at test time using the softmax distribution and our proposed distribution for the 'even' input label.

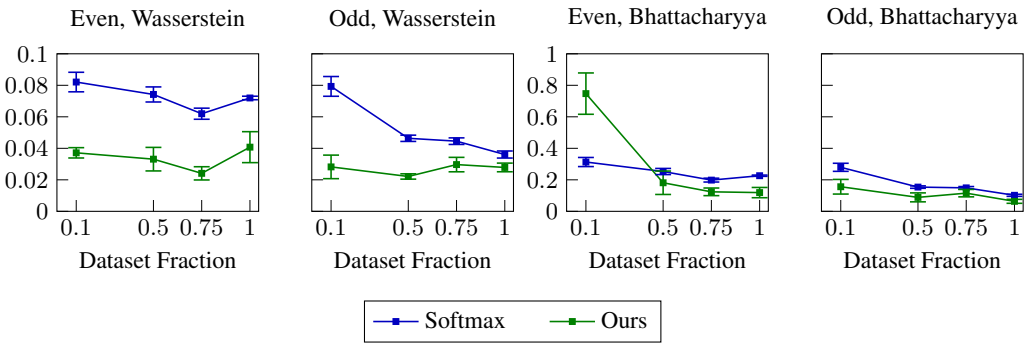


Figure 5: Filtered distribution performance across fewer training samples on the MNIST dataset.

F Quantitative Results on MNIST

F.1 Evaluation Metrics

Evaluating the quantitative performance of the filtered distribution is nontrivial as the ground truth desired distribution is ambiguous. In an attempt to standardize the performance evaluation, we introduce the following metric. The target distribution is computed by choosing the input binary class with the highest conditional softmax probability for each latent class $p(z_k \mid y)$, and then normalizing it to obtain a uniform distribution. Thus, we obtain a distribution over the more prominent classes as learned by the network. We write the target distribution definition as:

$$p_T(z_k \mid y) = \frac{\mathbb{1}\left\{p(z_k \mid y) \geq p(z_k \mid \bar{y})\right\}}{\sum_{k=1}^{K} \mathbb{1}\left\{p(z_k \mid y) \geq p(z_k \mid \bar{y})\right\}} \tag{19}$$

where $\bar{y}$ is the opposing binary class to y. We use the Wasserstein and Bhattacharyya distances to the target distribution as the metrics of performance. The Kullback-Leibler divergence was not used as it is undefined for probability mass values of zero. We baseline our filtered latent space distribution against the original softmax distribution learned by the neural network.

F.2 Reduced Data Performance

Fig. 5 summarizes the performance of the filtered distribution for a network trained on a reduced MNIST dataset. We use 0.1, 0.5, 0.75, and 1.0 fractions of the dataset for training, maintaining the class balance unchanged. We evaluate the distributions at the end of the 20^{th} training epoch. Fig. 5 demonstrates that the filtered distribution largely outperforms the softmax distribution outside of standard error across both metrics on the MNIST dataset. We note that the effectiveness of the learned latent representation decreases only marginally with fewer training samples. Due to the simplicity of our architecture, the latent space only needs to encode 10 digits, which can be learned from fewer samples, as long as the relative frequency of the observed digits remains balanced. We perform a similar analysis on the Fashion MNIST dataset in Appendix F.

G Results on Fashion MNIST

G.1 Qualitative Results

We investigate the qualitative performance of our proposed methodology on the Fashion MNIST dataset, where instead of 'odd' and 'even' inputs, we divide the dataset into 'tops' and 'bottoms/accessories'. The network learns a cleaner softmax distribution than that for the MNIST dataset as shown in Fig. 6. Our filtered distribution still provides some improvement by filtering out incorrect probability masses completely. We note that the filtered distribution makes two mistakes in keeping the first latent class (a boot) for the 'tops' label and keeping the sixth latent class (a shirt) for the 'bottoms' label. Nevertheless, we emphasize that these are false positives. Thus, our method performs at least as well as the original softmax distribution. In contrast, the sparsemax distribution results in several false negatives, such as the filtered out dress for the 'tops' and the filtered out boot, bag, and purse for the 'bottoms and accessories' class.

G.2 Quantitative Results

As in Appendix G.1, the latent classes for the Fashion MNIST data are learned more effectively than those for MNIST. We show in Fig. 7 that when the underlying network learns the latent space successfully, our filtered

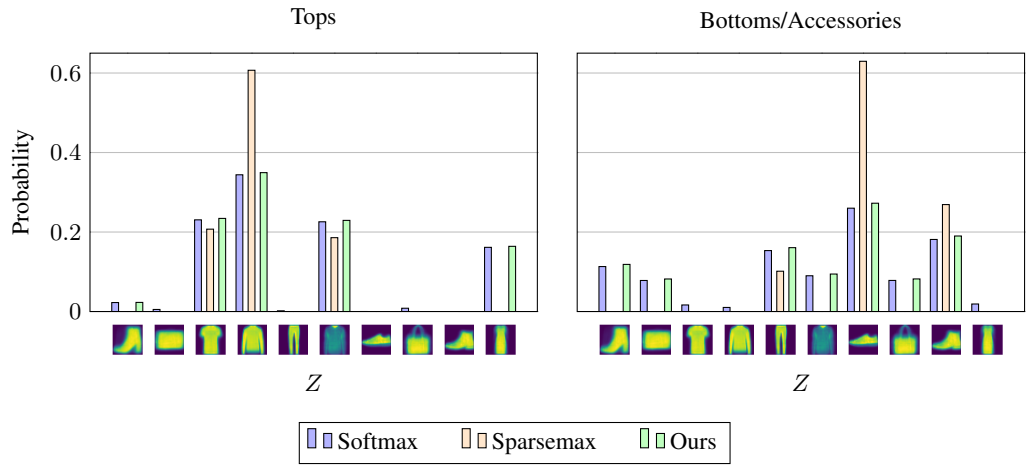


Figure 6: Visualization of the results on the Fashion MNIST test set for one random seed. The horizontal axis depicts decoded latent classes. Our proposed filtered distribution (green) is compared to the softmax (blue) and sparsemax (green) distributions. Our filtered distribution reduces the dimensionality of the relevant latent space without removing valid latent classes.

distribution performs no worse (and even slightly better) than the original softmax distribution. Thus, across both benchmarks, the proposed latent class distribution provides a more robust representation, retrieving richer information from the learned neural network weights with fewer training iterations.

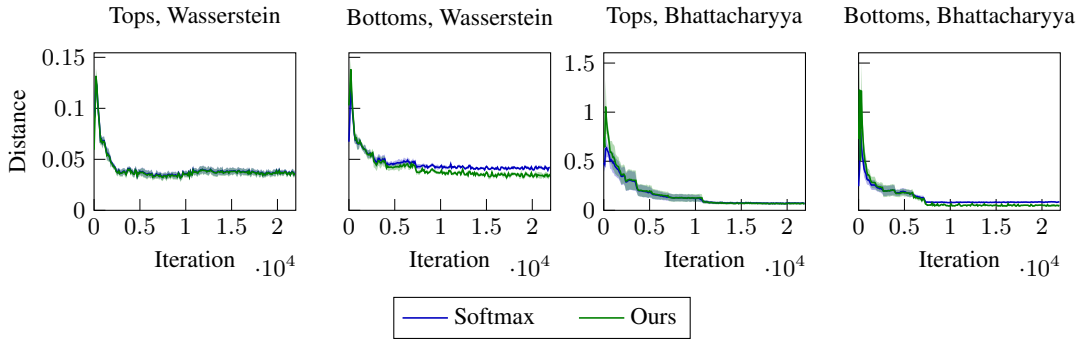


Figure 7: Results for the softmax and filtered distributions across training iterations on the Fashion MNIST datasets.

Fig. 8 summarizes the performance of the filtered distribution on a network trained on a reduced Fashion MNIST dataset. Due to the more effectively learned encoder weights, the filtered distribution performs within standard error and, for the bottoms class, even slightly better than the softmax distribution.

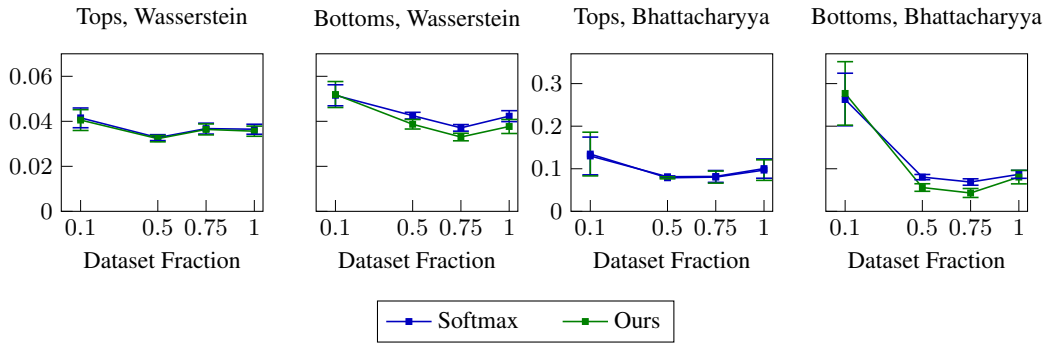


Figure 8: Filtered distribution performance across fewer training samples on Fashion MNIST.

H Related Work

Interpretability Many approaches for solving interpretability within neural networks have been proposed in recent years. The main focus of these approaches has been on learning low-dimensional representations that are understandable for human users [20, 21]. The usefulness of discrete latent spaces for interpretability in variational autoencoder models has been demonstrated for downstream tasks such as behavior prediction [3, 4, 5], multimodal transition modeling for reinforcement learning [7], and video frame prediction [22]. Several works disentangle latent representations, inducing interpretability by forcing structure into the latent space of the network [22, 23]. Often the problem of interpretability has been posed as the design of a regularization strategy for the neural network [21, 24, 25]. Melis and Jaakkola [21] use regularization to learn a locally linear, low-dimensional model for interpretability. Tsang et al. [24] employ regularization to disentangle or limit internal neural network interactions. Chen et al. [25] use the maximization of mutual information between a subset of latent variables and the observation to regularize the loss.

Our method provides a unique perspective from these works in that we generate a reduced dimensionality latent space at test time with the aim of improving interpretability for high-dimensional latent spaces. As an additional benefit, we achieve improved test time performance with a smaller latent space without having to retrain the neural network. The closest approach to our work is that of Olden and Jackson [26], where a post hoc analysis is performed to identify the features that do not contribute to the classification task. This approach is not applicable to our task as we are interested in identifying the classes (latent outputs) that best represent a particular input, thereby filtering the output classes rather than the input.

Softmax Function Alternatives In recent years, a number of new perspectives on the softmax function have been presented. The Gumbel-Softmax distribution was introduced to allow backpropagation through categorical distributions, giving rise to the popularity of discrete latent spaces within CVAE architectures [8, 13]. Sensoy et al. [27] present an evidential approach to deep learning for classification tasks. They propose learning Dirichlet distribution parameters, obtaining a distribution over softmax functions. The Dirichlet parameters then serve as evidence towards singleton classes, resulting in a loss function that regularizes misleading evidence towards the vacuous mass function. Unlike [27], where the method is incorporated into the loss function, we propose test time disambiguation of the softmax distribution. Duch and Itert [28] suggest a post hoc modification to further disperse the uncertainty among the classes, thus *flattening* the softmax distribution to improve classification performance. We are interested in the opposite objective of removing the latent classes that have probability mass assigned to them due to pure uncertainty allocation, in this way generating a more *sharply peaked* distribution.

Potentially the closest in motivation to our work is the sparsemax [14] distribution, which aims to (1) improve interpretability by reducing latent space dimensionality and (2) address multimodality in classification tasks. An important distinction between our work and sparsemax is the formulation of the problem to arrive at the sparse distribution. Sparsemax finds the Euclidean projection of the input onto the probability simplex. In contrast, we identify a sparse distribution by means of evidential theory, filtering the classes that do not have direct evidence towards them as determined by the learned weights of the neural network and the conditioned input. Empirically, we show that sparsemax results in undesirably more aggressive filtering than our method, eliminating viable latent classes unnecessarily.