Evidential Softmax for Sparse Multimodal Distributions in Deep Generative Models

PHIL CHEN, MASHA ITKINA, RANSALU SENANAYAKE, AND MYKEL J. KOCHENDERFER

Stanford University

NeurIPS 2021

SISL
Stanford Intelligent Systems Laboratory

Deep Generative Models with Discrete Probability Spaces

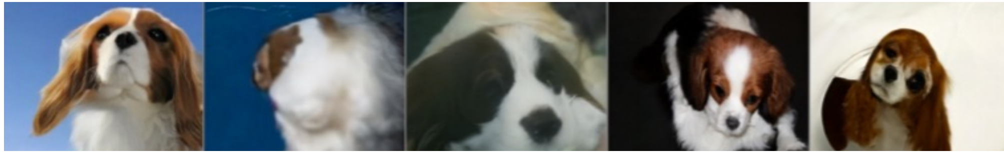
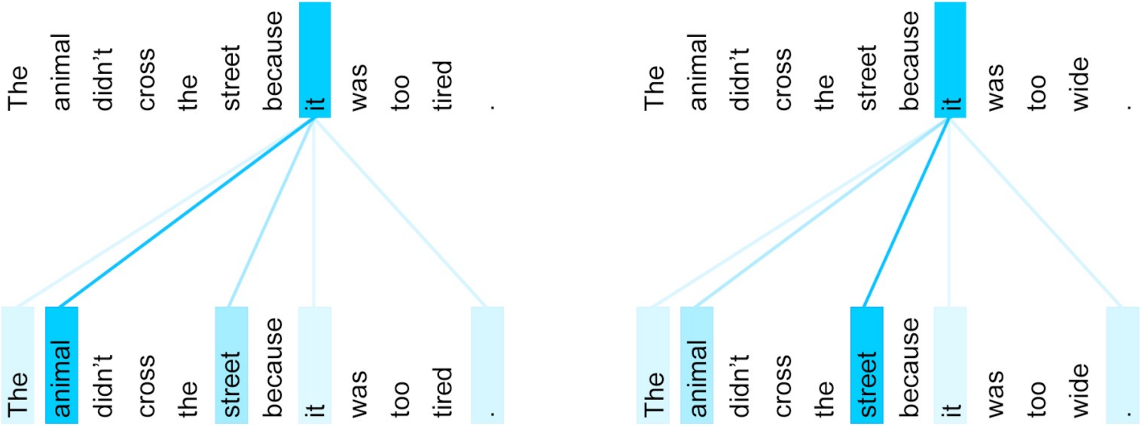


Image generation (VQ-VAE) [van den Oord et al. , NeurIPS, 2017]



Language Modeling [Vaswani et al., NeurIPS, 2017]

Deep Generative Models with Discrete Probability Spaces

Problem: Discrete probability spaces need to be **sufficiently large** to capture the complexities of real-world data, rendering downstream tasks **computationally challenging** [Kaiser et al., ICML, 2018].

Deep Generative Models with Discrete Probability Spaces

Problem: Discrete probability spaces need to be **sufficiently large** to capture the complexities of real-world data, rendering downstream tasks **computationally challenging** [Kaiser et al., ICML, 2018].

VQ-VAE for Image Generation [van den Oord et al., NeurIPS, 2017]

Deep Generative Models with Discrete Probability Spaces

Problem: Discrete probability spaces need to be **sufficiently large** to capture the complexities of real-world data, rendering downstream tasks **computationally challenging** [Kaiser et al., ICML, 2018].

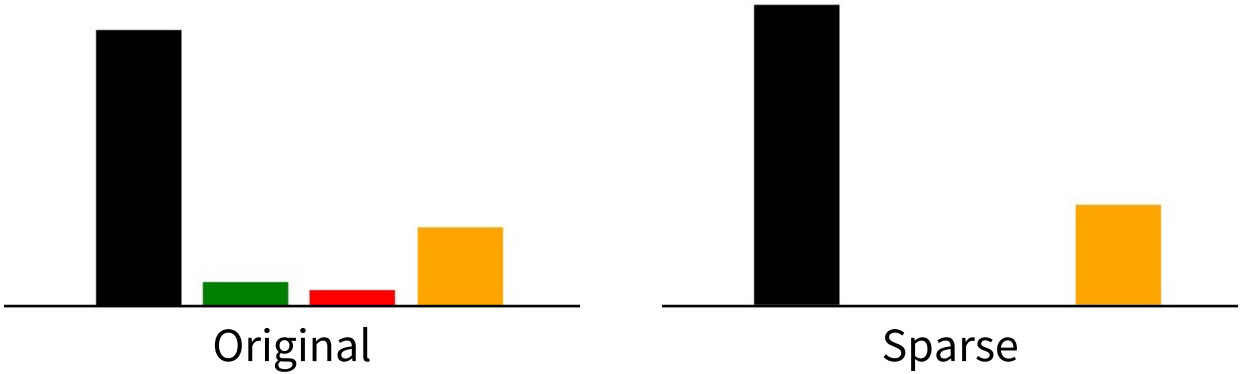
Solutions:

- Sample stochastically (e.g., score function estimator [Glynn et al., ACM, 1990])
- Generate sparse distribution (e.g., sparsemax [Martins et al., ICML, 2016])

VQ-VAE for Image Generation [van den Oord et al., NeurIPS, 2017]

Challenges with Sparse Normalization Functions

- Sparsifying distributions may collapse valid modes [Itkina et al., NeurIPS, 2020].
- Traditional loss functions, such as NLL and KL divergence, are undefined for zero probabilities.



Can we train ***discrete generative models*** *to predict* ***sparse and multimodal*** *probability distributions?*

Evidential Sparsification

Itkina et al. [NeurIPS, 2020] introduce a normalization function which is a post-hoc procedure to sparsify the latent space of conditional variational autoencoders (CVAEs) at test time ***without sacrificing multimodality***.

Evidential Sparsification

Itkina et al. [NeurIPS, 2020] introduce a normalization function which is a post-hoc procedure to sparsify the latent space of conditional variational autoencoders (CVAEs) at test time ***without sacrificing multimodality***.

Interpretation of Evidential Theory [Dempster, 2008]

Evidential Conflict ⟷ Multimodality

Evidential Sparsification

Itkina et al. [NeurIPS, 2020] introduce a normalization function which is a post-hoc procedure to sparsify the latent space of conditional variational autoencoders (CVAEs) at test time ***without sacrificing multimodality***.

Interpretation of Evidential Theory [Dempster, 2008]

Evidential Conflict ⟷ Multimodality

Lack of Evidence ⟷ Sparsity

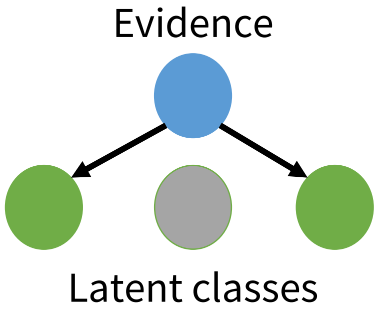
Evidential Sparsification

Itkina et al. [NeurIPS, 2020] introduce a normalization function which is a post-hoc procedure to sparsify the latent space of conditional variational autoencoders (CVAEs) at test time ***without sacrificing multimodality***.

Interpretation of Evidential Theory [Dempster, 2008]

Evidential Conflict ↔ Multimodality

Lack of Evidence ↔ Sparsity



Contributions

1. We introduce **ev-softmax**, a strategy for training neural networks with sparse probability distributions that is compatible with NLL and KL divergence.

2. We derive **properties** of ev-softmax and its training-time approximation.

3. Our approach outperforms baselines in **distributional accuracy** across tasks in **image generation and machine translation**.

Ev-Softmax

We find a simple, equivalent closed form of the post hoc sparsification method [Itkina et al., NeurIPS, 2020].

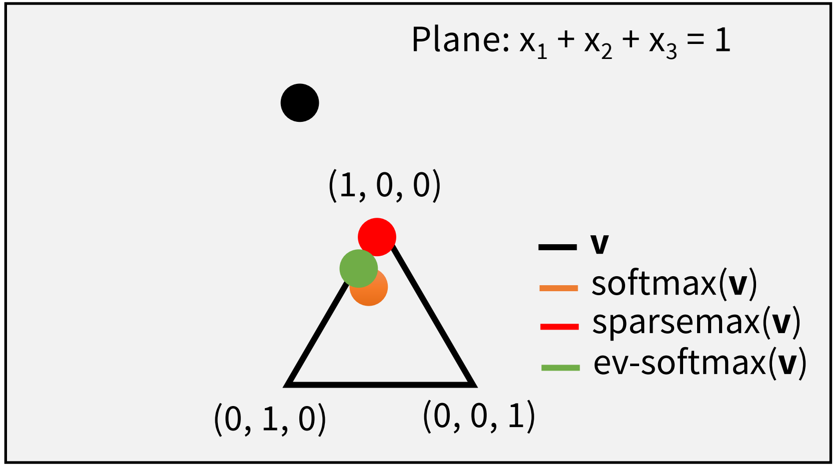
$$\textsc{Softmax}(\mathbf{v})_k \propto e^{v_k}$$

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

where $\mathbf{v} = \hat{\beta}^T \phi \in \mathbb{R}^K$ and $\overline{v} = \frac{1}{K}\sum_{i=1}^{K} v_i$.

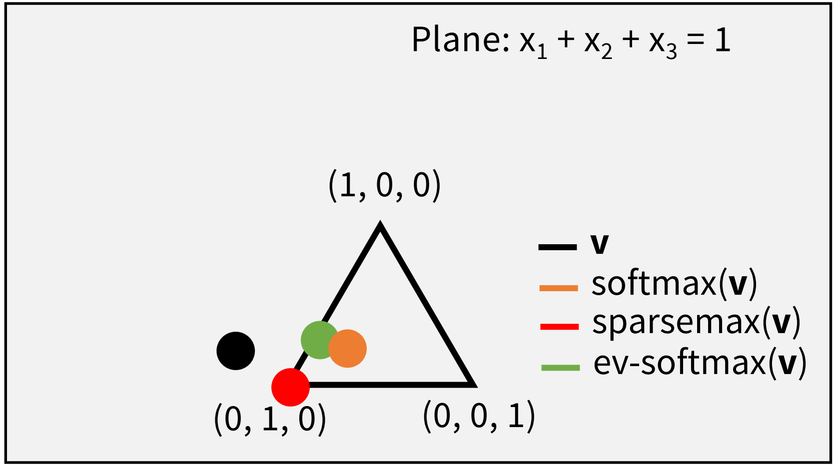
Ev-Softmax Examples

$\mathbf{v}$	Softmax($\mathbf{v}$)	Sparsemax($\mathbf{v}$)	**Ev-Softmax($\mathbf{v}$)**
$(1.3, 0.37, -0.67)$	$(0.65, 0.26, 0.09)$	$(0.97, 0.03, 0)$	$(0.72, 0.28, 0)$
$(0.4, 1.4, -0.8)$	$(0.25, 0.67, 0.07)$	$(0, 1, 0)$	$(0.27, 0.73, 0)$



Ev-Softmax Examples

$\mathbf{v}$	Softmax($\mathbf{v}$)	Sparsemax($\mathbf{v}$)	**Ev-Softmax($\mathbf{v}$)**
$(1.3, 0.37, -0.67)$	$(0.65, 0.26, 0.09)$	$(0.97, 0.03, 0)$	$(0.72, 0.28, 0)$
$(0.4, 1.4, -0.8)$	$(0.25, 0.67, 0.07)$	$(0, 1, 0)$	$(0.27, 0.73, 0)$



Ev-Softmax Jacobian

The Jacobians of softmax and ev-softmax are similar.

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

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

Ev-Softmax Properties

Ev-softmax exhibits similar properties to softmax.

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^*

Ev-Softmax Training

During training, we modify ev-softmax as follows.

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

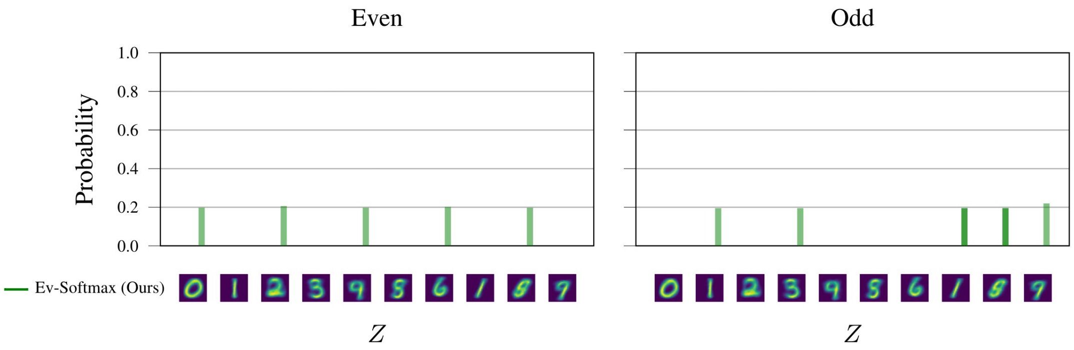
In the limit, the NLL loss term takes a similar form to softmax.

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

Experiments: Toy MNIST Example

Task: generate images of even and odd numbers.

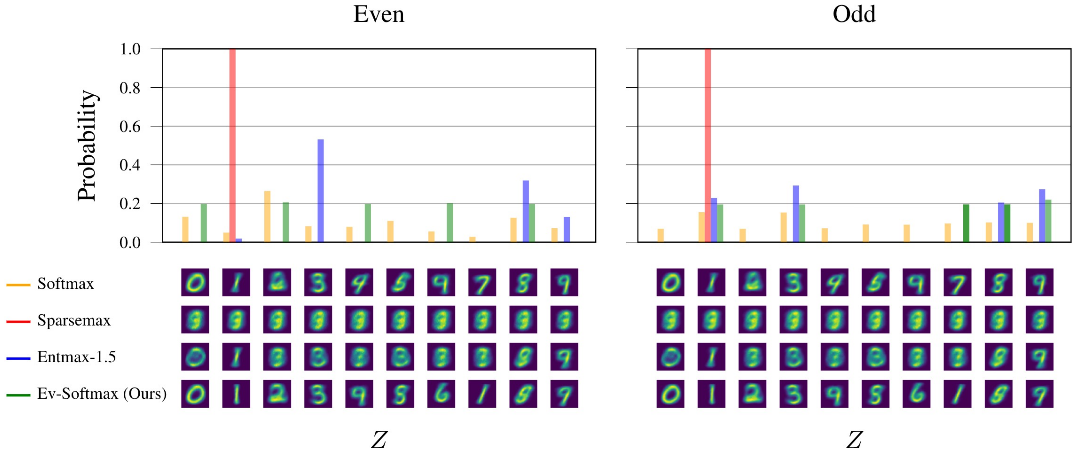
Results: Plot shows our ev-softmax (green) distribution.



Experiments: Toy MNIST Example

Task: generate images of even and odd numbers.

Results: Plot shows softmax (yellow), sparsemax (orange) [Martins et al., ICML, 2016], entmax-1.5 (blue) [Peters et al., ACL, 2019], and our ev-softmax (green) distributions.

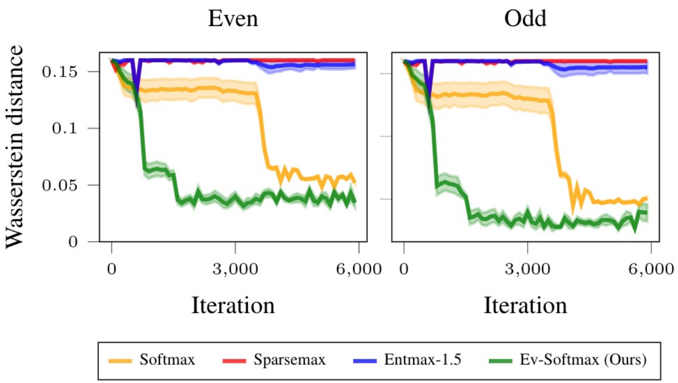


Note: Sparsemax and entmax-1.5 appear to collapse the latent space.

Experiments: Toy MNIST Example

Task: generate images of even and odd numbers.

Results: Plot shows the Wasserstein distance for the softmax (yellow), sparsemax (orange) [Martins et al., ICML, 2016], entmax-1.5 (blue) [Peters et al., ACL, 2019], and ev-softmax (green) distributions.



Note: Our method yields the lowest Wasserstein distance.

Experiments: Image Generation

Task: image generation on *tiny*ImageNet
Network Architecture: VQ-VAE [van den Oord et al., NeurIPS, 2017], 16x16 512-class latent space
Results: 85% reduction in the latent sample space

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

Note: Our method sparsifies the distribution at a level between sparsemax and entmax-1.5, but outperforms in accuracy.

Experiments: Machine Translation

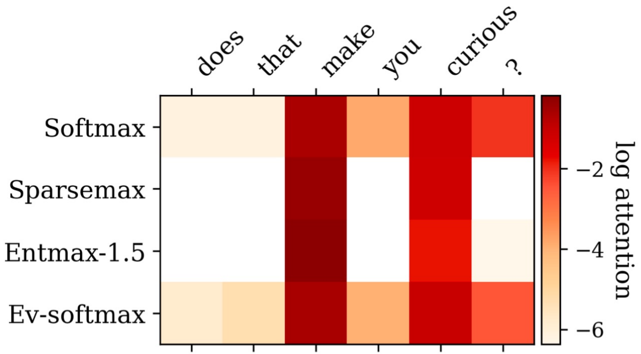
Task: machine translation on IWSLT 2014 [Cettolo et al., IWSLT, 2017]
Network Architecture: OpenNMT transformer [Klein et al., AMTA, 2020]

Metric	Softmax	Post-hoc Evidential	Sparsemax	Entmax-1.5	Ev-softmax
BLEU	29.2 ± 0.06	29.2 ± 0.05	29.0 ± 0.05	29.2 ± 0.07	$\mathbf{29.4 \pm 0.05}$
ROUGE-1	59.31	59.09	58.47	58.94	**59.32**
ROUGE-2	35.62	35.42	34.76	35.20	**35.74**
ROUGE-L	56.09	55.93	55.39	55.75	**56.18**
METEOR	57.02	56.84	56.33	5.83	**57.20**
p-val (METEOR)	< 0.05	< 0.01	< 0.001	< 0.01	N/A
# attended	39.5 ± 11.5	3.8 ± 0.93	2.3 ± 0.54	4.1 ± 1.3	8.2 ± 1.3

Note: Our method attends to a larger number of words than sparsemax and entmax-1.5, and outperforms all methods in all computed metrics.

Experiments: Machine Translation

Task: machine translation on IWSLT 2014 [Cettolo et al., IWSLT, 2017]
Network Architecture: OpenNMT transformer [Klein et al., AMTA, 2020]



Note: Sparsemax and entmax-1.5 both attend over two words while ev-softmax and softmax attend over the entire source.

Conclusions

- We present a sparse normalization function grounded in evidential theory for use in generative models with categorical output distributions.

Conclusions

- We present a sparse normalization function grounded in evidential theory for use in generative models with categorical output distributions.
- Ev-softmax satisfies many of the same properties as softmax and other alternatives to softmax.

Conclusions

- We present a sparse normalization function grounded in evidential theory for use in generative models with categorical output distributions.
- Ev-softmax satisfies many of the same properties as softmax and other alternatives to softmax.
- Ev-softmax outperforms softmax and sparse normalization functions across tasks in image generation and machine translation.

Conclusions

- We present a sparse normalization function grounded in evidential theory for use in generative models with categorical output distributions.
- Ev-softmax satisfies many of the same properties as softmax and other alternatives to softmax.
- Ev-softmax outperforms softmax and sparse normalization functions across tasks in image generation and machine translation.
- Future work can involve training computer vision and machine translation models from scratch on a larger scale.

Conclusions

- We present a sparse normalization function grounded in evidential theory for use in generative models with categorical output distributions.
- Ev-softmax satisfies many of the same properties as softmax and other alternatives to softmax.
- Ev-softmax outperforms softmax and sparse normalization functions across tasks in image generation and machine translation.
- Future work can involve training computer vision and machine translation models from scratch on a larger scale.
- Our code is available at https://github.com/sisl/EvSoftmax

References

[1] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In Advances in Neural Information Processing Systems (NeurIPS), pages 6306–6315, 2017.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Adrian N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems (NeurIPS), pages 5998 – 6008, 2017.

[3] Lukasz Kaiser, Samy Bengio, Aurko Roy, Ashish Vaswani, Niki Parmar, Jakob Uszkoreit, and Noam Shazeer. Fast decoding in sequence models using discrete latent variables. In International Conference on Machine Learning (ICML), pages 2390–2399, 2018.

[4] Arash Vahdat, William Macready, Zhengbing Bian, Amir Khoshaman, and Evgeny Andriyash. DVAE++: Discrete variational autoencoders with overlapping transformations. In International Conference on Machine Learning (ICML), pages 5035–5044, 2018.

[5] Brian Ichter, James Harrison, and Marco Pavone. Learning sampling distributions for robot motion planning. In International Conference on Robotics and Automation (ICRA), pages 7087–7094. IEEE, 2018.

[6] Masha Itkina, Boris Ivanovic, Ransalu Senanayake, Mykel J. Kochenderfer, and Marco Pavone. Evidential Sparsification of multimodal latent spaces in conditional variational autoencoders. In Advances in Neural Information Processing Systems (NeurIPS), volume 33, 2020.

References

[7] Arthur P. Dempster. A generalization of Bayesian inference. Classic works of the Dempster-Shafer Theory of Belief Functions, pages 73–104, 2008.

[8] Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In International Conference on Machine Learning (ICML), pages 1614–1623, 2016.

[9] Ben Peters, Vlad Niculae, and Andre FT Martins. Sparse Sequence-Sequence Models. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1504– 1519, 2019.

[10] Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. Report on the 11th IWSLT evaluation campaign, IWSLT 2014. In Proceedings of the International Workshop on Spoken Language Translation, Hanoi, Vietnam, volume 57, 2014.

[11] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In Proceedings of ACL 2017, System Demonstrations, pages 67-72, 2017.

Thank you!

SISL
Stanford Intelligent
Systems Laboratory