On the Benefits of Defining Vicinal Distributions in Latent Space

Puneet Mangla [1] Vedant Singh [1] Shreyas Havaldar [1] Vineeth N Balasubramanian [1]

[1]Department of Computer Science, Indian Institute of Technology, Hyderabad, India

Motivation and Contribution

Motivation: VAEs can capture the latent space from which a distribution is generated providing us an unfolded manifold, with observable linearity in between training examples.

Key Contributions:

- Propose a new vicinal distribution called *VarMixup (Variational Mixup)* to sample better Mixup images.
- Experiments shows that VarMixup boosts the robustness to out-of-distribution shifts as well calibration.
- Additional analysis show that VarMixup significantly decreases the local linearity error of the neural network.

Background and Related Work

- **Empirical Risk Minimization (ERM)** minimize the average error over the training dataset

$$p_{actual}(x,y) \approx p_{\delta}(x,y) = \frac{1}{N}\cdot\sum_{i=1}^{N}\delta(x=x_i, y=y_i)$$

$$w^* = arg\min_w \int \mathcal{L}(F_w(x),y)\cdot dp_{\delta}(x,y) = arg\min_w \frac{1}{N}\cdot\sum_{i=1}^{N}\mathcal{L}(F_w(x_i),y_i)$$

Drawback: Overparametrized NNs suffer from memorization → leads to undesirable behavior outside the training distribution.

- **Vicinal Risk Minimization**: Popularly known as data augmentation. → define a vicinity or neighbourhood around each training example (eg. in terms of brightness, contrast, noise, etc.)

$$p_{actual}(x,y) \approx p_v(x,y) = \frac{1}{N}\cdot\sum_{i=1}^{N}v(x,y|x_i,y_i)$$

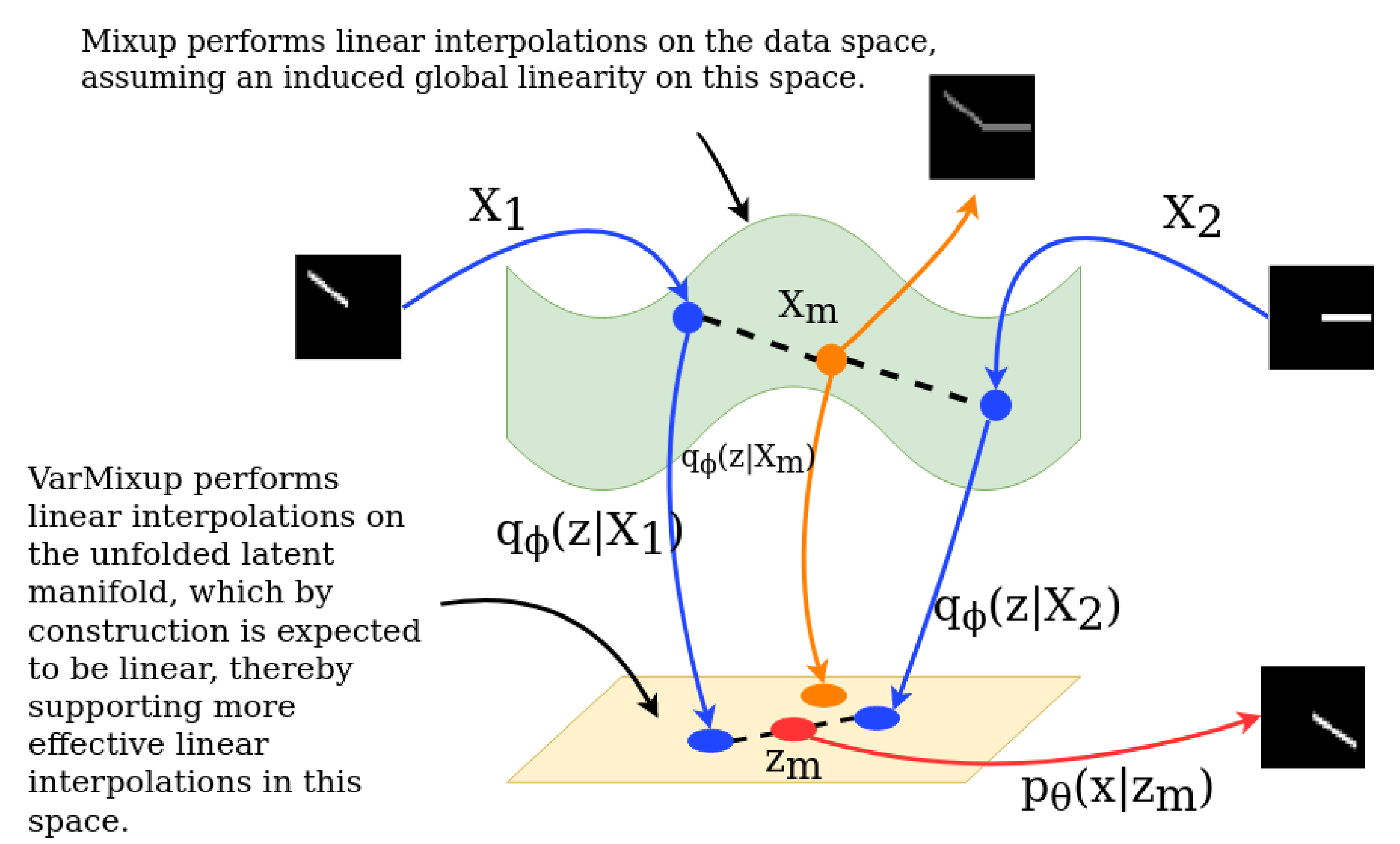
, where v is the *vicinal distribution* that calculates the probability of a data point (x,y) in the vicinity of other samples (x_i, y_i).
Expected Vicinal Risk, then is given by

$$w^* = arg\min_w \int \mathcal{L}(F_w(x),y)\cdot dp_v(x,y) = \frac{1}{N}\cdot\sum_{i=1}^{N}g(F_w,\mathcal{L},x_i,y_i)$$

where $g(F_w,\mathcal{L},x_i,y_i) = \int \mathcal{L}(F_w(x),y)\cdot dv(x,y|x_i,y_i)$.

- **MixUp** is a popular technique to train models for better generalisation
  - Pang et al., 2020, Hendrycks et al., 2020, Lamb et al., 2019 - Mixup to improve the robustness of models.
  - Thulasidasan et al., 2019 - Mixup-trained networks are significantly better calibrated.

Our Approach - VarMixup



- We opt for an MMD-VAE because of its advantage over vanilla KL based VAE

$$\mathcal{L}_{MMD-VAE} = \gamma\cdot MMD(q_\phi(z)\|p(z)) + \mathbb{E}_{x\sim p_{actual}}\mathbb{E}_{z\sim q_\phi(z|x)}[\log(p_\theta(x|z))]$$

- Equivalent to constructing VarMixup samples as:

$$x' = \mathbb{E}_x\left[p_\theta\left(x|\lambda\cdot\mathbb{E}_z[q_\phi(z|x_i)] + (1-\lambda)\cdot\mathbb{E}_z[q_\phi(z|x_j)]\right)\right]$$
$$y' = \lambda\cdot y_i + (1-\lambda)\cdot y_j$$

Experiments

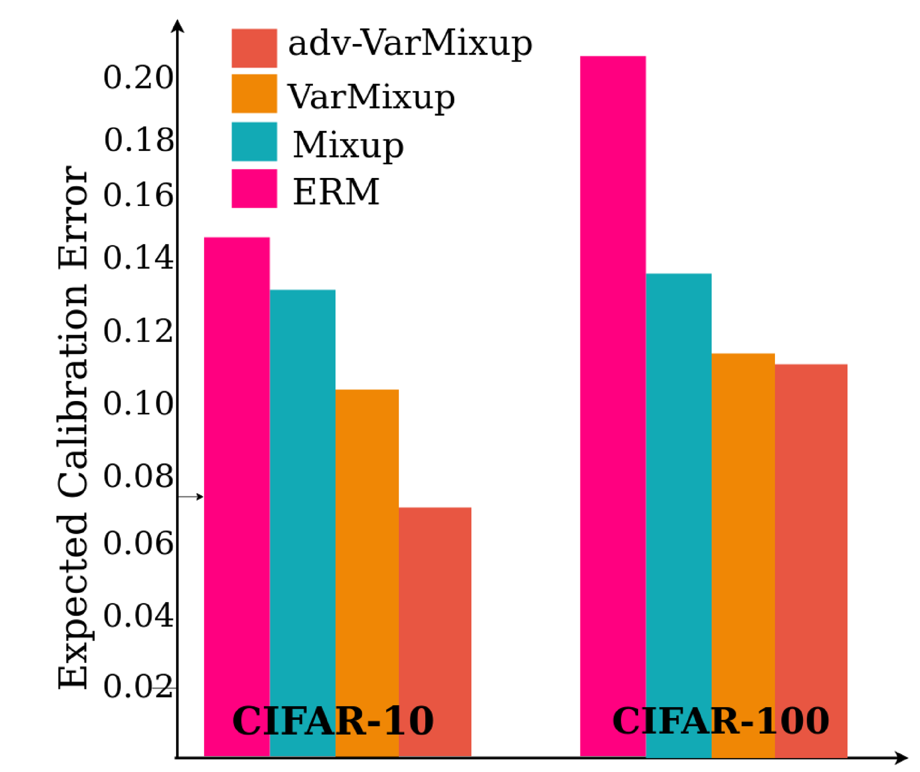
OOD Generalization

- Robustness to common input corruptions on CIFAR-10-C, CIFAR-100-C and Tiny-Imagenet-C
- *adv*-VarMixup : Variant of VarMixup where we use adversarial robust VAE.

Method	CIFAR-10-C	CIFAR-100-C
AT (Madry et al., 2018)	73.12 ± 0.31 (85.58 ± 0.14)	45.09 ± 0.31 (60.28 ± 0.13)
TRADES (Zhang et al., 2019)	75.46 ± 0.21 (88.11 ± 0.43)	45.98 ± 0.41 (63.3 ± 0.32)
IAT (Lamb et al., 2019)	81.05 ± 0.42 (89.7 ± 0.33)	50.71 ± 0.25 (62.7 ± 0.21)
ERM	69.29 ± 0.21 (94.5 ± 0.14)	47.3 ± 0.32 (64.5 ± 0.10)
Mixup	74.74 ± 0.34 (95.5 ± 0.35)	52.13 ± 0.43 (76.8 ± 0.41)
Mixup-R	74.27 ± 0.22 (89.88 ± 0.11)	43.54 ± 0.15 (62.24 ± 0.21)
Manifold-Mixup	72.54 ± 0.14 (95.2 ± 0.18)	41.42 ± 0.23 (75.3 ± 0.48)
VarMixup	**82.57 ± 0.42** (93.91 ± 0.45)	52.57 ± 0.39 (73.2 ± 0.44)
adv-VarMixup	82.12 ± 0.46 (92.19 ± 0.32)	**54.0 ± 0.41** (72.13 ± 0.34)

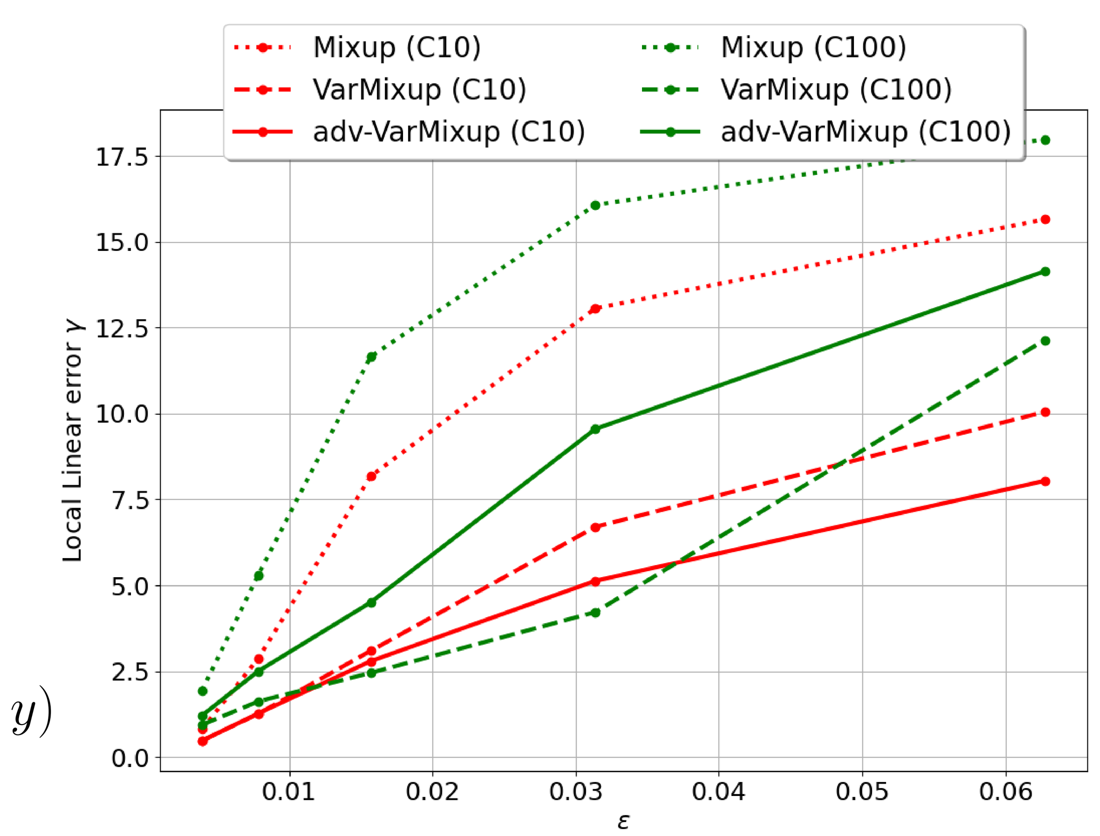
Calibration

Measures how good softmax scores are as indicators of the actual likelihood of a correct prediction. We measure the *Expected Calibration Error (ECE, lower the better)*



Local Linearity of Loss Surfaces

- Qin et al., 2020 - local linearity of loss landscapes of NNs positively correlates robustness.
- Local linearity at a data-point x within a neighbourhood $B(\epsilon)$ as $\gamma(\epsilon, x, y) = \max_{\delta\in B(\epsilon)} |\mathcal{L}(F_w(x+\delta),y) - \mathcal{L}(F_w(x),y) - \delta^T\nabla_x\mathcal{L}(F_w(x),y)|$



References

[1] D. P. Kingma and M. Welling.
Auto-encoding variational bayes.
arXiv, 2013.

[2] A. Lamb, V. Verma, J. Kannala, and Y. Bengio.
Interpolated adversarial training: Achieving robust neural networks without sacrificing too much accuracy.
In *AISec*, AISec'19, page 95–103, New York, NY, USA, 2019. Association for Computing Machinery.

[3] T. Pang*, K. Xu*, and J. Zhu.
Mixup inference: Better exploiting mixup to defend adversarial attacks.
In *ICLR*, 2020.

[4] C. Qin, J. Martens, S. Gowal, D. Krishnan, K. Dvijotham, A. Fawzi, S. De, R. Stanforth, and P. Kohli.
Adversarial robustness through local linearization.
NeurIPS, 2019.

[5] S. Thulasidasan, G. Chennupati, J. Bilmes, T. Bhattacharya, and S. Michalak.
On mixup training: Improved calibration and predictive uncertainty for deep neural networks.
NeurIPS, 2019.

[6] S. Zhao, J. Song, and S. Ermon.
Infovae: Information maximizing variational autoencoders.
arXiv, 2017.