Variational Classification

Shehzaad Dhuliawala **Mrinmaya Sachan**
Department of Computer Science, ETH Zurich
{shehzaad.dhuliawala, mrinmaya.sachan}@inf.ethz.ch

Carl Allen
AI Centre, ETH Zurich
carl.allen@ai.ethz.ch

Abstract

We present *variational classification* (VC), a latent variable generalisation of neural network softmax classification under cross-entropy loss. Our approach provides a novel probabilistic interpretation of the highly familiar softmax classification model, to which it relates comparably to variational vs deterministic autoencoders. We derive a training objective based on the evidence lower bound (ELBO) that is non-trivial to optimize, and an adversarial approach to maximise it. We reveal an inherent inconsistency within softmax classification that VC addresses, while also allowing flexible choices of distributions in the latent space in place of assumptions implicit in standard softmax classifiers. Empirical evaluation demonstrates that VC maintains accuracy while improving properties such as calibration and adversarial robustness, particularly under distribution shift and low data settings. This work brings new theoretical insight to modern machine learning practice.

1 Introduction

Classification is a core task in machine learning, from categorising objects (Klasson et al., 2019) and providing medical diagnoses (Adem et al., 2019; Mirbabaie et al., 2021), to identifying potentially life-supporting planets (Tiensuu et al., 2019). Classification tasks are commonly tackled by training domain-specific neural networks with a *sigmoid* or *softmax* output layer.[1] Data samples x (in a domain $\mathcal{X}$) are mapped deterministically by a network f_ω (with weights ω) to a real vector $z = f_\omega(x)$, which is transformed in the softmax layer to a point on the simplex $\Delta^{|\mathcal{Y}|}$, that parameterises $p_\theta(y|x)$, a discrete distribution over class labels $y \in \mathcal{Y}$:

$$p_\theta(y|x) = \frac{\exp\{z^\top w_y + b_y\}}{\sum_{y' \in \mathcal{Y}} \exp\{z^\top w_{y'} + b_{y'}\}} \ . \tag{1}$$

Although softmax classifiers often perform well, they suffer well-known issues: (i) they are are poorly understood theoretically and in many respects a "black box" with predictions $p_\theta(y|x)$ hard to explain; (ii) predictions can vary significantly for imperceptible changes in the data (adversarial examples); (iii) predictions may identify a true label as the most probable class but poorly reflect uncertainty in the prediction (miscalibration); and (iv) they typically require a lot of data to train.

We introduce *Variational Classification* (**VC**), which generalises softmax cross-entropy classification under a latent variable model (figure 2). The VC framework ascribes probabilistic roles to components of a softmax classifier: (i) the neural network (excluding the softmax layer) transforms a *mixture of unknown distributions* in the data space to a *mixture of chosen distributions* in the latent space; and (ii) the softmax layer converts the latter to class predictions by Bayes' rule. We show that, without tailoring the loss function to mitigate any particular issue with softmax classification, VC maintains predictive accuracy while the additional latent structure improves calibration, robustness to adversarial perturbations and domain shift, and performance in low data regimes.

[1] We refer throughout to the softmax function since it generalises sigmoid to multiple classes.

Mathematics of Modern Machine Learning Workshop at NeurIPS 2023.
Full paper available at: https://arxiv.org/pdf/2305.10406.pdf

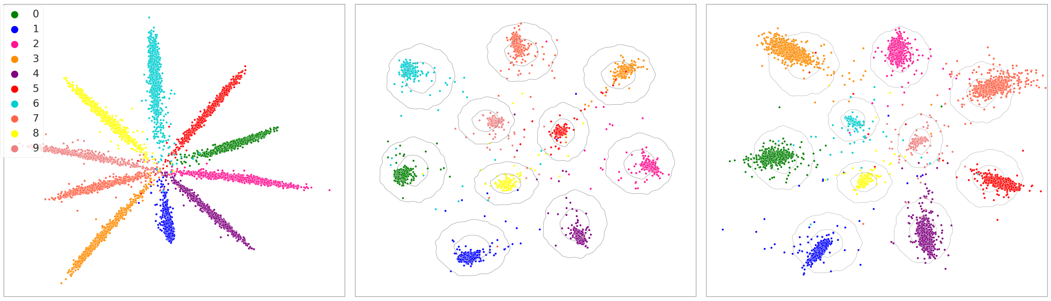

Figure 1: Empirical latent distributions: softmax inputs under (*l*) Softmax CE ["MLE"]; (*c*) MLE + Gaussian $p_\theta(\mathrm{z}|y)$ (contours); ["MAP"]; (*r*) VC objective ["Bayesian"]. Colour denotes MNIST class.

2 Background (Variational Auto-Encoder)

Estimatiing parameters of a latent variable model $p_\theta(x) = \int_z p_\theta(x|\mathrm{z})p_\theta(\mathrm{z})$ is typically intractable, and instead one maximises the *evidence lower bound* (ELBO):

$$\int_x p(x)\log p_\theta(x) \geq \int_x p(x)\int_z q_\phi(z|x)\Big\{\log p_\theta(x|z) - \log\tfrac{q_\phi(z|x)}{p_\theta(z)}\Big\}, \tag{2}$$

The *variational auto-encoder* (VAE, Kingma & Welling, 2014; Rezende et al., 2014) is an implementation of the ELBO in which all distributions are assumed Gaussian, with $p_\theta(x|z)$, $q_\phi(z|x)$ parameterised by neural networks. The VAE probabilistically generalises a deterministic auto-encoder, allowing for uncertainty or stochasticity in the latent $\mathrm{z}|x$, whose entropy is promoted and is constrained to a prior by the second ("regularisation") term.

3 Variational Classification

A Latent Variable Model for Classification: Data $x \in \mathcal{X}$ and labels $y \in \mathcal{Y}$ are treated as samples of random variables x, y jointly distributed by $p(\mathrm{x}, \mathrm{y})$. A softmax classifier is a deterministic function mapping x, via a sequence of intermediate representations, to a point on the simplex $\Delta^{|\mathcal{Y}|}$ that parameterises a categorical label distribution $p_\theta(\mathrm{y}|x)$.

Any intermediate representation $z = g(x)$ can be considered the realisation of a *latent* random variable sampled from a conditional (delta) distribution: $z \sim p(\mathrm{z}|x) = \delta_{z-g(x)}$. Under a (Markov) generative latent variable model (figure 2, *left*):

$$p(x) = \int_{y,z} p(x|z)p(z|y)p(y)\ , \tag{3}$$

class labels can be predicted:

$$p_\theta(y|x) = \int_z p_\theta(y|z)p_\theta(z|x)\ . \tag{4}$$

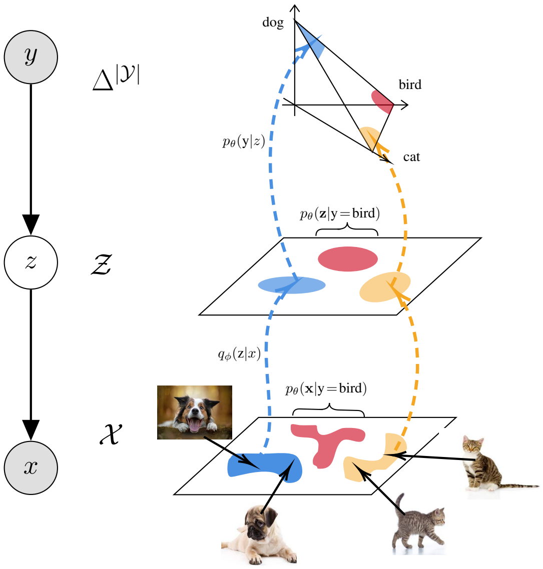

Figure 2: VC: $q_\phi(\mathrm{z}|x)$ stochastically maps data $x \in \mathcal{X}$ to the latent space $\mathcal{Z}$, where *empirical* $q_\phi(\mathrm{z}|y)$ are fitted to *anticipated* $p_\theta(\mathrm{z}|y)$; the output layer computes $p_\theta(y|z)$ by Bayes' rule for class prediction $p(y|x)$.

A softmax classifier is a special case of Eqn. 4 where: (i) f_ω, the neural network up to the softmax layer, parameterises $p_\theta(z|x) = \delta_{z-f_\omega(x)}$, a delta distribution; (ii) the softmax layer input is considered a sample from $p_\theta(z|x)$; and (iii) $p_\theta(\mathrm{y}|z)$ is defined by the softmax layer (see RHS of Equation 1).

Training a Classification LVM: Similarly to the latent model for $p_\theta(\mathrm{x})$ (§2), parameters of eqn 4 cannot generally be learned by directly maximising the likelihood, but rather a lower bound (*cf* eqn 2):

$$\begin{aligned}\int_{x,y} p(x,y) \log p_\theta(y|x) &= \int_{x,y} p(x,y) \int_z q_\phi(z|x) \Big\{ \log p_\theta(y|z, \cancel{x}) - \cancel{\log \tfrac{q_\phi(z|x)}{p_\theta(z|x)}} + \log \tfrac{q_\phi(z|x)}{p_\theta(z|x,y)} \Big\} \\ &\geq \int_{x,y} p(x,y) \int_z q_\phi(z|x) \log \tfrac{p_\theta(z|y) p_\theta(y)}{\sum_{y'} p_\theta(z|y') p_\theta(y')} \doteq \textbf{ELBO}_{\textbf{VC}} \qquad (5)\end{aligned}$$

Here, $p_\theta(y|z,x) = p_\theta(y|z)$ (figure 2), and the (freely chosen) variational posterior q_ϕ is assumed to depend only on x and to equal $p_\theta(z|x)$ (eliminating the second term).[2] **Maximising ELBO$_{\textbf{VC}}$ implicitly encourages $\boldsymbol{z}$ to learn a sufficient statistic for $\boldsymbol{y}|\boldsymbol{x}$**, i.e. $q_\phi(z|x) \to p(z|x,y)$. It can also be shown that **ELBO$_{\textbf{VC}}$ generalises Softmax Cross Entropy** (SCE), under above assumptions.

Two versions of the same class-conditional latent listributions:

- *Anticipated* class-conditional latent distributions $\underline{p_\theta(z|y)}$ are specified in ELBO$_{\text{VC}}$, encoded in the softmax layer, and need to be met for correct label predictions $p(y|x)$ to be output;
- *Empirical* class-conditional latent distributions are defined by $\underline{q_\phi(z|y)} \doteq \int_x q_\phi(z|x) p(x|y)$, i.e. by sampling $q_\phi(z|x)$ (parameterised by the neural network f_ω), given class samples $x \sim p(\text{x}|y)$.

In several scenarios that arise in practice, e.g. for finite samples from a continuous data domain $\mathcal{X}$ (e.g. images or sounds), or if classes are mutually exclusive, **ELBO$_{\textbf{VC}}$ is maximised if all latent representations of a class, hence the entire class-conditional distribution $\boldsymbol{q_\phi(z|y)}$, "collapse" to a point**, irrespective of any variance in $p_\theta(z|y)$. Since SCE is a special case of ELBO$_{\text{VC}}$, this suggests that softmax classifiers may learn over-concentrated latent distributions and so give *over-confident* and uncalibrated predictions (subject to the data distribution and model flexibility).

Aligning anticipated and empirical latent distributions: We align $p_\theta(z|y)$ and $q_\phi(z|y)$, or encourage $p_\theta(y|z)$ and $q_\phi(z|y)$ to be *consistent under Bayes' rule* (*cf* $p_\theta(x|z)$ and $q_\phi(z|x)$ in the ELBO, §2) by minimising $D_{\text{KL}}[\, q_\phi(\text{z}|y) \| \, p_\theta(\text{z}|y)]$, $\forall y \in \mathcal{Y}$, (weighted by $\beta > 0$) giving the full VC objective:

$$\mathcal{L}_{\textbf{VC}} = \int_{x,y} p(x,y) \Big\{ \int_z q_\phi(z|x) \log \tfrac{p_\theta(z|y) p_\pi(y)}{\sum_{y'} p_\theta(z|y') p_\pi(y')} - \beta \int_z q_\phi(z|y) \log \tfrac{q_\phi(z|y)}{p_\theta(z|y)} + \log p_\pi(y) \Big\}. \quad (6)$$

Taken incrementally, q_ϕ–terms of $\mathcal{L}_{\textbf{VC}}$ can be interpreted w.r.t. latent variable z as follows (see Fig. 1):

(i) maximising $\int_z q_\phi(z|x) \log p_\theta(y|z)$ may overfit $q_\phi(z|y)$ to δ_{z-z_y} for finite samples; [MLE]
(ii) adding *class priors* $\int_z q_\phi(z|y) \log p_\theta(z|y)$ constrains the MLE point estimates z_y [MAP]
(iii) adding *entropy* $-\int_z q_\phi(z|y) \log q_\phi(z|y)$ encourages $q_\phi(\text{z}|y)$ to "fill out" $p_\theta(\text{z}|y)$. [Bayesian]

VC abstracts a typical neural network classifier, giving interpretability to its components:

- the neural network up to the last layer (f_ω) transforms a mixture of unknown class-conditional data distributions $p(\text{x}|y)$ to a mixture of analytically defined latent distributions $p_\theta(\text{z}|y)$;
- assuming latent variables follow the anticipated class distributions $p_\theta(\text{z}|\text{y})$, the output layer applies Bayes' rule to give $p_\theta(\text{y}|\text{z})$ (see figure 2) and thus the class prediction $p(\text{y}|x)$ (by eqn 4).

3.1 Optimising the VC Objective

The second term of $\mathcal{L}_{\textbf{VC}}$ is not readily computable since $q_\phi(\text{z}|y)$ is implicit and cannot be evaluated only sampled from, as $z \sim q_\phi(\text{z}|x)$ (parameterised by f_ω) for class samples $x \sim p(\text{x}|y)$. We therefore *approximate* log ratios $\log \frac{q_\phi(z|y)}{p_\theta(z|y)}$ for each class y by training a binary classifier to distinguish $z \sim q_\phi(\text{z}|y)$ from $z \sim p_\theta(\text{z}|y)$ under an *auxiliary objective* $\mathcal{L}_{\textbf{aux}}$ with parameters ψ:

$$\mathcal{L}_{\textbf{aux}} = \int_y p(y) \Big\{ \int_z q_\phi(z|y) \log \sigma(T^y_\psi(z)) + \int_z p_\theta(z|y) \log(1 - \sigma(T^y_\psi(z)) \Big\} \qquad (7)$$

where $\sigma(x) = (1 + e^{-x})^{-1}$ is the logistic sigmoid, $T^y_\psi(z) = w_y^\top z + b_y$ and $\psi = \{w_y, b_y\}_{y \in \mathcal{Y}}$. This approach is *adversarial*: $\mathcal{L}_{\textbf{VC}}$ is maximised when log ratios give a *minimal* KL divergence (0), i.e. $q_\phi(z|y) = p_\theta(z|y)$ and $z \sim q_\phi(z|y)$ are indistinguishable from $z \sim p_\theta(z|y)$; whereas $\mathcal{L}_{\textbf{aux}}$ is maximised if the ratio is *maximal* and the distributions are optimally discriminated. See Algorithm 1 for a summary.

[2] We use notation "q_ϕ" by analogy to the VAE and later to distinguish $q_\phi(z|y)$, derived from $q_\phi(z|x)$, from $p_\theta(z|y)$.

4 Empirical Validation

We evaluate VC on tasks in the visual and text domains, to validate whether VC improves: **(H1)** uncertainty estimation and calibration; **(H2)** robustness to distribution shift; **(H3)** robustness to adversarial perturbations; and **(H4)** sample efficiency. We train under $\mathcal{L}_{\mathbf{VC}}$ with $q_\phi(\mathrm{z}|x) = \delta_{z - f_\omega(x)}$ where f_ω is a neural network; and $p_\theta(\mathrm{z}|y)$ are multi-variate Gaussians with learned mean and diagonal covariance. We compare classifiers trained under three objectives: standard softmax **CE** (MLE form, §3, (i)); **GM**, which adds class priors (MAP form, §3, (ii)) (*cf* Wan et al., 2018); and **VC**, which adds entropy $H(q_\phi(z|y))$ (Bayesian form, §3, (iii)). Further results are in Appendix E.

Accuracy and Calibration: We compare classification accuracy and calibration (*Expected Calibration Error*, ECE, see Appendix D) across CIFAR-10, CIFAR-100, and TINY-IMAGENET on two ResNet architectures (*WideResNet-28-10* (WRN) and *ResNet-50* (RNET)) (He et al., 2016; Zagoruyko & Komodakis, 2016).

	CIFAR-10				CIFAR-100				TINY-IMAGENET		
	CE	GM$^\diamond$	VC	vMF*	CE	GM $^\diamond$	VC	vMF*	CE	GM$^\diamond$	VC
Acc. (%, ↑)											
WRN	96.2 ± 0.1	95.0 ± 0.2	96.3 ± 0.2	-	80.3 ± 0.1	79.8 ± 0.2	80.3 ± 0.1	-	-	-	-
RNET	93.7 ± 0.1	93.0 ± 0.1	93.2 ± 0.1	94.0 ± 0.1	73.2 ± 0.1	74.2 ± 0.1	73.4 ± 0.1	69.94 ± 0.2	59.7 ± 0.2	59.3 ± 0.1	59.3 ± 0.1
ECE (%, ↓)											
WRN	3.1 ± 0.2	3.5 ± 0.3	**2.1** ± 0.2	-	11.1 ± 0.7	19.6 ± 0.4	**4.8** ± 0.3	-	-	-	-
RNET	3.8 ± 0.3	4.1 ± 0.2	**3.2** ± 0.2	5.9 ± 0.2	8.7 ± 0.2	10.5 ± 0.2	**5.1** ± 0.2	7.9 ± 0.3	12.3 ± 0.4	8.75 ± 0.2	**7.4** ± 0.5

Table 1: Classification Accuracy holds across models / data sets; Expected Calibration Error notably improves for VC (mean, std., 5 runs). $\star$ from (Scott et al., 2021), $\diamond$ re-implements (Wan et al., 2018)

Generalization under distribution shift: Models may encounter *distribution shift* relative to the training data and it may be important to know if a model's output is reliable, requiring **out-of-distribution (OOD) calibration**. We test on robustness benchmarks, CIFAR-10-C, CIFAR-100-C and TINY-IMAGENET-C (Hendrycks & Dietterich, 2019), which simulate distribution shift by synthetic corruptions of varying intensity.

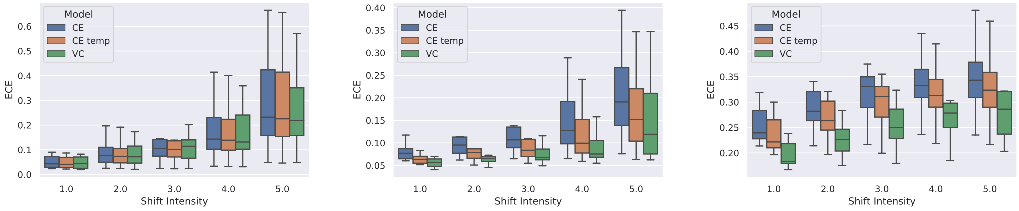

Figure 3: Calibration under distribution shift: boxes = quartiles, whiskers = min/max over 16 shift types. *(left)* CIFAR-10-C, *(middle)* CIFAR-100-C, *(right)* TINY-IMAGENET-C. Lower is better.

5 Conclusion

We present Variational Classification (VC), a latent variable model that generalises softmax cross-entropy classification, mirroring the relationship between the variational auto-encoder and the deterministic auto-encoder (§3). We show that softmax classification is a special case of VC under specific assumptions that are effectively "baked in" to a typical softmax layer. The latent VC model allows probabilistic interpretation of the roles played by the softmax layer and the layers beneath, and exposes an inconsistency that can arise between the latent distribution *expected* by the softmax layer and that *delivered* by the layers beneath. These distributions are not necessarily aligned in softmax classification, which is addressed by the VC objective. Experiments on image and text datasets show that, with marginal computational overhead and without increased hyper-parameters tuning, VC maintains prediction accuracy while significantly improving performance in terms of calibration, robustness to distribution shift and adversarial examples, and in low data regimes (Appendix E).

The VC framework provides novel theoretical insight into the highly familiar softmax classifier. We do, however, focus specifically on the *last* layer of a classifier, treating layers beneath as a "black-box". This leaves open questions as to how, and how well, the underlying network is able to perform its role in transforming a mixture of unknown (data) distributions to a mixture of specified (latent) distributions, or how that might be improved.

Acknowledgments and Disclosure of Funding

Carl is gratefully supported by an ETH AI Centre Postdoctoral Fellowships and a small projects grant from the Haslerstiftung (no. 23072). Mrinmaya acknowledges support from the Swiss National Science Foundation (Project No. 197155), a Responsible AI grant by the Haslerstiftung; and an ETH Grant (ETH-19 21-1).

References

Kemal Adem, Serhat Kiliçarslan, and Onur Cömert. Classification and diagnosis of cervical cancer with stacked autoencoder and softmax classification. *Expert Systems with Applications*, 115: 557–564, 2019.

Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. In *IEEE Transactions on Medical Imaging*, 2018.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

Ian J Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks. *arXiv preprint arXiv:1312.6082*, 2013.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Marcus Klasson, Cheng Zhang, and Hedvig Kjellström. A hierarchical grocery store image dataset with visual and semantic labels. In *IEEE Winter Conference on Applications of Computer Vision*, 2019.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, 2021.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *International Conference on Computer Vision*, 2015.

Milad Mirbabaie, Stefan Stieglitz, and Nicholas RJ Frick. Artificial intelligence in disease diagnostics: A critical review and classification on the current state of research guiding future direction. *Health and Technology*, 11(4):693–731, 2021.

Jishnu Mukhoti, Andreas Kirsch, Joost van Amersfoort, Philip HS Torr, and Yarin Gal. Deep deterministic uncertainty: A simple baseline. *arXiv e-prints*, pp. arXiv–2102, 2021.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *AAAI Conference on Artificial Intelligence*, 2015.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014.

Tyler R Scott, Andrew C Gallagher, and Michael C Mozer. von mises-fisher loss: An exploration of embedding geometries for supervised learning. In *International Conference on Computer Vision*, 2021.

Jacob Tiensuu, Maja Linderholm, Sofia Dreborg, and Fredrik Örn. Detecting exoplanets with machine learning: A comparative study between convolutional neural networks and support vector machines, 2019.

Weitao Wan, Yuanyi Zhong, Tianpeng Li, and Jiansheng Chen. Rethinking feature distribution for loss functions in image classification. In *Conference on Computer Vision and Pattern Recognition*, 2018.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

A Proofs

A.1 Optimising the ELBO$_{\text{VC}}$ w.r.t *q*

Rearranging Equation 5, the ELBO$_{\text{VC}}$ is optimised by

$$\begin{aligned}
&\arg\max_{q_\phi(z|x)} \int_x \sum_y p(x,y) \int_z q_\phi(z|x) \log p_\theta(y|z) \\
=&\arg\max_{q_\phi(z|x)} \int_x p(x) \int_z q_\phi(z|x) \sum_y p(y|x) \log p_\theta(y|z)
\end{aligned}$$

The integral over z is a $q_\phi(z|x)$-weighted sum of $\sum_y p(y|x)\log p_\theta(y|z)$ terms. Since $q_\phi(z|x)$ is a probability distribution, the integral is upper bounded by $\max_z \sum_y p(y|x)\log p_\theta(y|z)$. This maximum is attained *iff* support of $q_\phi(z|x)$ is restricted to $z^* = \arg\max_z \sum_y p(y|x)\log p_\theta(y|z)$ (which may not be unique). □

A.2 Optimising the VC objective w.r.t. *q*

Setting $\beta = 1$ in Equation 6 to simplify and adding a lagrangian term to constrain $q_\phi(z|x)$ to a probability distribution, we aim to find

$$\begin{aligned}
\arg\max_{q_\phi(z|x)} \int_x \sum_y p(x,y) \Big\{ \int_z & q_\phi(z|x) \log p_\theta(y|z) \\
- \int_z q_\phi(z|y) \log \tfrac{q_\phi(z|y)}{p_\theta(z|y)} \; + \; & \log p_\pi(y) \Big\} + \lambda(1 - \int_z q_\phi(z|x)) \; .
\end{aligned}$$

Recalling that $q_\phi(z|y) = \int_x q_\phi(z|x)p(x|y)$ and using calculus of variations, we set the derivative of this functional w.r.t. $q_\phi(z|x)$ to zero

$$\sum_y p(x,y) \Big\{ \log p_\theta(y|z) - (\log \tfrac{q_\phi(z|y)}{p_\theta(z|y)} + 1) \Big\} - \lambda = 0$$

Rearranging and diving through by $p(x)$ gives

$$\mathbb{E}_{p(y|x)}[\log q_\phi(z|y)] = \mathbb{E}_{p(y|x)}[\log p_\theta(y|z)p_\theta(z|y)] + c \; ,$$

where $c = -(1+\frac{\lambda}{p(x)})$. Further, if each label y occurs once with each x, due to sampling or otherwise, then this simplifies to

$$q_\phi(z|y^*)e^c = p_\theta(y^*|z)p_\theta(z|y^*) \; ,$$

which holds for all classes $y \in \mathcal{Y}$. Integrating over z shows $e^c = \int_z p_\theta(y|z)p_\theta(z|y)$ to give

$$q_\phi(z|y) = \tfrac{p_\theta(y|z)p_\theta(z|y)}{\int_z p_\theta(y|z)p_\theta(z|y)} = p_\theta(z|y) \tfrac{p_\theta(y|z)}{\mathbb{E}_{p_\theta(z|y)}[p_\theta(y|z)]} \; . \quad \square$$

We note, it is straightforward to include β to show

$$q_\phi(z|y) = p_\theta(z|y) \tfrac{p_\theta(y|z)^{1/\beta}}{\mathbb{E}_{p_\theta(z|y)}[p_\theta(y|z)^{1/\beta}]} \; .$$

B Justifying the Latent Prior in Variational Classification

Choosing Gaussian class priors in Variational classification can be interpreted in two ways:

Well-specified generative model: Assume data $x \in \mathcal{X}$ is generated from the hierarchical model: y → z → x, where $p(\mathrm{y})$ is categorical; $p(\mathrm{z}|y)$ are analytically known distributions, e.g. $\mathcal{N}(z; \mu_y, \Sigma_y)$; the dimensionality of z is not large; and $x = h(z)$ for an arbitrary invertible function $h : \mathcal{Z} \to \mathcal{X}$ (if $\mathcal{X}$ is of higher dimension than $\mathcal{Z}$, assume h maps one-to-one to a manifold in $\mathcal{X}$). Accordingly, $p(\mathrm{x})$ is a mixture of unknown distributions. If $\{p_\theta(\mathrm{z}|y)\}_\theta$ includes the true distribution $p(\mathrm{z}|y)$, variational classification effectively aims to invert h and learn the parameters of the true generative model. In practice, the model parameters and h^{-1} may only be identifiable up to some equivalence, but by reflecting the true latent variables, the learned latent variables should be semantically meaningful.

Miss-specified model: Assume data is generated as above, but with z having a large, potentially uncountable, dimension with complex dependencies, e.g. details of every blade of grass or strand of hair in an image. In general, it is impossible to learn all such latent variables with a lower dimensional model. The latent variables of a VC might learn a complex function of multiple true latent variables.

The first scenario is ideal since the model might learn disentangled, semantically meaningful features of the data. However, it requires distributions to be well-specified and a low number of true latent variables. For natural data with many latent variables, the second case seems more plausible but choosing $p_\theta(\mathrm{z}|y)$ to be Gaussian may nevertheless be justifiable by the Central Limit Theorem.

C Variational Classification Algorithm

Algorithm 1 Variational Classification (VC)

```
Input   $p_\theta(\mathrm{z}|y), q_\phi(\mathrm{z}|x), p_\pi(\mathrm{y}), T_\psi(z)$; learning rate schedule $\{\eta^t_\theta, \eta^t_\phi, \eta^t_\pi, \eta^t_\psi\}_t$
Initialise  $\theta, \phi, \pi, \psi;\ t \leftarrow 0$
while not converged do
    $\{x_i, y_i\}_{i=1}^m \sim \mathcal{D}$                    [sample batch from data distribution $p(\mathrm{x}, \mathrm{y})$]
    for z = {1 ... m} do
        $z_i \sim q_\phi(\mathrm{z}|x_i)$, $z'_i \sim p_\theta(\mathrm{z}|y_i)$        [e.g. $q_\phi(z|x_i) \doteq \delta_{z - f_\omega(x_i)}, \phi \doteq \omega \Rightarrow\ z_i = f_\omega(x_i)$]
        $p_\theta(y_i|z_i) = \frac{p_\theta(z_i|y_i) p_\pi(y_i)}{\sum_y p_\theta(z_i|y) p_\pi(y)}$
    end for
    $g_\theta \leftarrow \frac{1}{m} \sum_{i=1}^m \nabla_\theta \left[\log p_\theta(y_i|z_i) + p_\theta(z_i|y_i)\right]$
    $g_\phi \leftarrow \frac{1}{m} \sum_{i=1}^m \nabla_\phi \left[\log p_\theta(y_i|z_i) - T_\psi(z_i)\right]$        [e.g. using "reparameterisation trick"]
    $g_\pi \leftarrow \frac{1}{m} \sum_{i=1}^m \nabla_\pi \log p_\pi(y_i)$
    $g_\psi \leftarrow \frac{1}{m} \sum_{i=1}^m \nabla_\psi \left[\log \sigma(T_\psi(z_i)) + \log(1 - \sigma(T_\psi(z'_i))\right]$
    $\theta \leftarrow \theta + \eta^t_\theta\, g_\theta$,   $\phi \leftarrow \phi + \eta^t_\phi\, g_\phi$,   $\pi \leftarrow \pi + \eta^t_\pi\, g_\pi$,   $\psi \leftarrow \psi + \eta^t_\psi\, g_\psi$,     $t \leftarrow t + 1$
end while
```

D Calibration Metrics

One way to measure if a model is calibrated is to compute the expected difference between the confidence and expected accuracy of a model.

$$\mathbb{E}_{P(\hat{y}|x)}\Big[\mathbb{P}(\hat{y} = y | P(\hat{y}|x) = p) - p\Big] \tag{8}$$

This is known as expected calibration error (ECE) (Naeini et al., 2015). Practically, ECE is estimated by sorting the predictions by their confidence scores, partitioning the predictions in *M* equally spaced bins $(B_1 \dots B_M)$ and taking the weighted average of the difference between the average accuracy and average confidence of the bins. In our experiments we use 20 equally spaced bins.

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left| \mathit{acc}(B_m) - \mathit{conf}(B_m) \right| \tag{9}$$

E Further Results

E.1 Distribution Shift (continued)

When deployed in the wild, *natural* distributional shifts may occur in the data due to subtle changes in the data generation process, e.g. a change of camera. We test resilience to *natural* distributional shifts on two tasks: Natural Language Inference (NLI) and detecting whether cells are cancerous from microscopic images. NLI requires verifying if a hypothesis logically follows from a premise. Models are trained on the SNLI dataset (Bowman et al., 2015) and tested on the MNLI dataset (Williams et al., 2018) taken from more diverse sources. Cancer detection uses the CAMELYON17 dataset (Bandi et al., 2018) from the WILDs datasets (Koh et al., 2021), where the `train` and `eval` sets contain images from different hospitals.

Table 2 shows that the VC model achieves better calibration under these natural distributional shifts (**H2**). The CAMELYON17 (CAM) dataset has a relatively small number (1000) of training samples (hence wide error bars are expected), which combines distribution shift with a low data setting (**H4**) and shows that the VC model achieves higher (average) accuracy in this more challenging real-world setting.

	Accuracy (↑)		Calibration (↓)	
	CE	VC	CE	VC
NLI	**71.2** ± 0.1	**71.2** ± 0.1	7.3 ± 0.2	**3.4** ± 0.2
CAM	79.2 ± 2.8	**84.5** ± 4.0	8.4 ± 2.5	**1.8** ± 1.3

Table 2: Accuracy and Calibration (ECE) under distributional shift (mean, std. err., 5 runs)

We also test the ability to **detect OOD examples**. We compute the AUROC when a model is trained on CIFAR-10 and evaluated on the CIFAR-10 validation set mixed (in turn) with SVHN, CIFAR-100, and CELEBA (Goodfellow et al., 2013; Liu et al., 2015). We compare the VC and CE models using the probability of the predicted class $\arg\max_y p_\theta(y|x)$ as a means of identifying OOD samples.

Table 3 shows that the VC model performs comparably to the CE model. We also consider $p(\mathrm{z})$ as a metric to detect OOD samples and achieve comparable results, which is broadly consistent with the findings of (Grathwohl et al., 2019). Although the VC model learns to map the data to a more structured latent space and, from the results above, makes more calibrated predictions for OOD data, it does not appear to be better able to distinguish OOD data than a standard softmax classifier (CE) using the metrics tested (we note that "OOD" is a loosely defined term).

Model	SVHN	C-100	CelebA
$P_{\mathrm{CE}}(y\|x)$	0.92	0.88	0.90
$P_{\mathrm{VC}}(y\|z)$	0.93	0.86	0.89

Table 3: AUROC for the OOD detection task. Models are trained on CIFAR-10 and evaluated on in and out-of-distribution samples.

E.2 Adversarial Robustness

We test model robustness by measuring performance on adversarially generated images using the common *Fast Gradient Sign Method* (FGSM) of adversarial attack (Goodfellow et al., 2014). Perturbations are generated as $P = \epsilon \times \mathit{sign}\,(\mathcal{L}(x,y))$, where $\mathcal{L}(x,y)$ is the model loss for data sample x and correct class y; and ϵ is the *magnitude* of the attack. We compare all models trained on MNIST and CIFAR-10 against FGSM attacks of different magnitudes.

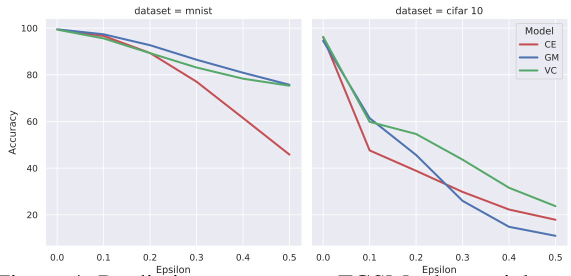

Figure 4: Prediction accuracy as FGSM adversarial attacks increase *(l)* MNIST; *(r)* CIFAR-10

Results in Figure 4 show that the VC model is consistently more adversarially robust relative to the standard CE model, across attack magnitudes on both datasets (**H3**).

E.3 Low Data Regime

	CE	GM	VC
MNIST	93.1 ± 0.2	**94.4** ± 0.1	**94.2** ± 0.2
CIFAR-10	52.7 ± 0.5	54.2 ± 0.6	**56.3** ± 0.6
AGNEWS	56.3 ± 5.3	61.5± 2.9	**66.3** ± 4.6

Table 4: Accuracy in low data regime (mean, std.err., 5 runs)

In many real-world settings, datasets may have relatively few data samples and it may be prohibitive or impossible to acquire more, e.g. historic data or rare medical cases. We investigate model performance when data is scarce on the hypothesis that a prior over the latent space enables the model to better generalise from fewer samples. Models are trained on 500 samples from MNIST, 1000 samples from CIFAR-10 and 50 samples from AGNEWS.

Results in Table 4 show that introducing the prior (GM) improves performance in a low data regime and that the additional entropy term in the VC model maintains or further improves accuracy (**H4**), particularly on the more complex datasets.

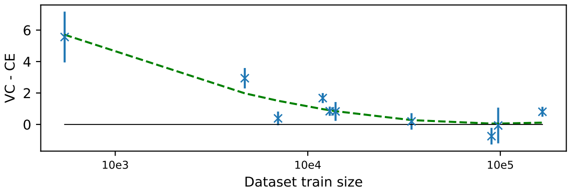

Figure 5: Accuracy increase of VC over CE on MedMNIST datasets of varying training set size (mean, std.err., 3 runs)

We further probe the relative benefit of the VC model over the CE baseline as the training sample size varies (**H4**) on MedMNIST, a collection of real-world medical datasets of varying sizes.

Figure 5 shows the increase in classification accuracy for the VC model relative to the CE model against number of training samples (log scale). The results show a clear trend that the benefit of the additional latent structure imposed in the VC model increases exponentially as the number of training samples decreases. Together with the results in Table 4, this suggests that the VC model offers most significant benefit for small, complex datasets.

E.4 Classification under Domain Shift

A comparison of accuracy between the VC and CE models under 16 different synthetic domain shifts. We find that VC performs comparably well as CE.

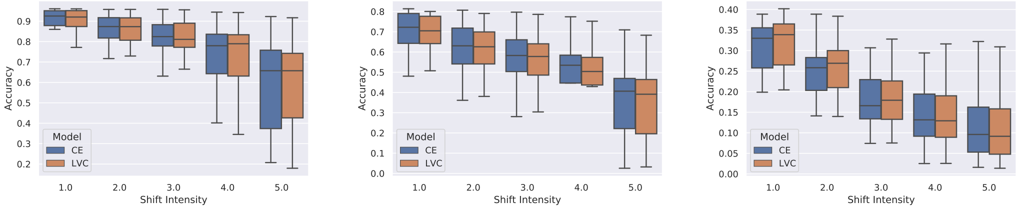

Figure 6: Classification accuracy under distributional shift: *(left)* CIFAR-10-C *(middle)* CIFAR-100-C *(right)* TINY-IMAGENET-C

E.5 OOD Detection

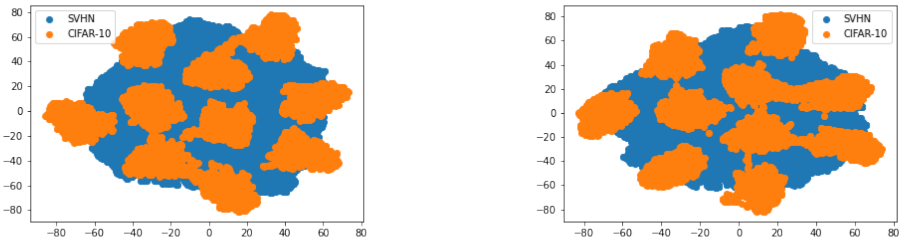

Figure 7: t-SNE plots of the feature space for a classifier trained on CIFAR-10. *(l)* Trained using CE. *(r)* Trained using VC. We posit that similar to CE, VC model is unable to meaningfully represent data from an entirely different distribution.

F Semantics of the latent space

To try to understand the semantics captured in the latent space, we use a pre-trained MNIST model on the *Ambiguous* MNIST dataset (Mukhoti et al., 2021). We interpolate between ambiguous 7's that are mapped close to the Gaussian clusters of classes of "1" and "2". It can be observed that traversing from the mean of the "7" Gaussian to that on the "1" class, the ambiguous 7's begin to look more like "1"s.

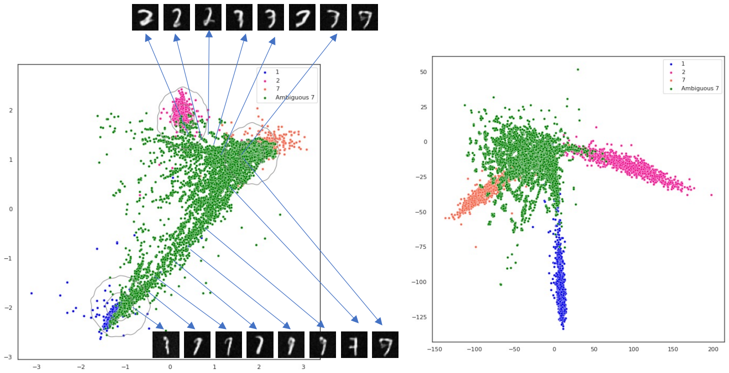

Figure 8: Interpolating in the latent space: Ambiguous MNIST when mapped on the latent space. *(l)* VC, *(r)* CE