A EVIDENCE LOWER BOUND AND ITS EXPECTATION

Next, we derive the Evidence Lower Bound and provide a closed-form expression for its expectation under the data distribution, which allows us to optimize our classifier.

A.1 DERIVATION OF THE EVIDENCE LOWER BOUND (ELBO)

The following is an ELBO for $\log p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$.

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}) = \int q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \log p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}) \, d\boldsymbol{z} \tag{17}$$

$$= \int q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y})} \, d\boldsymbol{z} \tag{18}$$

$$= \int q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \log \left[\frac{p_{\boldsymbol{\theta}}(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y})} \right] \, d\boldsymbol{z} \tag{19}$$

$$= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \right] + D_{KL} \left(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) || p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}) \right) \tag{20}$$

The second term of equation 20 is non-negative. Then, the first is an ELBO of $p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$:

$$\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \right] \tag{21}$$

A.2 EXPECTATION OF THE ELBO

Once we have proposed an ELBO, we calculate the expectation of that ELBO under the distribution of the data $\mathcal{D}$, which is what should be maximized.

$$\mathbb{E}_{q_{\mathcal{D}}(\boldsymbol{x},\boldsymbol{y})} \left[\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \right] = \iiint q_{\mathcal{D}}(\boldsymbol{x}, \boldsymbol{y}) q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \, d\boldsymbol{x} \, d\boldsymbol{y} \, d\boldsymbol{z} \tag{22}$$

$$= \iiint q_{\mathcal{D}}(\boldsymbol{x}, \boldsymbol{y}) q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \, d\boldsymbol{x} \, d\boldsymbol{y} \, d\boldsymbol{z} \tag{23}$$

$$\begin{aligned} = & \iiint q_{\mathcal{D}}(\boldsymbol{x}, \boldsymbol{y}) q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \log p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}) \, d\boldsymbol{x} \, d\boldsymbol{y} \, d\boldsymbol{z} \\ & + \iiint q_{\mathcal{D}}(\boldsymbol{x}, \boldsymbol{y}) q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \, d\boldsymbol{x} \, d\boldsymbol{y} \, d\boldsymbol{z} \end{aligned} \tag{24}$$

$$\begin{aligned} = & -CE \left(q_{\mathcal{D}}(\boldsymbol{x}, \boldsymbol{y}), \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \left[p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}) \right] \right) \\ & - \mathbb{E}_{q_{\mathcal{D}}(\boldsymbol{x},\boldsymbol{y})} \left[D_{KL} \left(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) || p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) \right) \right] \end{aligned} \tag{25}$$

$$\begin{aligned} = & -CE \left(q_{\mathcal{D}}(\boldsymbol{x}, \boldsymbol{y}), p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}) \right) \\ & - \mathbb{E}_{q_{\mathcal{D}}(\boldsymbol{x},\boldsymbol{y})} \left[D_{KL} \left(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) || p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) \right) \right] \end{aligned} \tag{26}$$

B VARIATIONAL CLASSIFIER TRAINING ALGORITHMS

Next, we present the algorithms for training each of the two alternatives to implement the proposed model using neural networks.

B.1 LVVC TRAINING ALGORITHM

Algorithm 1 Training algorithm for Learnable Variance Variational Classifier

Input: Dataset $\mathcal{D}$
Output: $\boldsymbol{\theta}$, $\boldsymbol{\phi} = \{\boldsymbol{\phi'}, \boldsymbol{\phi_\mu}, \boldsymbol{\phi_\sigma}\}$
  $\boldsymbol{\theta}$, $\boldsymbol{\phi} \leftarrow$ Initialize parameters
  repeat
    $\mathcal{D}^N \leftarrow \{(x^{(i)}, \boldsymbol{y}^{(i)})\}_{i=1}^N$ (Random minibatch from $\mathcal{D}$)
    for $i = 1$ to N **do**
      $\tilde{\boldsymbol{z}}^{(i)} \leftarrow h_{\boldsymbol{\phi'}}(x^{(i)})$
      $\boldsymbol{\mu}_z^{(i)}, \boldsymbol{\sigma}_z^{(i)} \leftarrow h_{\boldsymbol{\phi_\mu}}(\tilde{\boldsymbol{z}}^{(i)}),\ h_{\boldsymbol{\phi_\sigma}}(\tilde{\boldsymbol{z}}^{(i)})$
      $\boldsymbol{\epsilon} \leftarrow$ Draw L samples from $\mathcal{N}(0, I)$
      $\boldsymbol{z}^{(i,l)} \leftarrow \boldsymbol{\mu}_z^{(i)} + \boldsymbol{\sigma}_z^{(i)} \cdot \boldsymbol{\epsilon}^{(l)}$
      $\tilde{\boldsymbol{y}}^{(i)} \leftarrow \frac{1}{L} \sum_{l=1}^{L} f_{\boldsymbol{\theta}}(\boldsymbol{z}^{(i,l)})$
    end for
    $\mathcal{L} \leftarrow \frac{1}{N} \sum_{i=1}^{N} CE(\boldsymbol{y}^{(i)}, \tilde{\boldsymbol{y}}^{(i)}) + \frac{1}{2} \sum_{j=1}^{k} \log \frac{\boldsymbol{\sigma}_{\boldsymbol{\theta}}^2(\boldsymbol{y}^{(i)})}{(\boldsymbol{\sigma}_z^{(i)})^2} - 1 + \frac{((\boldsymbol{\mu}_z^{(i)})_j - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{y}^{(i)})_j)^2}{\boldsymbol{\sigma}_{\boldsymbol{\theta}}^2(\boldsymbol{y}^{(i)})} + \frac{(\boldsymbol{\sigma}_z^{(i)})^2}{\boldsymbol{\sigma}_{\boldsymbol{\theta}}^2(\boldsymbol{y}^{(i)})}$
    $\boldsymbol{\theta}$, $\boldsymbol{\phi} \leftarrow$ Update using gradients of $\mathcal{L}$
  until convergence of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$

B.2 FVVC TRAINING ALGORITHM

Algorithm 2 Training algorithm for Fixed Variance Variational Classifier

Input: Dataset $\mathcal{D}$
Output: $\boldsymbol{\theta}$, $\boldsymbol{\phi}$
  $\boldsymbol{\theta}$, $\boldsymbol{\phi} \leftarrow$ Initialize parameters
  repeat
    $\mathcal{D}^N \leftarrow \{(x^{(i)}, \boldsymbol{y}^{(i)})\}_{i=1}^N$ (Random minibatch from $\mathcal{D}$)
    for $i = 1$ to N **do**
      $\boldsymbol{\mu}_z^{(i)} \leftarrow h_{\boldsymbol{\phi}}(x)$
      $\boldsymbol{\epsilon} \leftarrow$ Draw L samples from $\mathcal{N}(0, I)$
      $\boldsymbol{z}^{(i,l)} \leftarrow \boldsymbol{\mu}_z^{(i)} + \boldsymbol{\sigma}_{\boldsymbol{\theta}} \cdot \boldsymbol{\epsilon}^{(l)}$
      $\tilde{\boldsymbol{y}}^{(i)} \leftarrow \frac{1}{L} \sum_{l=1}^{L} f_{\boldsymbol{\theta}}(\boldsymbol{z}^{(i,l)})$
    end for
    $\mathcal{L} \leftarrow \frac{1}{N} \sum_{i=1}^{N} CE(\boldsymbol{y}^{(i)}, \tilde{\boldsymbol{y}}^{(i)}) + \frac{1}{2\boldsymbol{\sigma}_{\boldsymbol{\theta}}^2} \sum_{j=1}^{k} ((\boldsymbol{\mu}_z^{(i)})_j - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{y}^{(i)})_j)^2$
    $\boldsymbol{\theta}$, $\boldsymbol{\phi} \leftarrow$ Update using gradients of $\mathcal{L}$
  until convergence of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$

C KL DIVERGENCE DERIVATIONS

C.1 KL DIVERGENCE BETWEEN TWO ISOTROPIC MULTIVARIATE GAUSSIAN DISTRIBUTIONS

To begin with, we should note that the density function of a multivariate Normal distribution $p = N(\boldsymbol{\mu_p}, \Sigma_p))$ is of the form:

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{k/2}|\Sigma_p|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_p})^T\Sigma_p^{-1}(\boldsymbol{x}-\boldsymbol{\mu_p})\right) \tag{27}$$

Therefore, the KL divergence of such a distribution with another multivariate Normal distribution $q = \mathcal{N}(\boldsymbol{\mu_q}, \Sigma_q))$ remains:

$$D_{KL}(p||q) = \mathbb{E}_p[\log p - \log q] \tag{28}$$

$$= \mathbb{E}_p\left[\frac{1}{2}\log\frac{|\Sigma_q|}{|\Sigma_p|} - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_p})^T\Sigma_p^{-1}(\boldsymbol{x}-\boldsymbol{\mu_p}) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_q})^T\Sigma_q^{-1}(\boldsymbol{x}-\boldsymbol{\mu_q})\right] \tag{29}$$

$$\begin{aligned} = \frac{1}{2}\log\frac{|\Sigma_q|}{|\Sigma_p|} &- \frac{1}{2}\mathbb{E}_p\left[(\boldsymbol{x}-\boldsymbol{\mu_p})^T\Sigma_p^{-1}(\boldsymbol{x}-\boldsymbol{\mu_p})\right] \\ &+\frac{1}{2}\mathbb{E}_p\left[(\boldsymbol{x}-\boldsymbol{\mu_q})^T\Sigma_q^{-1}(\boldsymbol{x}-\boldsymbol{\mu_q})\right] \end{aligned} \tag{30}$$

$$\begin{aligned} = \frac{1}{2}\log\frac{|\Sigma_q|}{|\Sigma_p|} &- \frac{1}{2}\mathbb{E}_p\left[\mathrm{Tr}\{(\boldsymbol{x}-\boldsymbol{\mu_p})^T\Sigma_p^{-1}(\boldsymbol{x}-\boldsymbol{\mu_p})\}\right] \\ &+\frac{1}{2}\mathbb{E}_p\left[(\boldsymbol{x}-\boldsymbol{\mu_q})^T\Sigma_q^{-1}(\boldsymbol{x}-\boldsymbol{\mu_q})\right] \end{aligned} \tag{31}$$

$$\begin{aligned} = \frac{1}{2}\log\frac{|\Sigma_q|}{|\Sigma_p|} &- \frac{1}{2}\mathbb{E}_p\left[\mathrm{Tr}\{(\boldsymbol{x}-\boldsymbol{\mu_p})^T(\boldsymbol{x}-\boldsymbol{\mu_p})\Sigma_p^{-1}\}\right] \\ &+\frac{1}{2}\mathbb{E}_p\left[(\boldsymbol{x}-\boldsymbol{\mu_q})^T\Sigma_q^{-1}(\boldsymbol{x}-\boldsymbol{\mu_q})\right] \end{aligned} \tag{32}$$

$$\begin{aligned} = \frac{1}{2}\log\frac{|\Sigma_q|}{|\Sigma_p|} &- \frac{1}{2}\mathrm{Tr}\{\mathbb{E}_p\left[(\boldsymbol{x}-\boldsymbol{\mu_p})^T(\boldsymbol{x}-\boldsymbol{\mu_p})\Sigma_p^{-1}\right]\} \\ &+\frac{1}{2}\mathbb{E}_p\left[(\boldsymbol{x}-\boldsymbol{\mu_q})^T\Sigma_q^{-1}(\boldsymbol{x}-\boldsymbol{\mu_q})\right] \end{aligned} \tag{33}$$

$$\begin{aligned} = \frac{1}{2}\log\frac{|\Sigma_q|}{|\Sigma_p|} &- \frac{1}{2}\mathrm{Tr}\{\mathbb{E}_p\left[(\boldsymbol{x}-\boldsymbol{\mu_p})^T(\boldsymbol{x}-\boldsymbol{\mu_p})\right]\Sigma_p^{-1}\} \\ &+\frac{1}{2}\mathbb{E}_p\left[(\boldsymbol{x}-\boldsymbol{\mu_q})^T\Sigma_q^{-1}(\boldsymbol{x}-\boldsymbol{\mu_q})\right] \end{aligned} \tag{34}$$

$$= \frac{1}{2}\log\frac{|\Sigma_q|}{|\Sigma_p|} - \frac{1}{2}\mathrm{Tr}\{\Sigma_p\Sigma_p^{-1}\} + \frac{1}{2}\mathbb{E}_p\left[(\boldsymbol{x}-\boldsymbol{\mu_q})^T\Sigma_q^{-1}(\boldsymbol{x}-\boldsymbol{\mu_q})\right] \tag{35}$$

$$= \frac{1}{2}\log\frac{|\Sigma_q|}{|\Sigma_p|} - \frac{k}{2} + \frac{1}{2}\mathbb{E}_p\left[(\boldsymbol{x}-\boldsymbol{\mu_q})^T\Sigma_q^{-1}(\boldsymbol{x}-\boldsymbol{\mu_q})\right] \tag{36}$$

$$= \frac{1}{2}\log\frac{|\Sigma_q|}{|\Sigma_p|} - \frac{k}{2} + \frac{1}{2}\mathbb{E}_p\left[(\boldsymbol{\mu_p}-\boldsymbol{\mu_q})^T\Sigma_q^{-1}(\boldsymbol{\mu_p}-\boldsymbol{\mu_q}) + \mathrm{Tr}\{\Sigma_q^{-1}\Sigma_p\}\right] \tag{37}$$

$$= \frac{1}{2}\left[\log\frac{|\Sigma_q|}{|\Sigma_p|} - k + (\boldsymbol{\mu_p}-\boldsymbol{\mu_q})^T\Sigma_q^{-1}(\boldsymbol{\mu_p}-\boldsymbol{\mu_q}) + \mathrm{Tr}\{\Sigma_q^{-1}\Sigma_p\}\right] \tag{38}$$

In line 31 we apply the fact that the trace of a scalar is equal to the scalar itself. Line 32 is because trace is invariant under cyclic permutations. Finally, in line 37 we apply equation 380 from Petersen et al. (2008).

When the distributions are isotropic, i.e., $\Sigma_p = \text{diag}((\sigma_p)_1^2, \dots (\sigma_p)_k^2)$ and $\Sigma_q = \text{diag}((\sigma_q)_1^2, \dots (\sigma_q)_k^2)$ the following equations are satisfied:

$$\log \frac{|\Sigma_q|}{|\Sigma_p|} = \log \frac{\prod_{j=1}^k (\sigma_q)_j^2}{\prod_{j=1}^k (\sigma_p)_j^2} = \sum_{j=1}^k \log \frac{(\sigma_q)_j^2}{(\sigma_p)_j^2} \tag{39}$$

$$(\boldsymbol{\mu_p} - \boldsymbol{\mu_q})^T \Sigma_q^{-1} (\boldsymbol{\mu_p} - \boldsymbol{\mu_q}) = \sum_{j=1}^k \frac{((\mu_p)_j - (\mu_q)_j)^2}{(\sigma_q)_j^2} \tag{40}$$

$$\text{Tr}\{\Sigma_q^{-1} \Sigma_p\} = \sum_{j=1}^k \frac{(\sigma_p)_j^2}{(\sigma_q)_j^2} \tag{41}$$

Therefore, it is satisfied that the KL divergence between the two distributions described p y q equals to:

$$D_{KL}(p||q) = \frac{1}{2} \sum_{j=1}^k \log \frac{(\sigma_q)_j^2}{(\sigma_p)_j^2} - 1 + \frac{((\mu_p)_j - (\mu_q)_j)^2}{(\sigma_q)_j^2} + \frac{(\sigma_p)_j^2}{(\sigma_q)_j^2} \tag{42}$$

C.2 Maximization of KL Divergence between two Isotropic Multivariate Gaussian Distributions with fixed means magnitudes and variance

Given two distributions $p = \mathcal{N}(\boldsymbol{\mu_p}, \text{diag}(\sigma^2))$ and $q = \mathcal{N}(\boldsymbol{\mu_q}, \text{diag}(\sigma^2))$ such that $||\boldsymbol{\mu_p}||$, $||\boldsymbol{\mu_q}||$ and σ are fixed, and following the equation 42, we have that:

$$D_{KL}(p||q) = \frac{1}{2\sigma^2} \sum_{j=1}^k ((\mu_p)_j - (\mu_q)_j)^2 = \frac{1}{2\sigma^2} (||\boldsymbol{\mu}_p||^2 + ||\boldsymbol{\mu}_q||^2) - \frac{1}{\sigma^2} \langle \boldsymbol{\mu}_p, \boldsymbol{\mu}_q \rangle \tag{43}$$

Since the magnitudes of the means and variance are fixed, we have that maximizing the KL Divergence in this scenario is equivalent to minimizing the inner product between the means.

D PROOF OF COROLLARY 1.1

Corollary 1.1 *Let ϵ, V, and f be as defined in Lemma 1, and furthermore let f be linear. Then for every $u,\ v \in V$, if $-v \in V$:*

$$\langle u, v\rangle - \epsilon||u||||v|| \leq \langle f(u), f(v)\rangle \leq \langle u, v\rangle + \epsilon||u||||v|| \tag{16}$$

Proving corollary 1.1 is equivalent to proving:

$$|\langle f(u), f(v)\rangle - \langle u, v\rangle| \leq \epsilon||u||||v|| \tag{44}$$

Then we have that if u or v are 0, then the corollary is immediately satisfied. Otherwise, if u and v are unit vectors and $||u+v|| \geq ||u-v||$, we have that:

$$\begin{aligned}
4\,|\langle f(u), f(v)\rangle - \langle u, v\rangle| &\leq \left|||f(u)+f(v)||^2 - ||f(u)-f(v)||^2 - 4\langle u, v\rangle\right| && (45)\\
&= \left|||u+v||^2 - ||u-v||^2 + \epsilon\left(||u+v||^2 + ||u-v||^2\right) - 4\langle u, v\rangle\right| && (46)\\
&= \left|4\langle u, v\rangle + \epsilon\left(||u+v||^2 + ||u-v||^2\right) - 4\langle u, v\rangle\right| && (47)\\
&= 2\epsilon\left|||u||^2 + ||v||^2\right| && (48)\\
&= 4\epsilon && (49)
\end{aligned}$$

In lines 45 and 47 we use the polarization identity and in line 48 the parallelogram law.

Otherwise, if $||u+v|| < ||u-v||$, we have that equation 46 remains as:

$$4\,|\langle f(u), f(v)\rangle - \langle u, v\rangle| \leq \left|||u+v||^2 - ||u-v||^2 - \epsilon\left(||u+v||^2 + ||u-v||^2\right) - 4\langle u, v\rangle\right| \tag{50}$$

but the final result is the same because of the absolute value.

Finally, if u or v are not unit vectors we can reduce it to the previous case:

$$\begin{aligned}
|\langle f(u), f(v)\rangle - \langle u, v\rangle| &= \left|\left\langle f\left(\frac{u}{||u||}\right), f\left(\frac{v}{||v||}\right)\right\rangle - \left\langle \frac{u}{||u||}, \frac{v}{||v||}\right\rangle\right| ||u||||v|| && (51)\\
&\leq \epsilon||u||||v||\ \square && (52)
\end{aligned}$$

E TRAINING FORMULAS FOR DATASETS

The formulas for training each of the proposed systems and dataset are given below.

E.1 MNIST

In all cases we use the SGD optimizer with weight decay of 0.0001 and momentum of 0.9. We train the model for 160 epochs starting with a learning rate of 0.1 and divide it by 10 at epochs 80 and 120. In addition, we use a minibatch size of 128. Finally, we follow the common formula of data augmentation during training: 4 pixels are added on each side and a 32×32 crop is randomly sampled from the padded image or its horizontal flip.

E.2 CIFAR-10

In all cases we use the SGD optimizer with weight decay of 0.0001 and momentum of 0.9. For ResNet-20 and ResNet-56 we train the model for 160 epochs starting with a learning rate of 0.1 and divide it by 10 at epochs 80 and 120. For ResNet-110 the formula is identical but we introduce a warm up of the learning rate for 5 epochs starting from 0.01. In addition, we use a minibatch size of 128. Finally, we follow the common formula of data augmentation during training: 4 pixels are added on each side and a 32×32 crop is randomly sampled from the padded image or its horizontal flip.

E.3 CIFAR-100

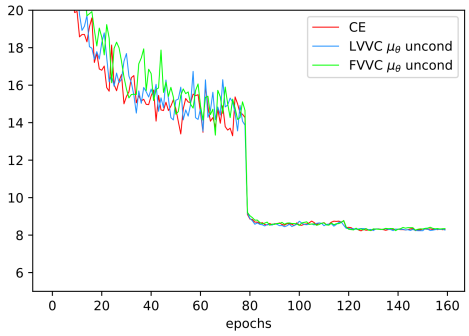
The training formula for CIFAR-100 is identical to that for CIFAR-10.

E.4 IMAGENET

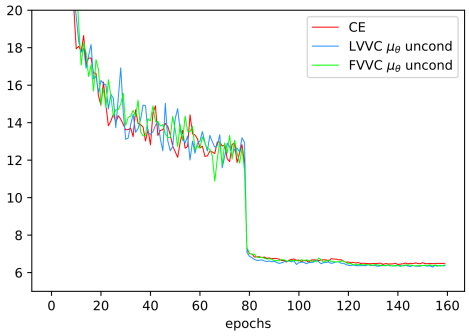
For ImageNet and ResNet-34, we train the models for 120 epochs with an SGD optimizer with weight decay of 0.0001 and momentum of 0.9 with a learning rate of 0.1, which we divided by 10 every 30 epochs. During training, a 224×224 random crop or its horizontal flip serves as input to the extractor embeddings for data augmentation.

F EVOLUTION OF ACCURACY DURING TRAINING

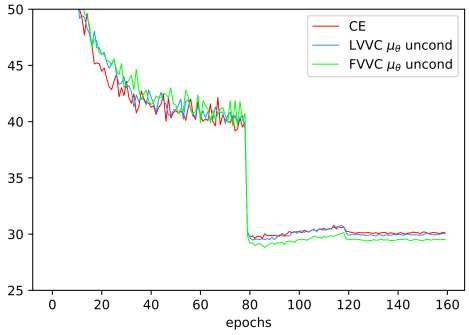
It is important to observe the evolution of the accuracy during the training. Analyzing these evolutions we can observe that, especially in the most complicated datasets, in the first epochs, where the learning rate is higher, the vanilla classifier performs better than the different VCs shown. However, it is in the last epochs of the training, with a lower learning rate, that the VC outperforms the vanilla classifier. Intuitively, we can think that the cost function is more complex in the VC because it is the sum of two terms. This fact makes the convergence more complex and there are more local minima far from each other. It is worth noting that in figure 5f we plot the evolution of the error when the means of the different classes are imposed to be orthogonal and in this case the convergence is faster than in figure 5e where no condition is imposed on the means. Giving fewer degrees of freedom to the objective distributions seems to result in the error decreasing faster in the first epochs.



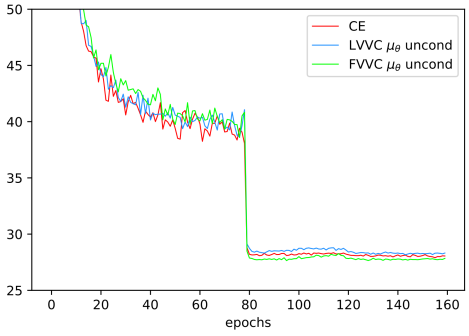
(a) Top-1 Error(%) in CIFAR-10 and ResNet-20



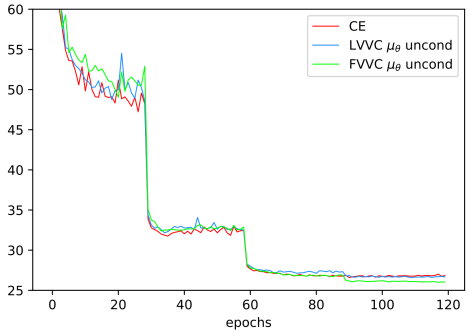
(b) Top-1 Error(%) in CIFAR-10 and ResNet-110



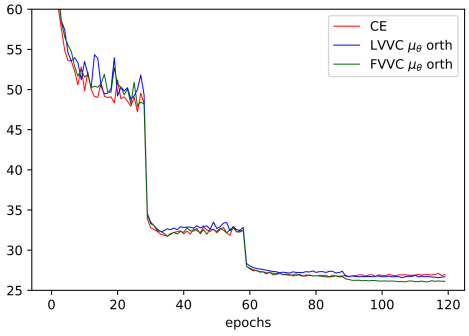
(c) Top-1 Error(%) in CIFAR-100 and ResNet-56



(d) Top-1 Error(%) in CIFAR-100 and ResNet-110



(e) Top-1 Error(%) in ImageNet and ResNet-34 ($\boldsymbol{\mu_\theta}$ uncond)



(f) Top-1 Error(%) in ImageNet and ResNet-34 ($\boldsymbol{\mu_\theta}$ orth)

Figure 5: Evolution of Top-1 Error in the vanilla classifier and Variational Classifiers for the different scenarios compared.

G UNCERTAINTY EXTRA MATERIAL

In this appendix, we provide additional information to verify the calibration of Variational Classifier predictions compared to other methods. For this purpose, we present the computational cost of each method for comparison, and expand on the analysis from section 5.2.

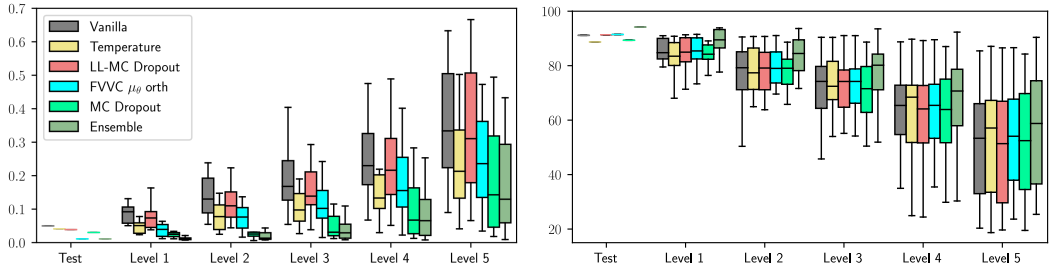
G.1 COMPUTATIONAL COST OF EACH METHOD COMPARED

Table 2: Computational time and memory of the compared methods. N is the number of samples in sampling methods; K the number of models in ensemble; f_e and f_c the number of flops to process an input through the embeddings extractor and last layer, respectively; x_t and x_v the size of test and validation datasets; and m the storage of the full model. We must note that $t_c << t_e$ for most cases.

Method	Compute/x_t	Storage
Vanilla	$f_e + f_c$	m
Temperature Scaling	$(f_e + f_c)(1 + x_v/x_t)$	m
Ensemble	$N(f_e + f_c)$	Nm
MC Dropout	$K(f_e + f_c)$	m
LL-MC Dropout	$f_e + Nf_c$	m
Variational Classifier	$f_e + Nf_c$	m

G.2 RESULTS ON CORRUPTED CIFAR-10

In Figure 6a, we can observe that for the uncorrupted test data of CIFAR-10, the VC performs the best compared to even the methods with significantly higher computational cost. On the other hand, for corrupted data, MC Dropout and Ensemble outperform the others again. In this case, the temperature scaling and the VC have a similar level of calibration, which is considerably better than the remaining two methods. Regarding accuracy, we see, that once again the ensemble is more accurate than the rest, among which there are no significant differences in this aspect.



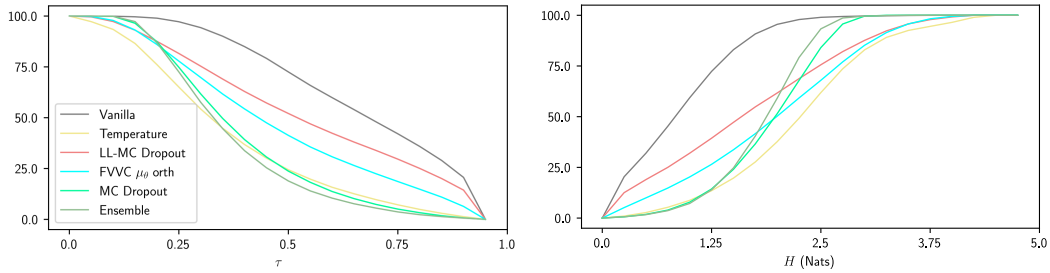
(a) CIFAR-10: ECE

(b) CIFAR-10: Top-1 Accuracy (%)

Figure 6: ECE and Top-1 Accuracy in the methods compared and for the different levels of corruption for CIFAR-10. Each box plot shows the quartiles across all corruption types for each level.

G.3 RESULTS ON OOD FOR CIFAR-100-TRAINED MODEL

In this case, we trained a ResNet-56 model with CIFAR-100. As shown in Figure 7a, we observe that the confidence of predictions is lower for the more computationally expensive methods and for Temperature Scaling. The result is more surprising in Figure 7b, where both Ensemble and MC Dropout have almost all predictions with entropy below approximately 3 nats, while the maximum entropy is $\log(100) \approx 4.6052$. However, very few predictions have extremely low entropy. Conversely, this does not happen with the other less costly methods. Additionally, similar to what happened with CIFAR-10, the VC performs worse than Temperature Scaling and better than MC Dropout in the last layer.



(a) % of predictions with confidence greater than τ (b) % of predictions with entropy lower than H

Figure 7: Confidence and entropy for predictions of an OOD dataset (SVHN) in a model trained with CIFAR-100.