DICE
DIVERSITY IN ENSEMBLE WITH CONDITIONAL ENTROPY BOTTLENECK THROUGH MUTUAL INFORMATION NEURAL ESTIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep ensembles perform better than a single network thanks to the diversity among their members. Recent approaches regularize predictions to increase diversity; however, they also drastically decrease individual members performances. In this paper, we argue that learning strategies for ensembles need to tackle the *trade-off between ensemble diversity and individual accuracies*. Motivated by information theory arguments, we introduce a novel training criterion, DICE, that enforces *diversity among members' features*. Leveraging variational approximations and recent advances in neural estimation of conditional mutual information, DICE deletes spurious correlations by *adversarial conditional features alignment*. Features extracted from pairs of members should only share information useful for target class prediction without being conditionally redundant: thus, besides the classification loss with information bottleneck, we adversarially prevent features from being predictable from each other. It reduces simultaneous errors while protecting class information. We obtain state-of-the-art accuracy results on CIFAR-10/100: e.g., an ensemble of 5 networks trained with DICE matches an ensemble of 7 trained independently. We improve calibration, uncertainty estimation in- and out- domain, out-of-distribution detection, before and after temperature scaling.

1 INTRODUCTION

Averaging the predictions of several models can significantly improve the generalization ability of a predictive system. Due to its effectiveness, **ensemble** has been a popular research topic (Nilsson, 1965; Hansen & Salamon, 1990; Wolpert, 1992; Krogh & Vedelsby, 1995; Breiman, 1996; Dietterich, 2000; Zhou et al., 2002; Kuncheva & Whitaker, 2003; Rokach, 2010) as a simple alternative of bayesian methods (Blundell et al., 2015; Gal & Ghahramani, 2016). It is currently the de facto solution for many machine learning applications and **competitions** such as Kaggle.

Intuitively, M identical members do not contain more information than a single one: the significant accuracy boost is therefore due to the predictions' diversity. The reduction factor of the variance in the Bias-Variance Decomposition (Kohavi et al., 1996) equals to M only when errors are uncorrelated (Krogh & Vedelsby, 1995; Kuncheva) (see Appendix E.1). In deep ensembles (Lakshminarayanan et al., 2016), the weights are traditionally trained independently: diversity among members only relies on the randomness of the initialization and of the learning procedure. Figure 1 shows that it does not not fully leverage additional members and that benefits get quickly plateaued.

To obtain more diverse ensembles, we could adapt the training samples through bagging (Breiman, 1996) and bootstrapping (Efron & Tibshirani, 1994), but a reduction of training samples has a negative impact on members with multiple local minima (Lee et al., 2015). Sequential boosting does not scale well for time-consuming deep learners that overfit their training dataset. Liu & Yao (1999a;b); Brown et al. (2005b) tried to explicitly quantify the diversity and regularized the members into having negatively correlated errors. However, the ensemble accuracy remains almost unchanged when applied with deep learners (Shui et al., 2018; Pang et al., 2019): while members should predict the same target, they force disagreements and therefore increase members' bias. It highlights our main challenge: the **trade-off between ensemble diversity and individual accuracies**.

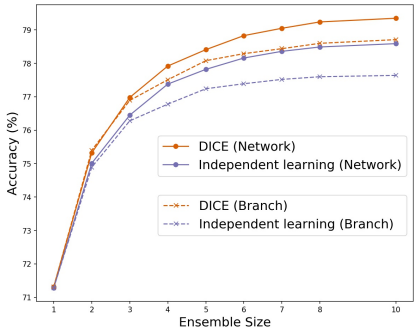

Figure 1: **DICE better leverages ensemble size**. Without weights sharing, 5 networks trained with DICE match 7 networks trained independently. With low-level weights sharing, 4 branches trained with DICE match 7 traditional branches. Dataset: CIFAR-100. Member: ResNet-32. See details in Table 7.

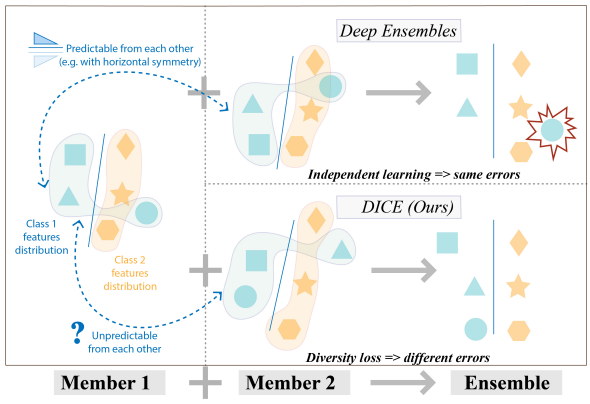

Figure 2: **Outline**. To delete spurious correlations, DICE adversarially prevents features from being predictable from each other *conditionally* upon the class: features extracted by two members from one input (●,●) should not share more information that when features are from two inputs from the same class (●,▲).

We took an opposite line as we believe that all members should predict the same thing, but not for the same reasons. Therefore, to protect the ability of each member to predict correctly, the diversity should be enforced in the features and not in the predictions. Intuitively, to maximize the impact of a new member, extracted features should bring information about the target that is yet absent therefore unpredictable from other members' features. They should fit different modes and have uncorrelated bias to have non-simultaneous errors: they will not fail on the same hard examples.

We draw inspiration from Information Bottleneck (IB) principles, (Tishby et al., 2000; Alemi et al., 2017) that force learned representation to be concise by forgetting the task-irrelevant factors: it limits overfitting while increasing robustness. The **Minimum Necessary Information** (MNI) criterion (Fischer, 2020) for deep ensembles leads to reducing the mutual information (MI) between the features and the inputs, but also the conditional mutual information (CMI) between features conditioned on the target. We make our **features conditionally independent** to reduce the covariance in predictions, by aligning the conditional features distributions. As MI can detect arbitrary dependencies between random variables (not just linear ones, as seen in Figure 2), reducing CMI increases the distance between pairs of members: it promotes diversity without decreasing the capacity of extracted features to predict the target. We build upon recent neural approaches (Belghazi et al., 2018) for MI estimation from samples, based on the Donsker-Varadhan representation of the KL.

We summarize our contributions as follows:

- We introduce DICE, a new framework to explicitly increase the diversity between members by minimizing the conditional mutual information (CMI) between members' features.
- We justify our training objective with arguments from information theory.
- We make our criterion tractable through adversarial neural estimation of CMI.

We empirically demonstrate consistent improvements in accuracy and robustness on CIFAR-10/100 (Krizhevsky et al., 2009). We analyze the two components of our loss and their relative impact on the accuracy-diversity trade-off. We give hints about our adversarial discriminator' behaviour while improving out-of-distribution detection. Finally, we extend DICE for online co-distillation.

2 DICE MODEL

Notations Given an input distribution X, a network θ is trained to extract the best possible dense features Z to model the distribution $p_\theta(Y|X)$ over the targets, that should be close to the true dirac. Our approach is designed for ensembles with M members $\theta_m, m \in \{1, \dots, M\}$ that extract Z_m. In the branch-based setup, members share low-level weights to reduce the computation cost. We average members predictions in inference. We initially consider an ensemble of $M = 2$ members.

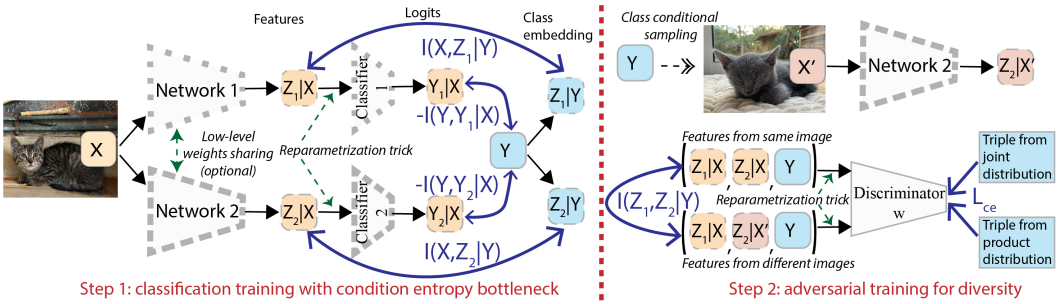

Figure 3: **Overview of our learning scheme**. Blue arrows represent training criteria: (1) classification with conditional entropy bottleneck for individual accuracy, and (2) adversarial training for increased diversity. X and X' belong to the same class Y for *conditional* distribution alignment.

Summary First, we train each member for *classification* with *conditional entropy bottleneck*. Second, the discriminator learns to distinguish features sampled from one image versus features sampled from two images belonging to the same class, conditioned on the class Y. Simultaneously, both members *adversarially* delete spurious correlations found only when features are extracted from the same image, which would otherwise simplify the discriminator task. In conclusion, our members learn to classify while having conditionally uncorrelated features for increased diversity and accuracy. Our procedure, summarized in Figure 3, is motivated by the following theoretical findings.

2.1 Minimum Necessary Information

2.1.1 Mutual information formulation

The Minimum Necessary Information (MNI) criterion from (Fischer, 2020) aims at finding minimal statistics. In deep ensembles, Z_1 and Z_2 should capture only *minimal information* from X, while preserving the *necessary information* about the task Y. As entropy measures information, we derive:

$$I(X;Y) = I(X;Z_1) = I(X;Z_2) = I(Z_1;Z_2) = I(Z_1;Y) = I(Z_2;Y).$$

We first consider separately the two Markov chains $Z_1 \leftarrow X \leftrightarrow Y$ and $Z_2 \leftarrow X \leftrightarrow Y$. The entropy of Z_1 and Z_2 not related to Y should be minimized. We recover the IB objectives (Alemi et al., 2017): $\text{IB}_{\beta_{ib}}(Z_1, Z_2) \equiv \text{IB}_{\beta_{ib}}(Z_1) + \text{IB}_{\beta_{ib}}(Z_2) = \frac{1}{\beta_{ib}}[I(X;Z_1) + I(X;Z_2)] - [I(Y;Z_1) + I(Y;Z_2)]$. Let's additionally consider $I(Z_1;Z_2)$ that measures the shared information between both features. We minimize it to respect the *minimal* criterion of (Fischer, 2020).

$$\begin{aligned}&\text{IBMI}_{\beta_{ib},\delta_{mi}}(Z_1, Z_2) \equiv \text{IB}_{\beta_{ib}}(Z_1) + \text{IB}_{\beta_{ib}}(Z_2) + \delta_{mi} I(Z_1;Z_2)\\ &= \frac{1}{\beta_{ib}} \underbrace{[I(X;Z_1) + I(X;Z_2)]}_{\text{Compression}} - \underbrace{[I(Y;Z_1) + I(Y;Z_2)]}_{\text{Relevancy}} + \delta_{mi} \underbrace{I(Z_1;Z_2)}_{\text{Redundancy}}.\end{aligned}$$

The information from X is compressed into two independent & disentangled views Z_1 and Z_2 that both try to predict Y, in a relevancy-redundancy trade-off depending on β_{ib} and δ_{mi}.

2.1.2 Conditional mutual information formulation

IBMI ignores Y in the regularization terms which causes loss of necessary information: it is impossible to have Z_1 and Z_2 fully independent while training them to predict the same Y. As shown on Figure 4, redundancy regions (blue horizontal stripes) overlap with relevancy regions (red stripes). What the MNI really entails are these conditional equalities:

$$I(X;Z_1|Y) = I(X;Z_2|Y) = I(Z_1;Z_2|Y) = 0.$$

Mutual information being non-negative, we transform them into the following objective:

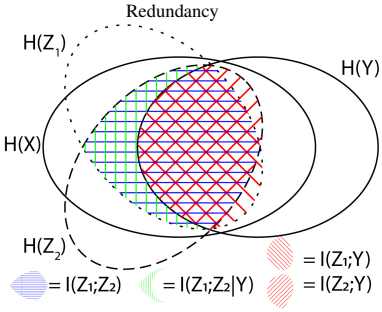

Figure 4: **Venn Information Diagram**. DICE minimizes conditional redundancy $I(Z_1;Z_2|Y)$ (green vertical stripes) with no overlap with relevancy (red stripes).

$$\begin{aligned}&\text{DICE}_{\beta_{ceb},\delta_{cmi}}(Z_1, Z_2) \equiv \text{CEB}_{\beta_{ceb}}(Z_1) + \text{CEB}_{\beta_{ceb}}(Z_2) + \delta_{cmi} I(Z_1;Z_2|Y)\\ &= \frac{1}{\beta_{ceb}} \underbrace{[I(X;Z_1|Y) + I(X;Z_2|Y)]}_{\text{Conditional Compression}} - \underbrace{[I(Y;Z_1) + I(Y;Z_2)]}_{\text{Relevancy}} + \delta_{cmi} \underbrace{I(Z_1;Z_2|Y)}_{\text{Conditional Redundancy}}.\end{aligned} \tag{1}$$

Analyze DICE only minimizes irrelevant information to the task. To specifically remove the spurious correlations between Z_1 and Z_2, we directly target the *conditional* redundancy. It encourages diversity by forcing members to have uncorrelated bias without affecting their individual precision as it protects information related to the target class. Useless information from X to predict Y should not be simultaneously in Z_1 and Z_2 as it would correlate bias causing double-faults. Note that reducing $I(Z_1, X|Y)$ and $I(Z_2, X|Y)$, the information Z_1 and Z_2 share with X given Y, also indirectly reduces $I(Z_1, Z_2|Y)$, but less efficiently. DICE benefits from both components (see Appendix E.7).

2.2 Transforming DICE into a tractable loss

2.2.1 Variational approximation of conditional entropy bottleneck

We leverage Markov assumptions in $Z \leftarrow X \leftrightarrow Y$ and empirically estimate on the classification training dataset D consisting of N i.i.d. data points $D = \{x^n, y^n\}_{n=1}^N, y^n \in \{1, \ldots, K\}$. Following (Fischer, 2020), $CEB_{\beta_{ceb}}(Z) = \frac{1}{\beta_{ceb}} I(X; Z|Y) - I(Y; Z)$ is variationally upper bounded by:

$$\text{VCEB}_{\beta_{ceb}}(\theta = \{e, b, c\}) = \frac{1}{N} \sum_{n=1}^{N} \{ \frac{1}{\beta_{ceb}} D_{\text{KL}}(e(z|x^n) \| b(z|y^n)) - \mathbb{E}_\epsilon \left[\log c(y^n | e(x^n, \epsilon)) \right] \}. \quad (2)$$

See detailed explanation in Appendix E.4. $e(z|x)$ is the true features distribution generated by the encoder, $c(y|z)$ is a variational approximation of $p(y|z)$ by the classifier, and $b(z|y)$ is a variational approximation of $p(z|y)$ by the backward encoder, all with parameters to be learned. We parametrize our distributions with gaussians: it enables (1) the exact computation of the KL divergence (see Appendix E.3) and (2) the reparameterization trick that gives a valid sample from the encoding distribution to the classifier, $z = e(x, \epsilon) = e^\mu(x) + \epsilon e^\sigma(x)$ where $\epsilon \sim N(0, 1)$. The **only advantage of VCEB versus VIB** (Alemi et al., 2017) is the **class conditional** $b(z|y)$ versus unconditional $b(z)$ variational marginal: VCEB protects class information while constraining the information shared between X and Z. We apply this loss on each individual member, as done in step 1 from Figure 3.

2.2.2 Neural conditional mutual information estimation

Theoretical problem We now focus on estimating $I(Z_1; Z_2|Y)$, with no such Markov properties. Despite being a pivotal measure, mutual information estimation historically relied on nearest neighbors (Singh et al., 2003; Kraskov et al., 2004; Gao et al., 2018) or density kernels (Kandasamy et al., 2015) and do not scale well in high dimensions. We benefit from recent advances in mutual information neural estimation (Belghazi et al., 2018), built on optimizing Donsker & Varadhan (1975) dual representations of the KL divergence. See (Poole et al., 2019; Song & Ermon, 2020) for a review. (Mukherjee et al., 2019; Molavipour et al., 2020) extended this formulation for CMI estimation.

$$\begin{aligned} I(Z_1; Z_2|Y) &= D_{\text{KL}}(P(Z_1, Z_2, Y) \| P(Z_1, Y) p(Z_2|Y)) \\ &= \sup_f \mathbb{E}_{x \sim p(z_1, z_2, y)}[f(x)] - \log \left(\mathbb{E}_{x \sim p(z_1, y) p(z_2|y)}[\exp(f(x))] \right) \\ &= \mathbb{E}_{x \sim p(z_1, z_2, y)}[f^*(x)] - \log \left(\mathbb{E}_{x \sim p(z_1, y) p(z_2|y)}[\exp(f^*(x))] \right), \end{aligned}$$

where f^* computes the point-wise log-likelihood ratio, i.e $f^*(z_1, z_2, y) = \frac{p(z_1, z_2, y)}{p(z_1, y) p(z_2|y)}$.

Empirical neural estimation We estimate conditional mutual information (1) using the empirical data distribution and (2) substituting the optimal f^* by the output of a trained neural-based classifier w. Let $\mathcal{B}_j$ and $\mathcal{B}_p$ be two random batches sampled respectively from the observed joint distribution $p(z_1, z_2, y) = p(e_1(z|x), e_2(z|x), y)$ and the product distribution $p(z_1, y)p(z_2|y) = p(e_1(z|x), y)p(z_2|y)$. To sample from $p(z_1, y)p(z_2|y)$, we select the features extracted by the two networks from two different inputs that share the same class, from the same batch when possible, or from previous batches otherwise. We train w, a multi-layer network with sigmoid as final activation, on the binary task of distinguishing these two distributions with the standard cross-entropy loss:

$$\mathcal{L}_{ce}(w) = -\frac{1}{|\mathcal{B}_j| + |\mathcal{B}_p|} \left[\sum_{(z_1, z_2, y) \in \mathcal{B}_j} \log w(z_1, z_2, y) + \sum_{(z_1, z_2', y) \in \mathcal{B}_p} \log(1 - w(z_1, z_2', y)) \right]. \quad (3)$$

At the end of training, if w is calibrated (see Appendix B.4) and by the "Universal Function Approximation Theorem" (Cybenko, 1989), w will be close to the optimal w^* that satisfies:

$$f^*(z_1, z_2, y) = \frac{w^*(z_1, z_2, y)}{1 - w^*(z_1, z_2, y)} \approx \frac{w(z_1, z_2, y)}{1 - w(z_1, z_2, y)} = f(z_1, z_2, y).$$

Therefore a consistent estimate of the CMI (see theorem 1 of Mukherjee et al. (2019)) is:

$$\hat{\mathcal{I}}_{DV}^{CMI} = \frac{1}{|\mathcal{B}_j|} \sum_{(z_1,z_2,y)\in\mathcal{B}_j} \underbrace{\log f(z_1,z_2,y)}_{\text{Diversity}} - \log\left(\frac{1}{|\mathcal{B}_p|} \sum_{(z_1,z_2',y)\in\mathcal{B}_p} \underbrace{f(z_1,z_2',y)}_{\text{Fake correlations}}\right).$$

Intuition By training our members to minimize $\hat{\mathcal{I}}_{DV}^{CMI}$, we force triples from the *true* joint distribution to become **indistinguishable** from triples from the *false* product distribution. Let's imagine that two features are conditionally correlated, some spurious information is shared between features only when they are from the same input and not from two inputs (from the same class). Therefore the joint and the product distribution are easily distinguishable by the discriminator. The first adversarial component will force the extracted features to reduce the correlation, and increase diversity. The second term would create fake correlations between features from different inputs and hinder the discriminator training. As we seek increased diversity and not a precise CMI estimation, we get rid of this second term, which adds complexity and empirically destabilizes the training.

$$\hat{\mathcal{L}}_{DV}^{CMI}(e_1,e_2) = \frac{1}{|\mathcal{B}_j|} \sum_{(z_1,z_2,y)\in\mathcal{B}_j \sim p(e_1(z|x),e_2(z|x),y)} \log f(z_1,z_2,y). \tag{4}$$

Summary It gives proper differentiable signals for adversarial (Goodfellow et al., 2014; Arjovsky et al., 2017) conditional features distributions alignment, as shown in step 2 from Figure 3. Features are *class-conditionally* independent while sharing class information. Our global loss is therefore:

$$\mathcal{L}_{DICE}(\theta_1,\theta_2) = \text{VCEB}_{\beta_{ceb}}(\theta_1) + \text{VCEB}_{\beta_{ceb}}(\theta_2) + \delta_{cmi}\hat{\mathcal{L}}_{DV}^{CMI}(e_1,e_2). \tag{5}$$

We derive similar losses for IBMI and CEBMI in Appendix E.5 and provide a pseudo code in B.2.

2.3 Full procedure with M members

We expand our objective for an ensemble with $M > 2$ members. We only consider pairwise interactions for simplicity to keep quadratic rather than exponential growth in number of components and leave the analysis of higher order interactions, such as $I(Z_i; Z_j, Z_k|Y)$, for future work.

$$\text{DICE}_{\beta_{ceb},\delta_{cmi}}(Z_{1:M}) \equiv \sum_{m=1}^{M} \text{CEB}_{\beta_{ceb}}(Z_m) + \frac{\delta_{cmi}}{(M-1)} \sum_{m=1}^{M} \sum_{m'=m+1}^{M} I(Z_m; Z_{m'}|Y). \tag{6}$$

Justified by previous variational lower bounds and neural estimations, we jointly train $\theta_m = \{e_m, b_m, c_m\}, m \in \{1,\dots,M\}$ on $\mathcal{L}_{DICE}(\theta_{1:M})$ (see Appendix E.6) while training **adversarially** w on $\mathcal{L}_{ce}$. $\mathcal{B}_j$ (resp. $\mathcal{B}_p$) samples from the concatenation of $p(z_m, z_{m'}, y)$ (resp. $p(z_m, y)p(z_{m'}|y)$) where $m \neq m'$. We use the same discriminator w for $\binom{M}{2}$ estimates for simplicity.

3 Related work

To reduce the training cost of deep ensembles (Hansen & Salamon, 1990; Lakshminarayanan et al., 2016), Huang et al. (2017) collect **snapshots** on training trajectories. One stage end-to-end **co-distillation** (Song & Chai, 2018; Lan et al., 2018; Chen et al., 2019) share low-level features among members in *branch-based* ensemble while forcing each member to mimic a dynamic weighted combination of the predictions to increase individual accuracy. However both methods correlate errors among members, homogenize predictions and fail to fit the different modes of the data which overall reduce diversity. Beyond random initializations (Kolen & Pollack, 1991), some **implicitly** introduced stochasticity into the training, by providing *subsets of data* to learners with bagging (Breiman, 1996) or by backpropagating *subsets of gradients* (Lee et al., 2016): however, the reduction of training samples hurts performance for sufficiently complex models that overfit their training dataset (Nakkiran et al., 2019). Boosting with sequential training is not suitable for deep members (Lakshminarayanan et al., 2016). Some approaches applied different *data augmentations* (Dvornik et al., 2019; Stickland & Murray, 2020), used *different networks* or *hyper-parameters* (Singh et al., 2016; Ruiz & Verbeek, 2020; Yang & Soatto, 2020), but are not general-purpose and depend on specific engineering choices. Others **explicitly** encourage orthogonality of the *gradients* (Ross et al., 2019;

Kariyappa & Qureshi, 2019; Dabouei et al., 2020) or of the *predictions*, by *boosting* (Freund & Schapire, 1999; Margineantu & Dietterich) or with a *negative correlation* regularization (Shui et al., 2018), but they reduce members accuracy. As far as we know, adaptive diversity promoting (ADP) (Pang et al., 2019) is the unique approach that performs better than the independent baseline by decorrelating the non-maximal predictions. The limited success of these logits approaches suggests that we seek diversity in *features*. Empirically we found that the increase of (L_1, L_2, $-\cos$) distances between features (Kim et al., 2018b) reduce performance: they are not invariant to variables' translation. Simultaneously to our findings, Sinha et al. (2020) is somehow equivalent to our IBMI objective (see Appendix C.2) but without information bottleneck motivations for the diversity loss.

The uniqueness of **mutual information** (see Appendix E.2) as a distance measure between variables has been applied in countless machine learning projects, such as reinforcement learning (Kim et al., 2018a), metric learning (Kemertas et al., 2020), or evolutionary algorithms (Aguirre & Coello, 2004). Objectives are often a trade-off between (1) informativeness and (2) compression. In computer vision, unsupervised deep representation learning (Hjelm et al.; Oord et al., 2018; Tian et al., 2019; Bachman et al., 2019) maximizes correlation between features and inputs following Infomax (Linsker, 1988; Bell & Sejnowski, 1995), while discarding information not shared among different views (Bhardwaj et al., 2020), or penalizing predictability of one latent dimension given the others for disentanglement (Comon, 1994; Schmidhuber, 1992; Kingma & Welling, 2014; Higgins et al., 2016; Kim & Mnih, 2018; Blot et al., 2018). The ideal level of compression is **task dependent** (Soatto & Chiuso, 2014). As a selection criterion, features should not be redundant (Battiti, 1994; Peng et al., 2005) but relevant and complementary given the task (Novovičová et al., 2007; Brown, 2009; Brown et al., 2012). As a learning criteria, correlations between features and inputs are minimized according to the Information Bottleneck principle (Tishby et al., 2000; Alemi et al., 2017; Fischer, 2020; Kirsch et al., 2020; Saporta et al., 2019), while those between features and targets are maximized (LeCun et al., 2006; Qin & Kim, 2019). It forces the features to ignore task-irrelevant factors (Zhao et al., 2019), to reduce overfitting (Alemi et al., 2018) while protecting needed information (Tian et al., 2020). Fischer & Alemi (2020) concludes in the superiority of conditional distributions alignment to reach the MNI with CMI regularization (Steinke & Zakynthinou, 2020).

4 EXPERIMENTS

In this section, we present our experimental results on the *CIFAR-10* and *CIFAR-100* (Krizhevsky et al., 2009) classification datasets. We detail our implementation in Appendix B. We took most hyper-parameters from Chen et al. (2019). Hyper-parameters for adversarial training and information bottleneck were fine-tuned on a validation dataset made of 5% of the training dataset, see Appendix D.1. **Bold** highlights best score. We improve accuracy, calibration, uncertainty estimates in-domain and out-domain, out-of-distribution detection and co-distillation performances.

4.1 COMPARISON OF CLASSIFICATION ACCURACY

Table 1: **Ensemble classification accuracy** (Top-1, %) on *CIFAR-100* and *CIFAR-10*

Method			CIFAR-100									CIFAR-10	
Name	Components		ResNet-32				ResNet-110		WRN-28-2			ResNet-32	ResNet-110
	Div.	I.B.	3-branch	4-branch	5-branch	4-net	3-branch	4-branch	3-branch	4-branch	3-net	4-branch	3-branch
Ind.			76.28 ± 0.12	76.78 ± 0.19	77.24 ± 0.25	77.38 ± 0.12	80.54 ± 0.09	80.89 ± 0.31	78.83 ± 0.12	79.10 ± 0.08	80.01 ± 0.15	94.75 ± 0.08	95.62 ± 0.06
ONE (Lan et al., 2018)			75.17 ± 0.35	75.13 ± 0.25	75.25 ± 0.22	76.25 ± 0.32	78.97 ± 0.24	79.86 ± 0.25	78.38 ± 0.45	78.47 ± 0.32	77.53 ± 0.36	94.41 ± 0.05	95.25 ± 0.08
OKDDip (Chen et al., 2019)			75.37 ± 0.32	76.85 ± 0.25	76.95 ± 0.18	77.27 ± 0.31	79.07 ± 0.27	80.46 ± 0.35	79.01 ± 0.19	79.32 ± 0.17	80.02 ± 0.14	94.86 ± 0.08	95.21 ± 0.09
ADP (Pang et al., 2019)	Pred.		76.37 ± 0.11	77.21 ± 0.21	77.67 ± 0.25	77.51 ± 0.25	80.73 ± 0.38	81.40 ± 0.27	79.21 ± 0.19	79.71 ± 0.18	80.01 ± 0.17	94.92 ± 0.04	95.43 ± 0.12
IB		VIB	76.01 ± 0.12	76.93 ± 0.24	77.22 ± 0.19	77.72 ± 0.12	80.43 ± 0.34	81.12 ± 0.19	79.19 ± 0.35	79.15 ± 0.12	80.15 ± 0.13	94.76 ± 0.12	94.54 ± 0.07
CEB		VCEB	76.36 ± 0.06	76.98 ± 0.18	77.35 ± 0.14	77.64 ± 0.15	81.08 ± 0.12	81.17 ± 0.16	78.92 ± 0.08	79.20 ± 0.13	80.38 ± 0.18	94.93 ± 0.11	94.65 ± 0.05
IBMI (equation 9)	MI	VIB	76.68 ± 0.13	77.25 ± 0.13	77.77 ± 0.21	77.84 ± 0.12	81.34 ± 0.21	81.38 ± 0.08	79.33 ± 0.15	79.90 ± 0.10	80.22 ± 0.10	94.91 ± 0.14	95.68 ± 0.05
CEBMI (equation 10)	MI	VCEB	76.72 ± 0.08	77.30 ± 0.12	77.81 ± 0.10	77.82 ± 0.11	81.52 ± 0.11	81.55 ± 0.37	79.25 ± 0.15	79.98 ± 0.07	80.35 ± 0.15	94.94 ± 0.12	95.67 ± 0.06
DICE (equation 5)	CMI	VCEB	**76.89** ± 0.09	**77.51** ± 0.17	**78.08** ± 0.18	**77.92** ± 0.08	**81.67** ± 0.14	**81.93** ± 0.13	**79.59** ± 0.13	**80.05** ± 0.11	**80.55** ± 0.12	**95.01** ± 0.09	**95.74** ± 0.08

Table 1 reports the Top-1 classification accuracy averaged over 3 runs with standard deviation. DICE surpasses concurrent approaches (summarized in Appendix C), including the baseline Ind. that refers to the training of *independent* deterministic members without interactions between them, for ResNet and Wide-ResNet architectures, in network-based and even more in the branch-based setup. Branch-based (Song & Chai, 2018) ensemble is attractive as it reduces bias by gradient diffusion (shared layers receive various gradients), at only a slight cost in diversity which makes our approach

even more valuable. The more networks the better, but accuracy quickly reaches a plateau. The **saturation** worsens with co-distillation methods (ONE(Lan et al., 2018), OKDDip (Chen et al., 2019)) that often perform worst than Ind.: they sacrifice diversity for individual accuracy. On the contrary, diversity approaches more deeply benefit from additional members, as shown in Figure 1 and Table 7. We bring systematic improvement to the current state-of-the-art ADP (Pang et al., 2019): e.g. on CIFAR-100, $\{+0.52, +0.30, +0.41\}$ for {3,4,5}-branches ResNet-32, $\{+0.94, +0.53\}$ for {3,4}-branches ResNet-110 and finally $+0.34$ for 3-nets WRN-28-2.

4.2 Ablation Study

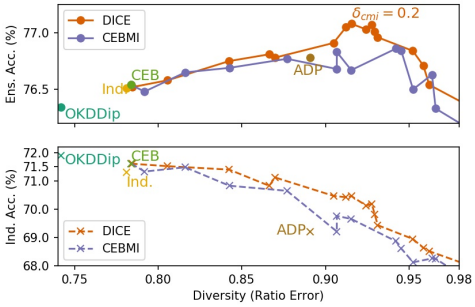

Figure 5: **Trade-off** between members' accuracies and ensemble diversity for different strategies. DICE and CEBMI are learned with different δ_{cmi}. Solid/dashed lines give the ensemble/member validation accuracies.

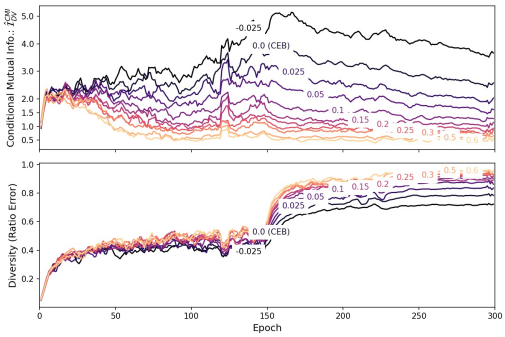

Figure 6: **Impact of the diversity coefficient** δ_{cmi} in DICE on the training dynamics on validation: CMI is negatively correlated with diversity.

We study the 4-branches ResNet-32 on CIFAR-100 in following experiments. We ablate the two components of DICE: (1) deterministic, with VIB (Alemi et al., 2017) or VCEB (Fischer, 2020) (2) no adversarial loss, with MI or CMI loss. We measure diversity by the ratio-error (Aksela, 2003), $r = \frac{N_{\text{single}}}{N_{\text{shared}}}$, which computes the ratio between the number of single errors N_{single} and of shared errors N_{shared}. A higher average over the $\binom{M}{2}$ pairs means higher diversity as members are less likely to err on the same inputs. Our analysis remains valid for non-pairwise measures, see Appendix A.5.

First, CEB has higher diversity than the deterministic Ind. (see Figure 5): reducing what features share with the input automatically reduces what they share between them (see Appendix E.7). It also benefits from additional randomness brought by the reparameterization trick. Moreover CEB has lower standard deviation than IB (Fischer, 2020) and is less dependent on the hyper-parameter values: β_{ib} can hinder the learnability (Wu et al., 2019b). ADP reaches higher diversity than CEB but reduces individual accuracy. Finally, DICE performs best. While CEBMI reaches same diversity, it is more damaging to individual accuracies as it forces features to lose some categorical information.

In Figure 6, we see that starting from random initializations, diversity begins small and increases quickly. Once the ramp up is over at step 80, our approach minimizes the estimated CMI among branches and therefore increases diversity compared to independent learners. The effect is correlated with the diversity coefficient δ_{cmi}: a negative value (-0.025) has the opposite effect while a high value (0.6) creates too much diversity. We chose $\delta_{cmi} = 0.2$ that induces an **optimal trade-off between ensemble diversity and individual accuracies** on the validation dataset (see Figure 8).

4.3 In-domain and out-domain uncertainty estimation and calibration

Table 2 compares in-domain (CIFAR-100) and out-domain (CIFAR-100-C, with corrupted images at five different levels) performance before and after temperature scaling (procedure in Appendix D.2): indeed, Ashukha et al. (2020) showed that "comparison of [...] ensembling methods without temperature scaling might not provide a fair ranking". We recover that ensemble and IB improve robustness. ADP performs well on out-domain because its entropy maximisation loss causes a low internal T (≈ 0.64). Members's disagreement in DICE increases uncertainty estimation and calibration (see Figure 8 in Appendix A.4), notably for harder examples from CIFAR-100-C.

Table 2: **Uncertainty estimation** (NLL, BS) and **calibration** (ECE, TACE) performances.

			Temp. scaling	1-net	Ind.	OKDDip-E	ADP	IB	CEB	IBMI	CEBMI	DICE
Dataset	Metrics	Unit	Temp. avg value	1.49	1.31	1.33	0.64	1.21	1.24	1.17	1.19	1.11
CIFAR-100	Acc. ↑	%	-	71.28	76.71	76.85	77.21	76.93	76.98	77.25	77.30	**77.51**
	NLL ↓	10^{-1}	✗/✓	11.56/10.38	8.55/8.10	8.38/8.13	10.85/8.51	8.37/8.12	8.37/8.11	8.27/8.09	8.25/8.05	**8.06/7.98**
	BS ↓	10^{-3}	✗/✓	4.12/3.92	3.35/3.24	3.28/3.19	3.79/3.27	3.25/3.20	3.25/3.19	3.21/3.17	3.23/3.18	**3.15/3.12**
	ECE ↓	10^{-2}	✗/✓	10.47/1.83	7.45/**1.60**	6.67/1.73	21.14/2.99	5.32/2.17	5.76/2.07	5.15/1.97	5.46/2.02	**4.05**/2.59
	TACE ↓	10^{-3}	✗/✓	2.42/1.98	1.86/1.78	1.81/1.74	4.53/1.79	1.58/**1.68**	1.67/1.69	1.60/1.75	1.65/1.72	**1.46**/1.70
CIFAR-100c	Acc. ↑	%	-	43.67	48.23	47.19	49.01	48.39	48.45	48.71	48.74	**49.34**
	NLL ↓	10^{-1}	✗/✓	32.7/25.9	27.8/23.9	29.7/25.1	**23.1**/24.6	26.8/24.3	27.3/24.8	26.21/24.38	25.78/24.76	25.6/**22.77**
	BS ↓	10^{-3}	✗/✓	8.10/7.31	7.32/6.86	7.56/7.04	**6.69**/6.91	7.19/6.88	7.26/6.88	7.09/6.83	7.08/6.95	6.93/**6.63**
	ECE ↓	10^{-2}	✗/✓	27.9/13.8	23.2/14.0	24.83/15.29	**7.59**/16.27	21.21/14.68	21.9/14.55	20.14/14.54	18.04/14.91	18.08/**13.47**
	TACE ↓	10^{-3}	✗/✓	4.94/3.60	4.42/3.51	4.62/3.75	3.71/3.57	4.18/3.56	4.26/3.46	4.01/3.43	3.72/3.41	**3.69/3.21**

4.4 OUT-OF-DISTRIBUTION DETECTION

To measure the ability of our ensemble to distinguish in- and out-of-distribution (OOD) images, we consider other datasets at test time following (Hendrycks & Gimpel, 2016) (see Appendix D.3). The confidence score is estimated with the maximum softmax value: the confidence for OOD images should ideally be lower than for CIFAR-100 test images.

(a) Ind.+TS (74.0) (b) DICE+TS (75.2) (c) DICE$\times w$ (**77.2**)

Figure 7: **Confidence estimates** separate images from CIFAR-100 and OOD images from TinyImageNet (crop) for different strategies (AUROC ↑). DICE$\times w$ uses the discriminator to scale its confidence: w's predictions behave like an "input-dependant temperature".

Results are detailed in Table 5 (resp. Table 6), before (resp. after) temperature scaling (TS), and are illustrated in Figure 7. DICE beats Ind. and CEB. We suspected that features were more correlated for OOD images as they would share artifacts. DICE$\times w$ multiplies the classification logits by $1 - \frac{2}{(M-1)M} \sum_{m=1}^{M} \sum_{m'=m+1}^{M} w(z_m, z_{m'}, \hat{y})$, with predicted $\hat{y}$ (as the true y is not available at test time). It performs better than traditional TS, at the cost of additional operations. It shows the ability of w to detect conditional residual correlations, adversarially deleted only when found in training.

4.5 DIVERSE TEACHER FOR IMPROVED CO-DISTILLATION

Table 3: **Individual accuracy for branch-based co-distillation** on CIFAR-100

		1-net	Ind.	ONE (Lan et al., 2018)	OKDDip (Chen et al., 2019)			PCL (Wu & Gong, 2020)	OKDDip+CEB			OKDDip+DICE		
T Distillation		-	-	3	3	2.5	2	3	3	2.5	2	3	2.5	2
ResNet-32	3-branch	$71.28_{\pm 0.11}$	$72.15_{\pm 0.08}$	$73.32_{\pm 0.22}$	$73.90_{\pm 0.15}$	$74.01_{\pm 0.08}$	$74.12_{\pm 0.12}$	$74.14_{\pm 0.16}$	$73.95_{\pm 0.09}$	$74.10_{\pm 0.09}$	$74.08_{\pm 0.11}$	$74.14_{\pm 0.11}$	$74.28_{\pm 0.12}$	$\mathbf{74.56}_{\pm 0.18}$
	4-branch			$73.42_{\pm 0.18}$	$74.40_{\pm 0.13}$	$74.42_{\pm 0.11}$	$74.31_{\pm 0.09}$	-	$74.01_{\pm 0.11}$	$74.15_{\pm 0.21}$	$74.61_{\pm 0.17}$	$74.22_{\pm 0.08}$	$74.43_{\pm 0.18}$	$\mathbf{74.95}_{\pm 0.15}$

The inference time in network-based ensemble grows linearly with M, limiting applicability. Sharing early-features is one solution. We experiment another solution by using only the M-th branch at test time. We combine DICE with OKDDip (Chen et al., 2019): the M-th branch learns to mimic the soft predictions from the M-1 first branches, among which we enforce diversity. Our teacher has lower internal temperature (as shown in experiment 4.3): DICE performs best when soft predictions are generated with lower co-distillation T. We improve sota by $\{+0.42, +0.53\}$ for {3,4}-branches.

5 CONCLUSION

In this paper we addressed the task of improving deep ensembles by increased diversity. Motivated by information theoretic arguments, we derive a novel adversarial loss, based on conditional mutual information estimation. We highlight the trade-off between individual accuracy and ensemble diversity and surpass state-of-the-art performances on standard image classification benchmarks. As the success of many real-world systems in production depends on the robustness of their deep learning systems, we hope to pave the way towards general-purpose methods for diverse & better ensemble.

REFERENCES

Arturo Hernández Aguirre and Carlos A Coello Coello. Mutual information-based fitness functions for evolutionary circuit synthesis. In *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753)*, volume 2, pp. 1309–1316. IEEE, 2004.

Matti Aksela. Comparison of classifier selection methods for improving committee performance. In *International Workshop on Multiple Classifier Systems*, pp. 84–93. Springer, 2003.

Alex Alemi, Ian Fischer, Josh Dillon, and Kevin Murphy. Deep variational information bottleneck. In *ICLR*, 2017. URL `https://arxiv.org/abs/1612.00410`.

Alexander A Alemi, Ian Fischer, and Joshua V Dillon. Uncertainty in the variational information bottleneck. *arXiv preprint arXiv:1807.00906*, 2018.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 214–223, 2017.

Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. *arXiv preprint arXiv:2002.06470*, 2020.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pp. 15535–15545, 2019.

Roberto Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on neural networks*, 5(4):537–550, 1994.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International Conference on Machine Learning*, pp. 531–540, 2018.

Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

Hedi Ben-Younes, Remi Cadene, Nicolas Thome, and Matthieu Cord. Block: Bilinear superdiagonal fusion for visual question answering and visual relationship detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 8102–8109, 2019.

Sangnie Bhardwaj, Ian Fischer, Johannes Ballé, and Troy Chinen. An unsupervised information-theoretic perceptual quality metric. *arXiv preprint arXiv:2006.06752*, 2020.

Michael Blot, Thomas Robert, Nicolas Thome, and Matthieu Cord. Shade: Information-based regularization for deep learning. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 813–817. IEEE, 2018.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, pp. 1613–1622, 2015.

Nicholas A Bowman. How much diversity is enough? the curvilinear relationship between college diversity interactions and first-year student outcomes. *Research in Higher Education*, 54(8):874–894, 2013.

Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

Gavin Brown. A new perspective for information theoretic feature selection. In *Artificial intelligence and statistics*, pp. 49–56, 2009.

Gavin Brown, Jeremy Wyatt, and Ping Sun. Between two extremes: Examining decompositions of the ensemble objective function. In *International workshop on multiple classifier systems*, pp. 296–305. Springer, 2005a.

Gavin Brown, Jeremy L Wyatt, and Peter Tiňo. Managing diversity in regression ensembles. *Journal of machine learning research*, 6(Sep):1621–1650, 2005b.

Gavin Brown, Adam Pocock, Ming-Jie Zhao, and Mikel Luján. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *The journal of machine learning research*, 13(1):27–66, 2012.

Changrui Chen, Xin Sun, Yang Hua, Junyu Dong, and Hongwei Xv. Learning deep relations to promote saliency detection.

Defang Chen, Jian-Ping Mei, Can Wang, Yan Feng, and Chun Chen. Online knowledge distillation with diverse peers. 2019.

Nadezhda Chirkova, Ekaterina Lobacheva, and Dmitry Vetrov. Deep ensembles on a fixed memory budget: One wide network or several thinner ones? *arXiv preprint arXiv:2005.07292*, 2020.

Inseop Chung, SeongUk Park, Jangho Kim, and Nojun Kwak. Feature-map-level online adversarial knowledge distillation. *arXiv preprint arXiv:2002.01775*, 2020.

Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

Charles Corbière, Nicolas THOME, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure detection by learning model confidence. In *Advances in Neural Information Processing Systems*, pp. 2898–2909, 2019.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Ali Dabouei, Sobhan Soleymani, Fariborz Taherkhani, Jeremy Dawson, and Nasser M. Nasrabadi. Exploiting joint robustness to adversarial perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Terrance DeVries and Graham W Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018.

Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pp. 1–15. Springer, 2000.

Monroe D Donsker and SR Srinivasa Varadhan. Asymptotic evaluation of certain markov process expectations for large time, i. *Communications on Pure and Applied Mathematics*, 28(1):1–47, 1975.

Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Diversity with cooperation: Ensemble methods for few-shot classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3723–3731, 2019.

Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. CRC press, 1994.

Ian Fischer. The conditional entropy bottleneck. *arXiv preprint arXiv:2002.05379*, 2020.

Ian Fischer and Alexander A Alemi. Ceb improves model robustness. *arXiv preprint arXiv:2002.05380*, 2020.

Yoav Freund and Robert Schapire. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059, 2016.

Weihao Gao, Sewoong Oh, and Pramod Viswanath. Demystifying fixed k-nearest neighbor information estimators. *IEEE Transactions on Information Theory*, 64(8):5629–5661, 2018.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Advances in Neural Information Processing Systems*, pp. 8789–8798, 2018.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo.

Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001, 1990.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, and A Object Detection Baselines. Deep residual learning for image recognition. *context*, 51(30.0):53–3.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=HJz6tiCqYm`.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q. Weinberger. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*, 2017.

Kirthevasan Kandasamy, Akshay Krishnamurthy, Barnabas Poczos, Larry Wasserman, et al. Nonparametric von mises estimators for entropies, divergences and mutual informations. In *Advances in Neural Information Processing Systems*, pp. 397–405, 2015.

Sanjay Kariyappa and Moinuddin K. Qureshi. Improving adversarial robustness of ensembles with diversity training. *arXiv preprint arXiv:1901.09981*, 2019.

Mete Kemertas, Leila Pishdad, Konstantinos G. Derpanis, and Afsaneh Fazly. Rankmi: A mutual information maximizing ranking loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Hyoungseok Kim, Jaekyeom Kim, Yeonwoo Jeong, Sergey Levine, and Hyun Oh Song. Emi: Exploration with mutual information. *arXiv preprint arXiv:1810.01176*, 2018a.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

Jangho Kim, Minsung Hyun, Inseop Chung, and Nojun Kwak. Feature fusion for online mutual knowledge distillation. *arXiv preprint arXiv:1904.09058*, 2019.

Wonsik Kim, Bhavya Goyal, Kunal Chawla, Jungmin Lee, and Keunjoo Kwon. Attention-based ensemble for deep metric learning. *arXiv preprint arXiv:1804.00382*, 2018b.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

Andreas Kirsch, Clare Lyle, and Yarin Gal. Unpacking information bottlenecks: Unifying information-theoretic objectives in deep learning. *arXiv preprint arXiv:2003.12537*, 2020.

Ron Kohavi, David H Wolpert, et al. Bias plus variance decomposition for zero-one loss functions. 1996.

John F Kolen and Jordan B Pollack. Back propagation is sensitive to initial conditions. In *Advances in neural information processing systems*, pp. 860–867, 1991.

Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in neural information processing systems*, pp. 231–238, 1995.

Solomon Kullback. *Information Theory and Statistics*. Wiley, New York, 1959.

Ludmila I Kuncheva. Methods and algorithms.

Ludmila I Kuncheva and Christopher J Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51(2):181–207, 2003.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*, 2016.

Xu Lan, Xiatian Zhu, and Shaogang Gong. Knowledge distillation by on-the-fly native ensemble. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 7528–7538, 2018.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. 2006.

Stefan Lee, Senthil Purushwalkam, Michael Cogswell, David Crandall, and Dhruv Batra. Why m heads are better than one: Training a diverse ensemble of deep networks. *arXiv preprint arXiv:1511.06314*, 2015.

Stefan Lee, Senthil Purushwalkam, Michael Cogswell, Viresh Ranjan, David Crandall, and Dhruv Batra. Stochastic multiple choice learning for training diverse deep ensembles. *arXiv preprint arXiv:1606.07839*, 2016.

Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

Ralph Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

Cheng-Lin Liu and Masaki Nakagawa. Evaluation of prototype learning algorithms for nearest-neighbor classifier in application to handwritten character recognition. *Pattern Recognition*, 34 (3):601–615, 2001.

Yong Liu and Xin Yao. Ensemble learning via negative correlation. *Neural networks*, 12(10):1399–1404, 1999a.

Yong Liu and Xin Yao. Simultaneous training of negatively correlated neural networks in an ensemble. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(6):716–725, 1999b.

Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems*, pp. 13153–13164, 2019.

Dragos D Margineantu and Thomas G Dietterich. Pruning adaptive boosting. Citeseer.

Sina Molavipour, Germán Bassi, and Mikael Skoglund. On neural estimators for conditional mutual information using nearest neighbors sampling. *arXiv preprint arXiv:2006.07225*, 2020.

Sudipto Mukherjee, Himanshu Asnani, and Sreeram Kannan. Ccmi: Classifier based conditional mutual information estimation. *arXiv preprint arXiv:1906.01824*, 2019.

Ryan Muldoon. *Social contract theory for a diverse world: Beyond tolerance*. Taylor & Francis, 2016.

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

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *arXiv preprint arXiv:1912.02292*, 2019.

Nils J. Nilsson. Learning machines: Foundations of trainable pattern-classifying systems. 1965.

Jeremy Nixon, Michael W Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning.

Jana Novovičová, Petr Somol, Michal Haindl, and Pavel Pudil. Conditional mutual information based feature selection for classification task. In *Iberoamerican Congress on Pattern Recognition*, pp. 417–426. Springer, 2007.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, pp. 271–279, 2016.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. *arXiv preprint arXiv:1901.08846*, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

Ben Poole, Sherjil Ozair, Aaron van den Oord, Alexander A Alemi, and George Tucker. On variational bounds of mutual information. *arXiv preprint arXiv:1905.06922*, 2019.

Zhenyue Qin and Dongwoo Kim. Rethinking softmax with cross-entropy: Neural network classifier as mutual information estimator. *arXiv preprint arXiv:1911.10688*, 2019.

Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, volume 6. International Conference on Representation Learning, 2018.

Lior Rokach. Ensemble-based classifiers. *Artificial intelligence review*, 33(1-2):1–39, 2010.

Andrew Slavin Ross, Weiwei Pan, Leo Anthony Celi, and Finale Doshi-Velez. Ensembles of locally independent prediction models. *arXiv preprint arXiv:1911.01291*, 2019.

Adrià Ruiz and Jakob Verbeek. Distilled hierarchical neural ensembles with adaptive inference cost. *arXiv preprint arXiv:2003.01474*, 2020.

Antoine Saporta, Yifu Chen, Michael Blot, and Matthieu Cord. Reve: Regularizing deep learning with variational entropy bound. *arXiv preprint arXiv:1910.06816*, 2019.

Jürgen Schmidhuber. Learning factorial codes by predictability minimization. *Neural computation*, 4(6):863–879, 1992.

Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Changjian Shui, Azadeh Sadat Mozafari, Jonathan Marek, Ihsen Hedhli, and Christian Gagné. Diversity regularization in deep ensembles. *arXiv preprint arXiv:1802.07881*, 2018.

Demetrio Sierra-Mercado and Gabriel Lázaro-Muñoz. Enhance diversity among researchers to promote participant trust in precision medicine research. *The American Journal of Bioethics*, 18(4): 44–46, 2018.

Harshinder Singh, Neeraj Misra, Vladimir Hnizdo, Adam Fedorowicz, and Eugene Demchuk. Nearest neighbor estimates of entropy. *American journal of mathematical and management sciences*, 23(3-4):301–321, 2003.

Saurabh Singh, Derek Hoiem, and David Forsyth. Swapout: Learning an ensemble of deep architectures. In *Advances in neural information processing systems*, pp. 28–36, 2016.

Samarth Sinha, Homanga Bharadhwaj, Anirudh Goyal, Hugo Larochelle, Animesh Garg, and Florian Shkurti. Dibs: Diversity inducing information bottleneck in model ensembles. *arXiv preprint arXiv:2003.04514*, 2020.

Stefano Soatto and Alessandro Chiuso. Visual representations: Defining properties and deep approximations. *arXiv preprint arXiv:1411.7676*, 2014.

Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. Amortised map inference for image super-resolution. *arXiv preprint arXiv:1610.04490*, 2016.

Guocong Song and Wei Chai. Collaborative learning for deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 1832–1841, 2018.

Jiaming Song and Stefano Ermon. Understanding the limitations of variational mutual information estimators. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=B1x62TNtDS`.

Thomas Steinke and Lydia Zakynthinou. Reasoning about generalization via conditional mutual information. *arXiv preprint arXiv:2001.09122*, 2020.

Asa Cooper Stickland and Iain Murray. Diverse ensembles improve calibration. *arXiv preprint arXiv:2007.04206*, 2020.

Talia H Swartz, Ann-Gel S Palermo, Sandra K Masur, and Judith A Aberg. The science and value of diversity: Closing the gaps in our understanding of inclusion and diversity. *The Journal of infectious diseases*, 220(Supplement_2):S33–S41, 2019.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Naonori Ueda and Ryohei Nakano. Generalization error of ensemble estimators. In *Proceedings of International Conference on Neural Networks (ICNN'96)*, volume 1, pp. 90–95. IEEE, 1996.

Bogdan Vasilescu, Daryl Posnett, Baishakhi Ray, Mark GJ van den Brand, Alexander Serebrenik, Premkumar Devanbu, and Vladimir Filkov. Gender and tenure diversity in github teams. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pp. 3789–3798, 2015.

David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

A Wu, S Nowozin, E Meeds, RE Turner, JM Hernández-Lobato, and AL Gaunt. Deterministic variational inference for robust bayesian neural networks. In *7th International Conference on Learning Representations, ICLR 2019*, 2019a.

Guile Wu and Shaogang Gong. Peer collaborative learning for online knowledge distillation. *arXiv preprint arXiv:2006.04147*, 2020.

Tailin Wu and Ian Fischer. Phase transitions for the information bottleneck in representation learning. *arXiv preprint arXiv:2001.01878*, 2020.

Tailin Wu, Ian Fischer, Isaac L Chuang, and Max Tegmark. Learnability for the information bottleneck. *Entropy*, 21(10):924, 2019b.

Pingmei Xu, Krista A Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R. Kulkarni, and Jianxiong Xiao. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*, 2015.

Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4085–4095, 2020.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

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

Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical stochastic gradient mcmc for bayesian deep learning. In *International Conference on Learning Representations*, 2019.

Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Han Zhao, Amanda Coston, Tameem Adel, and Geoffrey J. Gordon. Conditional learning of fair representations. *arXiv preprint arXiv:1910.07162*, 2019.

Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. Ensembling neural networks: many could be better than all. *Artificial intelligence*, 137(1-2):239–263, 2002.

Appendices

The structure of the Appendix is as follows. Appendix A shows additional experiments. Then in Appendix B, we give a detailed description of our implementation to facilitate reproduction. In Appendix C, we summarize the concurrent approaches (see Table 8). In Appendix D, we describe the datasets and the metrics used in our experiments. Appendix E clarifies certain theoretical formulations. We conclude with a sociological analogy in Appendix F.

A Additional Experiments

A.1 Comparisons with co-distillation and snapshot-based approaches

Table 4: **Ensemble Accuracy** on different setups. Concurrent approaches accuracies are those reported in recent papers. DICE outperforms co-distillation and snapshot-based ensembles collected on the training trajectory, which fail to capture the different modes of the data (Ashukha et al., 2020).

Architecture				Concurrent Approach			Baseline	Ours
Dataset	Network	Structure	Ens. Size	Name	Acc.	According to	Ind. Acc.	DICE Acc.
CIFAR-100	ResNet-32	Branches	3	CL-ILR (Song & Chai, 2018)	72.99	(Chen et al., 2019)	76.28	**76.89**
		Nets	3	DML (Zhang et al., 2018)	76.11	(Chung et al., 2020)	76.45	**76.98**
				AFD (Chung et al., 2020)	76.64	(Chung et al., 2020)		
	ResNet-110	Branches	3	FFL (Kim et al., 2019)	78.22	(Wu & Gong, 2020)	80.54	**81.67**
				PCL-E (Wu & Gong, 2020)	80.51	(Wu & Gong, 2020)		
			4	CL-ILR (Song & Chai, 2018)	79.81	(Chen et al., 2019)	80.89	**81.93**
		Nets	5	SWAG (Maddox et al., 2019)	77.69	(Ashukha et al., 2020)	81.7 (Ashukha et al., 2020)	**81.82**
				Cyclic SGLD (Zhang et al., 2019)	74.27	(Ashukha et al., 2020)		
				Fast Geometric Ens (Garipov et al., 2018)	78.78	(Ashukha et al., 2020)		
				Variational Inf. (FFG) (Wu et al., 2019a)	77.59	(Ashukha et al., 2020)		
				KFAC-Laplace (Ritter et al., 2018)	77.13	(Ashukha et al., 2020)		
				Snapshot Ensembles (Huang et al., 2017)	77.17	(Ashukha et al., 2020)		
	WRN-28-2	Nets	3	DML (Zhang et al., 2018)	79.41	(Chung et al., 2020)	80.01	**80.55**
				AFD (Chung et al., 2020)	79.78	(Chung et al., 2020)		
CIFAR-10	ResNet-110	Branches	3	FFL (Kim et al., 2019)	95.01	(Wu & Gong, 2020)	95.62	**95.74**
				PCL-E (Wu & Gong, 2020)	95.58	(Wu & Gong, 2020)		

A.2 Out-of-distribution Detection

Table 5 summarizes our OOD experiments in the 4-branches ResNet-32 setup. We recover that IB improves OOD detection (Alemi et al., 2018). Moreover, we empirically validate our intuition: features for in-distribution images are in average less predictive from each other compared to pairs of features of OOD images. w can perform alone as a OOD-detector, but is best used in complement to DICE: in DICE$\times w$, logits from DICE are multiplied by the sigmoid output of w averaged over all pairs. Table 6 shows that temperature scaling improves all approaches but do not modify the ranking. Finally, DICE$\times w$ without temperature scaling is better than DICE with temperature scaling.

Table 5: **Out-of-distribution** performances **before** temperature scaling.

Dataset		FPR (95 % TPR) ↓					Detection ↓					AUROC ↑					AUPR In ↑					AUPR Out ↑				
Train	Test OOD	Ind.	CEB	DICE	w only	DICE$\times w$	Ind.	CEB	DICE	w only	DICE$\times w$	Ind.	CEB	DICE	w only	DICE$\times w$	Ind.	CEB	DICE	w only	DICE$\times w$	Ind.	CEB	DICE	w only	DICE$\times w$
CIFAR-100	TinyImageNet (crop)	80.1	80.4	77.9	82.2	**73.7**	33.0	32.1	31.2	32.4	**28.8**	72.4	73.8	74.7	71.1	**77.2**	71.5	73.0	73.4	66.0	**74.3**	70.5	71.7	72.4	69.6	**75.9**
	TinyImageNet (resize)	84.4	83.6	81.0	87.9	**78.8**	35.5	34.5	33.6	35.7	**31.7**	69.1	70.6	71.7	66.1	**73.6**	68.0	69.3	**70.4**	60.5	70.3	66.8	68.5	69.3	64.4	**71.9**
	LSUN (crop)	79.1	82.7	81.1	74.9	**73.3**	28.6	29.2	29.3	29.9	**28.5**	77.7	76.2	75.9	76.3	**78.9**	**79.9**	78.6	77.8	74.7	79.2	73.9	71.8	71.8	75.1	**76.8**
	LSUN (resize)	83.1	81.0	80.0	82.5	**75.9**	34.2	32.1	31.4	32.0	**29.2**	71.5	74.2	74.2	71.6	**77..1**	73.2	75.5	75.4	66.3	**76.5**	68.3	71.5	71.2	69.7	**75.0**
	iSUN	85.3	83.4	83.8	84.2	**79.7**	35.3	33.2	33.3	34.3	**31.7**	69.6	72.5	71.9	68.7	**74.4**	72.7	**75.3**	74.7	65.6	74.7	63.7	66.9	65.7	64.9	**69.5**
	TinyImageNet+LSUN+iSUN	82.3	82.2	80.7	82.3	**76.2**	33.4	32.2	31.8	32.8	**30.2**	72.1	73.5	73.8	70.9	**76.4**	38.1	39.9	**40.3**	28.7	39.4	91.5	91.9	91.9	91.4	**93.1**
	CIFAR-10	80.1	82.9	**78.5**	90.0	79.9	30.0	30.3	28.8	36.6	**28.7**	76.6	76.0	78.1	66.5	**78.4**	79.7	79.1	**80.9**	64.6	80.8	72.5	71.6	**73.9**	62.8	**73.9**

Table 6: **Out-of-distribution** performances **after** temperature scaling.

Dataset		FPR (95 % TPR) ↓				Detection ↓				AUROC ↑				AUPR In ↑				AUPR Out ↑			
Train	Test OOD	Ind.	CEB	DICE	DICE$\times w$	Ind.	CEB	DICE	DICE$\times w$	Ind.	CEB	DICE	DICE$\times w$	Ind.	CEB	DICE	DICE$\times w$	Ind.	CEB	DICE	DICE$\times w$
CIFAR-100	TinyImageNet (crop)	77.7	78.4	77.1	**73.2**	32.1	31.2	31.5	**27.9**	74.0	74.8	75.2	**77.9**	72.6	74.1	74.0	**74.8**	72.3	73.4	73.4	**76.4**
	TinyImageNet (resize)	83.0	82.3	80.4	**78.5**	34.4	33.6	32.9	**30.7**	70.5	71.5	72.6	**74.3**	69.0	70.5	**70.9**	70.8	68.3	70.2	70.3	**72.5**
	LSUN (crop)	76.7	81.7	80.2	**72.7**	27.6	28.6	28.5	**28.2**	79.3	77.2	77.0	**79.2**	**81.2**	79.4	78.7	79.5	75.9	73.1	72.5	**77.1**
	LSUN (resize)	81.5	79.0	79.5	**75.4**	33.1	30.9	30.6	**28.3**	73.0	75.4	75.1	**78.1**	74.3	76.8	76.0	**76.9**	70.1	73.4	72.2	**75.6**
	iSUN	84.3	82.3	83.2	**79.3**	34.5	32.3	32.2	**30.8**	70.9	73.5	72.7	**75.0**	73.5	**76.4**	74.6	75.6	65.2	68.6	66.6	**70.1**
	TinyImageNet+LSUN+iSUN	80.5	80.7	80.0	**75.8**	32.3	31.3	31.2	**29.3**	73.6	74.4	74.7	**76.9**	39.3	**41.4**	40.9	39.8	92.0	92.5	92.2	**93.3**
	CIFAR-10	78.8	82.1	**78.2**	80.0	29.6	29.9	28.6	**28.3**	77.5	76.7	78.5	**78.6**	80.2	79.4	**81.2**	81.0	73.5	72.5	**74.5**	74.0

A.3 ACCURACY VERSUS SIZE

We recover from Table 7 the Memory Split Advantage (MSA) from Chirkova et al. (2020): splitting the memory budget between three branches of ResNet-32 results in better performance than spending twice the budget on one ResNet-110. DICE further improves this advantage. Our framework is particularly effective in the branch-based setting, as it reduces the computational overhead (especially in terms of FLOPS) at a slight cost in diversity. A 4-branches DICE ensemble has the same accuracy in average than a classical 7-branches ensemble.

Table 7: **Ensemble effectiveness evaluation**. Top-1 accuracy (%), number of parameters (M) and floating point operations (GFLOPs). This table is summarized in Figure 1. DICE always outperform the independent learning baseline, even with only 1 member because of the CEB component. We reduce the saturation phenomenon.

Architecture			CIFAR-100			
Network	Structure	Ens. Size	Params. (M)	GFLOPs	Ind.	DICE
ResNet-32	Base	1	0.47	0.14	71.28	**71.31**
	Branches	2	0.83	0.18	74.89	**75.40**
		3	1.19	0.23	76.28	**76.89**
		4	1.55	0.28	76.78	**77.51**
		5	1.91	0.32	77.24	**78.08**
		6	2.27	0.36	77.39	**78.29**
		7	2.63	0.40	77.52	**78.44**
		8	2.99	0.44	77.60	**78.60**
		10	3.71	0.51	77.64	**78.71**
	Nets	2	0.95	0.28	75.01	**75.32**
		3	1.42	0.42	76.45	**76.98**
		4	1.89	0.56	77.38	**77.92**
		5	2.36	0.70	77.82	**78.41**
		6	2.83	0.84	78.16	**78.83**
		7	3.29	0.98	78.36	**79.05**
		8	3.78	1.12	78.49	**79.24**
		10	4.71	1.41	78.59	**79.35**
ResNet-110	Base	1	1.73	0.51	76.21	**76.25**
	Branches	3	4.33	0.84	80.54	**81.67**
		4	5.68	1.02	80.89	**81.93**

A.4 TRAINING DYNAMICS IN TERMS OF ACCURACY, UNCERTAINTY ESTIMATION AND CALIBRATION

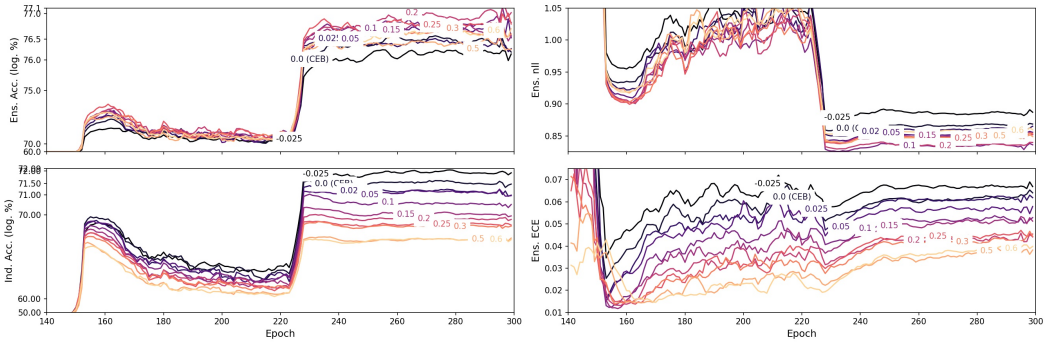

Figure 8: **Training Dynamics on the validation dataset** while training on 95% of the training dataset. A higher diversity coefficient decreases individual performance (lower left), but increases ensemble performance in terms of accuracy (upper left), uncertainty estimation (upper right) up to a value, found at $\delta_{cmi} = 0.2$ for 4-branches ResNet-32. Calibration (lower right) highly benefits from higher diversity before temperature scaling.

A.5 TRAINING DYNAMICS IN TERMS OF DIVERSITY

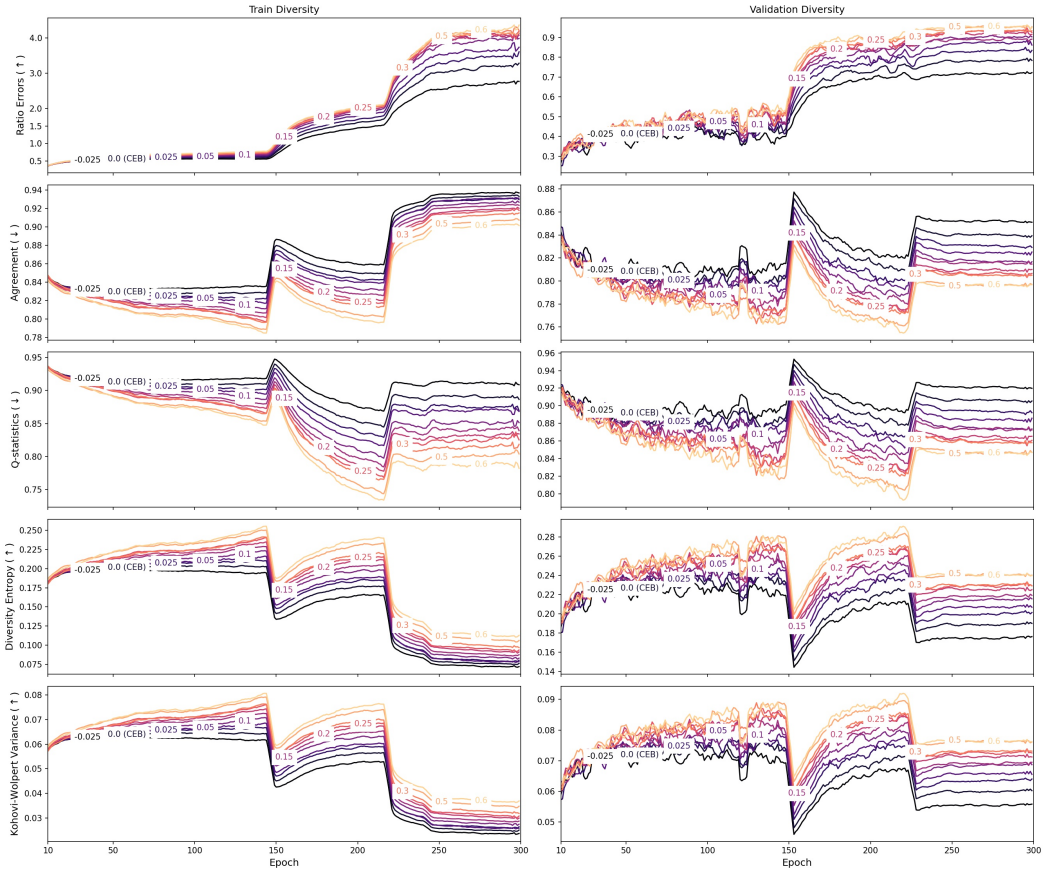

Figure 9: **Diversity dynamics** on train and validation dataset. DICE increases diversity for pairwise (ratio errors, agreement, Q-statistics) and non-pairwise (entropy, Kohavi-Wolpert variance) measures.

We measured diversity in 4.2 with the ratio error (Aksela, 2003). But as stated by Kuncheva & Whitaker (2003), diversity can be measured in numerous ways. For pairwise measures, we averaged over the $\binom{M}{2}$ pairs: the Q-statistics is positive when classifiers recognize the same object, the agreement score measures the frequency that both classifiers predict the same class. Note that even if we only apply pairwise constraints, we also increase non-pairwise measures: for example, the Kohavi-Wolpert variance (Kohavi et al., 1996) which measures the variability of the predicted class, and the entropy diversity which measures overall disagreement.

B TRAINING DETAILS

B.1 GENERAL OPTIMIZATION

Experiments We conducted all experiments in Pytorch 1.4 (Paszke et al., 2019) on an NVIDIA Tesla V100 SXM2 32 GB. Our code was forked from `https://github.com/valeoai/ConfidNet` (Corbière et al., 2019) and will be released upon acceptance. Classical hyper-parameters were taken from (Chen et al., 2019) for conducting fair comparisons. Newly added hyper-parameters were fine-tuned on a validation dataset made of 5% of the training dataset.

Architecture We implemented the proposed method with ResNet (He et al.) and Wide-ResNet (Zagoruyko & Komodakis, 2016) architectures. Following standard practices, we average the log-

its of our predictions uniformly. For branch-based ensemble, we separate the last block and the classifier of each member from the weights sharing while the other low-level layers were shared.

Learning Following (Chen et al., 2019), we used SGD with Nesterov with momentum of 0.9, mini-batch size of 128, weight decay of 5e-4, 300 epochs, a standard learning rate scheduler that sets values $\{0.1, 0.001, 0.0001\}$ at steps $\{0, 150, 225\}$ for CIFAR-10/100. In CIFAR-100, we additionally set the learning rate at 0.00001 at step 250. We used basic data augmentation that consists of horizontal flips and a random crop of 32 pixels with a padding of 4 pixels[1], but other augmentations would improve overall performance. The learning curve is shown on Figure 8.

B.2 Pseudo code

Algorithm 1: Full DICE Procedure

/* Setup */

Parameters: $\theta_1 = \{e_1, b_1, c_1\}$, $\theta_2 = \{e_2, b_2, c_2\}$ and discriminator w, randomly initialized

Input: Observations $\{x^n, y^n\}_{n=1}^N$, coefficients β_{ceb} and δ_{cmi}, schedulings $sche_{ceb}$ and $rampup_{startstep}^{endstep}$, clipping threshold τ, batch size b, optimisers $g_{\theta_{1,2}}$ and g_w, number of discriminators step $nstep_d$, number of samples num_s, ratio of positive/negative sample $ratio_{pos}^{neg}$

/* Training procedure */

for $s \leftarrow 1$ ***to*** 300 **do**

  $\beta_{ceb}^s \leftarrow sche_{ceb}(\textit{startvalue}=0, \textit{endvalue}=\beta_{ceb}, \textit{step}=s)$

  $\delta_{cmi}^s \leftarrow rampup_0^{80}(\textit{startvalue}=0, \textit{endvalue}=\delta_{cmi}, \textit{step}=s)$

  Randomly select batch $\{(x^n, y^n)\}_{n\in\mathcal{B}}$ *of size* b // Batch sampling

  /* Step 1: classification loss */

  for $i \leftarrow 1$ ***to*** 2 **do**

    $z_i^n \leftarrow e_i^\mu(z|x^n) + \epsilon e_i^\sigma(z|x^n)$, $\forall n \in \mathcal{B}$ *with* $\epsilon \sim N(0,1)$

    $VCEB_i \leftarrow \frac{1}{b} \sum_{n\in\mathcal{B}} \{\frac{1}{\beta_{ceb}^s} D_{\mathrm{KL}}(e_i(z|x^n) \| b_i(z|y^n)) - \log c_i(y^n|z_i^n\}$

  /* Step 2: diversity loss */

  for $i \leftarrow 1$ ***to*** 2 **do**

    $e_i^{\sigma,s}(z|x^n) = rampup_{100}^{250}(\textit{startvalue}=\mathbf{1}, \textit{endvalue}=e_i^\sigma(z|x^n), \textit{step}=s)$

    for $k \leftarrow 1$ ***to*** num_s **do**

      $z_{i,k}^n \leftarrow e_i^\mu(z|x^n) + \epsilon e_i^{\sigma,s}(z|x^n)$, $\forall n \in \mathcal{B}$ *with* $\epsilon \sim N(0,1)$

  $\mathcal{B}_j \leftarrow \{(z_{1,k}^n, z_{2,k}^n, y^n)\}$, $\forall n \in \mathcal{B}, \forall k \in \{1, \ldots, num_s\}$ // Joint distrib.

  $\hat{\mathcal{L}}_{DV}^{CMI} \leftarrow \frac{1}{|\mathcal{B}_j|} \sum_{t\in\mathcal{B}_j} \log f(t)$ *with* $f(t) \leftarrow \textit{tanhclip}(\frac{w(t)}{1-w(t)}, \tau)$

  $\theta_{1,2} \leftarrow g_{\theta_{1,2}}(\nabla_{\theta_1} VCEB_1 + \nabla_{\theta_2} VCEB_2 + \delta_{cmi}^s \nabla_{\theta_{1,2}} \hat{\mathcal{L}}_{DV}^{CMI})$ // Backprop ensemble

  /* Step 3: adversarial training */

  for $_ \leftarrow 1$ ***to*** $nstep_d$ **do**

    $\mathcal{B}_j \leftarrow \{(z_{1,k}^n, z_{2,k}^n, y^n)\}$, $\forall n \in \mathcal{B}, \forall k \in \{1, \ldots, num_s\}$ // Joint distrib.

    $\mathcal{B}_p \leftarrow \{(z_{1,k}^n, z_{2,k'}^{n'}, y^n)\}$, $\forall n \in \mathcal{B}, \forall k \in \{1, \ldots, num_s\}, k' \in \{1, \ldots, ratio_{pos}^{neg}\}$ *with* $n' \in \mathcal{B}$, $y^n = y^{n'}$, $n \neq n'$ // Product distribution

    $w \leftarrow g_w(\nabla_w \mathcal{L}_{ce}(w))$ // Back. adv.

    Sample new $z_{i,k}^n$

/* Test procedure */

Data: *Inputs* $\{x^n\}_{n=1}^T$ // Test data

Output: $\underset{k\in\{1,\ldots,K\}}{\arg\max}\ (\frac{1}{2}[c_1(e_1^\mu(z|x^n)) + c_2(e_2^\mu(z|x^n))])$, $\forall n \in \{1, \ldots, T\}$

[1] Compose([RandomHorizontalFlip(), RandomCrop(32, padding=4)])

B.3 INFORMATION BOTTLENECK DETAILS

Architecture We use the last feature map just before the dense layer since deeper layers are more semantics, of size $d = \{64, 128, 256\}$ for {ResNet-32, WRN-28-2, ResNet-110}. Our encoder does not provide a deterministic point in the features space but a feature distribution encoded by a mean and a diagonal covariance matrix. The covariance is predicted after a Softplus activation function with an additional dense layer, taking as input the features mean, therefore with $d(d+1)$ trainable weights. In training we sample once from this features' distribution with the reparameterization trick. In inference, we predict from the distribution's mean (and therefore only once). We parameterized $b(z|y) \sim N(b^{\mu}(y), \mathbf{1})$ with a trainable mean and an unit diagonal covariance, with d additional trainable weights per class. As noticed in (Fischer & Alemi, 2020), this can be represented as a single embedding layer mapping one-hot classes to d-dimensional tensors. Therefore in total we only add $d(d+1+K)$ trainable weights, that all can be discarded during inference. For VIB, the embedding b^{μ} is shared among classes: in total it adds $d(d+2)$ trainable weights. Contrary to recent IB approaches (Wu et al., 2019b; Wu & Fischer, 2020; Fischer & Alemi, 2020), we only have one dense layer to predict logits after the features bottleneck, and we did not change the batch normalization, for fair comparisons with traditional ensemble methods.

Scheduling We employ the jump-start method that facilitates the learning of bottleneck-inspired models (Wu et al., 2019b; Wu & Fischer, 2020; Fischer & Alemi, 2020): we progressively anneal the value of β_{ceb}. For CIFAR-10, $\log(\beta_{ceb})$ reaches values $\{100, 10, 2\}$ at steps $\{0, 5, 100\}$. For CIFAR-100, $\log(\beta_{ceb})$ reaches values $\{100, 10, 2, 1.5, 1\}$ at steps $\{0, 8, 175, 250, 300\}$. For VIB, we scheduled similarly using the equivalence from (Fischer, 2020): $\log(\beta_{ib}) = \log(\beta_{ceb} - 1)$. This slow ramp up is critical to progressively increase the covariance predictions $e^{\sigma}(x)$, facilitate learning and reduce performances' variance. Otherwise $e^{\sigma}(x)$ would early reach $\mathbf{1}$ and the high variance in the reparameterization trick would cause underfitting. We found VCEB to be less dependent on the hyper-parameters choices than VIB.

B.4 DISCRIMINATOR DETAILS

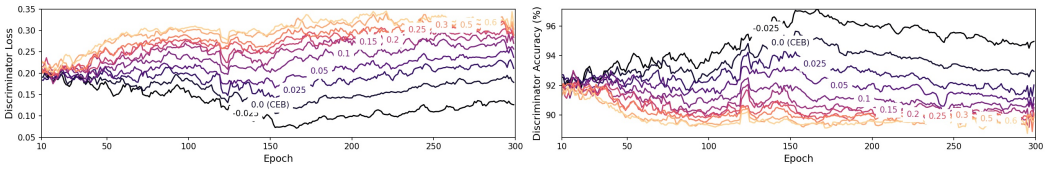

Figure 10: **Discriminator** dynamics and learning curve. The task becomes harder for higher values of δ_{cmi}: the joint and product features distributions tend to be indistinguishable.

Mutual Information Following standard adversarial learning practises, our discriminator for MI estimation is a MLP with 4 layers of size {256, 256, 100, 1}, with leaky-ReLus of slope 0.2, optimized by RMSProp with learning rate $\{0.003, 0.005\}$ for CIFAR-{10, 100}. We empirically found that four steps for the discriminator for one step of the classifier increases stability. Specifically, it takes as input the concatenation of the two hidden representations of size d, sampled with a reparameterization trick. Gradients are not back-propagated in the layer that predicts the covariance, as it would artificially increase the variance to reduce the information shared among branches. The output, followed by a sigmoid activation function, should be close to 1 (resp. 0) if the sample comes from the joint (resp. product) distribution.

Conditional Mutual Information In order to take into account the category information, the discriminator for CMI estimation needs to take into account he class information. It first embeds Y in an embedding layer of size 64, which is concatenated at the inputs of the first and second layers. Improved features merging method could be applied, such as (Ben-Younes et al., 2019). The output has size K, and we select the index associated with the sample's class. We note in Figure 11 that our discriminator remains calibrated.

Ensemble with M Models In the general case, we only consider pairwise interactions, therefore we need to estimate $\binom{M}{2}$ values. To reduce the number of parameters, we use only one discriminator w. We fill with zeros the features associated with z_k when we sample from $p(z_i, z_j, y)$ or from $p(z_i, y)p(z_j|y)$, where $i, j, k \in \{1, \dots, M\}$.

Adversarial Training We follow the traditional ramp-up (Lan et al., 2018; Kim et al., 2019; Chen et al., 2019) scheduling up to step 80 before reaching the final value of δ_{cmi}. The optimal value depends on the diversity-accuracy trade-off. For branch-based and network-based CIFAR-100, we set δ_{cmi} at $\{0.1, 0.15, 0.2, 0.22, 0.25\}$ for $\{2, 3, 4, 5, 6\}$ members. For CIFAR-10, $\{0.1\}$ for 4 members. We found that lower values of δ_{mi} were necessary for our baselines IBMI and CEBMI.

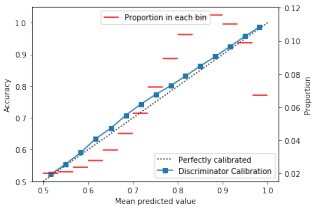

Figure 11: **The discriminator remains calibrated** even at the end of the adversarial training.

Sampling To sample from $p(z_1, z_2, y)$, we select features extracted from one image. To sample from $p(z_1, y)p(z_2|y)$, we select features extracted from two different inputs, that share the same class. In practise, we keep a memory from previous batches as the batch size is 128 whereas we have 100 classes in CIFAR-100. Our sampling is a special case of k-NN sampling (Molavipour et al., 2020): as we sample from a discrete categorical variable, the closest neighbour has exactly the same discrete value. The training can be unstable as it minimises the divergence between two distributions. To make them overlap over the features space, we sample $num_{sample} = \{4\}$ times from the gaussian distribution of Z_1 and Z_2 with the reparameterization trick. This procedure is similar to instance noise (Sønderby et al., 2016) and it allows us to safely optimise w at each iteration. It gives better robustness than just giving the gaussian mean. Moreover, we progressively ease the discriminator task by scheduling the variance through time with a linear rampup. First the variance is set to 1 until epoch 100, then it linearly reduces to the predicted variance $e^{\sigma}(x)$ until step 250. We sample a ratio $ratio_{pos}^{neg}$ of one positive pair for $\{2, 4\}$ negative pairs on CIFAR-$\{10, 100\}$

Clipping Following Bachman et al. (2019), we clip the density ratios ($tanhclip$) by computing the non linearity $\exp[\tau \tanh \frac{\log[f(z_1,z_2,y)]}{\tau}]$. A lower τ reduces the variance of the estimation and stabilizes the training even with a strong discriminator, at the cost of increasing bias. The clipping threshold τ was set to 10 like in Song & Ermon (2020).

Empirical Limitations Our approach relies on very recent works in neural network estimation of mutual information, that still suffer from high variance and loose approximations. Improvements in this area would deeply facilitate our learning procedure. Our approach increases the number of operations because of the adversarial procedure, but only during training: the inference time remains exactly the same.

C Concurrent approaches

Table 8: **Summary concurrent approaches**.

Method	Co-distillation	Diversity	I.B.	Merging	Others	Branch/Net	Consistently better than Ind.
DML (Zhang et al., 2018)	Pred. pairwise					Net	
CL-ILR (Song & Chai, 2018)	Pred.					Branch	
ONE (Lan et al., 2018)	Preds			Gate		Branch	
FFL (Kim et al., 2019)	Pred.			Feat. Fus.		Both	
OKDDip (Chen et al., 2019)	Pred. asymetric					Both	≈
KDCL (Guo et al.)	Pred.	Data Augmentation		Weights on val		Net	
PCL (Wu & Gong, 2020)	Pred.	Data Augmentation		Feat. Fus.	Mean teacher	Branch	≈
AFD (Chung et al., 2020)	Features					Net	≈
GAL (Kariyappa & Qureshi, 2019)		Gradients				Net	
GPMR (Dabouei et al., 2020)		Gradients			Grads. Magnitude	Net	
ADP (Pang et al., 2019)		Non maximum pred.			Entropy Pred.	Both	✓
DIBS (Sinha et al., 2020)		JSD Features	VIB		Custom sampling	Both	?
IB			VIB			Both	✓
CEB			VCEB			Both	✓
IBMI (Ours equation 9)		MI	VIB			Both	✓
CEBMI (Ours equation 10)		MI	VCEB			Both	✓
DICE (Ours equation 5)		CMI	VCEB			Both	✓

Concurrent approaches can be divided in two general patterns: they promote either individual accuracy by co-distillation either ensemble diversity.

C.1 CO-DISTILLATION APPROACHES

Contrary to the traditional distillation (Hinton et al., 2015) that aligns the soft prediction between a static pre-trained strong teacher towards a smaller student, online co-distillation performs teaching in an end-to-end one-stage procedure: the teacher and the student are trained simultaneously.

Distillation in logits The seminal "Deep Mutual Learning" (DML) (Zhang et al., 2018) introduced the main idea: multiple networks learn to mimic each other by reducing KL-losses between pairs of predictions. "Collaborative learning for deep neural networks" (CL-ILR) (Song & Chai, 2018) used the branch-based architecture by sharing low-level layers to reduce the training complexity, and "Knowledge Distillation by On-the-Fly Native Ensemble" (ONE) (Lan et al., 2018) used a weighted combination of logits as teacher hence providing better information to each network. "Online Knowledge Distillation via Collaborative Learning" (KDCL) (Guo et al.) computed the optimum weight on an held-out validation dataset. "Feature Fusion for Online Mutual Knowledge Distillation" (FFL) (Kim et al., 2019) introduced a feature fusion module. These approaches improve individual performance at the cost of increased homogenization. "Online Knowledge Distillation with Diverse Peers" (OKDDip) (Chen et al., 2019) slightly alleviates this problem with an asymmetric distillation and a self-attention mechanism. "Peer Collaborative Learning for Online Knowledge Distillation" (PCL) (Wu & Gong, 2020) benefited from the mean-teacher paradigm with temporal ensembling and from diverse data augmentation, at the cost of multiple inferences through the shared backbone.

Distillation in features Whereas all previous approaches only apply distillation on the logits, the recent "Feature-map-level Online Adversarial Knowledge Distillation" (AFD) (Chung et al., 2020) aligned features distributions by adversarial training. Note that this is not opposite to our approach, as they force distributions to be similar while we force them to be uncorrelated.

C.2 DIVERSITY APPROACHES

On the other hands, some recent papers in computer vision explicitly encourage diversity among the members with regularization losses.

Diversity in logits "Diversity Regularization in Deep Ensembles" (Shui et al., 2018) applied negative correlation (Liu & Yao, 1999a) to regularize the training for improved calibration, with no impact on accuracy. "Adaptive Diversity Promoting" (ADP) (Pang et al., 2019) decorrelates only the non-maximal predictions to maintain the individual accuracies, while promoting ensemble entropy. It forces different members to have different ranking of predictions among non maximal predictions. However, Liang et al. (2017) has shown that ranking of outputs are critical: for example, non maximal logits tend to be more separated from each other for in-domain inputs compared to out-of-domain inputs. Therefore individual accuracy are decreased. We apply the same scheduling for weighting the diversity-inducing loss in ADP and found it to stabilize their results and increase accuracy. Coefficients α and β are respectively set to 2 and 0.5, as in the original paper.

Diversity in features One could think about increasing classical distances among features like L_2 in (Kim et al., 2018b), but in our experiments it reduces overall accuracy: it is not even invariant to linear transformations such as translation. "Diversity inducing Information Bottleneck in Model Ensembles" from Sinha et al. (2020) trains a multi-branch network and applies VIB on individual branch, by encoding $p(z|y) \sim \mathcal{N}(0, 1)$, which was shown to be hard to learn (Wu & Fischer, 2020). Moreover, we notice that their diversity-inducing adversarial loss is an estimation of the JS-divergence between pairs of features, built on the dual f-divergence representation (Nowozin et al., 2016): similar idea was recently used for saliency detection (Chen et al.). As the JS-divergence is a symmetrical formulation of the KL, we argue that DIBS and IBMI share the same theoretical motivations and diverge only on few optimization tricks: the adversarial terms with both terms sampled from the same branch and both terms sampled from the same prior. In our experiments, these

differences reduce overall performance. We will include their scores when they publish measurable results on CIFAR datasets or when they release their code.

Diversity in gradients "Improving adversarial robustness of ensembles with diversity training." (GAL) (Kariyappa & Qureshi, 2019) enforced diversity in the gradients with a gradient alignment loss. "Exploiting Joint Robustness to Adversarial Perturbations" (Dabouei et al., 2020) considered the optimal bound for the similarity of gradients. However, as stated in the latter, "promoting diversity of gradient directions slightly degrades the classification performance on natural examples ... [because] classifiers learn to discriminate input samples based on distinct sets of representative features". Therefore we do not consider them as concurrent work.

D EXPERIMENTAL SETUP

D.1 TRAINING DATASETS

We train our procedure on two image classification benchmarks, CIFAR-100 and CIFAR-10, (Krizhevsky et al., 2009). They consist of 60k 32*32 natural and colored images in respectively 100 classes and 10 classes, with 50k training images and 10k test images. For hyper-parameters selection and ablation studies, we train on 95% of the training dataset, and analyze performances on the validation dataset made of the remaining 5%.

D.2 ROBUSTNESS

Dataset For CIFAR-100, we evaluate results on shifted data from *The Common Corruptions Benchmark* (Hendrycks & Dietterich, 2019): CIFAR-100-C modifies the CIFAR-100 test set with 15 corruptions (speckle noise, shot noise, impulse noise, defocus blur, gaussian blur, glass blur, motion blur, zoom blur, snow, fog, brightness, contrast, elastic transform, pixelate and jpeg compression) at five different levels. Note that none of these transformations are used in training.

Metrics We follow the procedure from (Ashukha et al., 2020). To measure the calibration, i.e. how classification confidences match the observed prediction accuracy, we report the Expected Calibration Error (ECE) (Naeini et al., 2015) and the Thresholded Adaptive Calibration Error (TACE) (Nixon et al.) with 15 bins: while they both evaluate the calibration properties of our models, TACE resolve some pathologies found in ECE by thresholding and adaptive binning. To evaluate the quality of the uncertainty estimates, we reported two complementary proper scoring rules (Gneiting & Raftery, 2007), the Negative Log-Likelihood (NLL) and the Brier Score (BS) (Brier, 1950).

Temperature scaling On CIFAR-100, we randomly divide the test set into two equal parts, then compute metrics for each half of the test set using the temperature optimized on another half. For CIFAR-100c, we compute the temperature on the full CIFAR-100 test set. These 3 values have low variance and their mean is reported in Table 2 under the name "Temp. avg value".

D.3 OOD

Dataset We used the traditional out-of-distribution datasets for CIFAR-100, described in (Liang et al., 2017): TinyImageNet (Deng et al., 2009), LSUN (Yu et al., 2015), iSUN(Xu et al., 2015), and CIFAR-10. We run all evaluations with the code from `https://github.com/uoguelph-mlrg/confidence_estimation` (DeVries & Taylor, 2018).

Metrics We reported the standard metrics for binary classification: FPR at 95 % TPR, Detection error, AUROC (Area Under the Receiver Operating Characteristic curve) and AUPR(Area under the Precision-Recall curve, -in or -out depending on which dataset is specified as positive). See (Liang et al., 2017) for definitions and interpretations of these metrics

E ADDITIONAL THEORETIC COMPONENTS

E.1 BIAS VARIANCE COVARIANCE DECOMPOSITION

The Bias-Variance-Covariance Decomposition (Ueda & Nakano, 1996) generalizes the Bias-Variance Decomposition (Kohavi et al., 1996) by treating the ensemble of M members as a single learning unit.

$$\mathbb{E}[(\overline{f} - t)^2] = \overline{\text{bias}}^2 + \frac{1}{M}\overline{\text{var}} + (1 - \frac{1}{M})\overline{\text{covar}}, \tag{7}$$

with

$$\overline{\text{bias}} = \frac{1}{M}\sum_i (\mathbb{E}[f_i] - t),$$

$$\overline{\text{var}} = \frac{1}{M}\sum_i \mathbb{E}[(\mathbb{E}[f_i] - t)^2],$$

$$\overline{\text{covar}} = \frac{1}{M(M-1)}\sum_i \sum_{j \neq i} \mathbb{E}[(f_i - \mathbb{E}[f_i])(f_j - \mathbb{E}[f_j])].$$

The estimation improves when the covariance between members is zero, i.e when the errors are unbiased and uncorrelated. Compared to the Bias-Variance Decomposition (Kohavi et al., 1996), it leads to a variance reduction of $\frac{1}{M}$. Brown et al. (2005a;b) summarized it this way: "in addition to the bias and variance of the individual estimators, the generalisation error of an ensemble also depends on the covariance between the individuals. This raises the interesting issue of why we should ever train ensemble members separately; why shouldn't we try to find some way to capture the effect of the covariance in the error function?". We could not agree more.

E.2 MUTUAL INFORMATION

Nobody knows what entropy really is.

John Van Neumann to *Claude Shannon*

At the cornerstone of the Shannon's information theory in 1948 (Shannon, 1948), Mutual Information (MI) is the difference between the sum of individual entropies and the entropy of the variables considered jointly. Stated otherwise, it is the reduction in the uncertainty of one variable due to the knowledge of the other variable (Cover, 1999). Entropy owed its name to the thermodynamic measure of uncertainty introduced by Rudolf Clausius and developed by Ludwig Boltzmann.

$$\begin{aligned} &I(Z_1; Z_2) = H(Z_1) + H(Z_2) - H(Z_1, Z_2) \\ &= H(Z_1) - H(Z_1|Z_2) \\ &= D_{\text{KL}}(P(Z_1, Z_2) \| P(Z_1)p(Z_2)). \end{aligned}$$

The conditional mutual information (CMI) generalizes MI when a third variable is given:

$$I(Z_1; Z_2|Y) = D_{\text{KL}}(P(Z_1, Z_2|Y) \| P(Z_1|Y)p(Z_2|Y)).$$

E.3 KL BETWEEN GAUSSIANS

The Kullback-Leibler divergence (Kullback, 1959) between two gaussian distributions takes a particularly simple form:

$$D_{\text{KL}}(e(z|x) \| b(z|y)) = \log \frac{b^\sigma(y)}{e^\sigma(x)} + \frac{e^\sigma(x)^2 + (e^\mu(x) - b^\mu(y))^2}{2b^\sigma(y)^2} - \frac{1}{2} \qquad \text{(Gaussian param.)}$$

$$= \frac{1}{2}[\underbrace{(1 + e^\sigma(x)^2 - \log(e^\sigma(x)^2))}_{\text{Variance}} + \underbrace{(e^\mu(x) - b^\mu(y))^2}_{\text{Mean}}]. \quad (b^\sigma(y) = \mathbf{1})$$

The variance component forces the predicted variance $e^\sigma(x)$ to be close to $b^\sigma(y) = \mathbf{1}$. The mean component forces the class-embedding $b^\mu(y)$ to converge to the average of the different elements

in its class. These class-embeddings are similar to class-prototypes, highlighting a theoretical links between CEB (Fischer, 2020; Fischer & Alemi, 2020) and prototype based learning methods (Liu & Nakagawa, 2001).

E.4 DIFFERENCE VCEB AND VIB

In Fischer (2020), CEB is variationally upper bounded by VCEB. We detail the computations:

$$
\begin{aligned}
\text{CEB}_{\beta_{ceb}}(Z) &= \frac{1}{\beta_{ceb}} I(X;Z|Y) - I(Y;Z) && \text{(Definition)}\\
&= \frac{1}{\beta_{ceb}} [I(X,Y;Z) - I(Y;Z)] - I(Y;Z) && \text{(Chain rule)}\\
&= \frac{1}{\beta_{ceb}} [I(X;Z) - I(Y;Z)] - I(Y;Z) && \text{(Markov assumptions)}\\
&= \frac{1}{\beta_{ceb}} [-H(Z|X) + H(Z|Y)] - [H(Y) - H(Y|Z)] && \text{(MI as diff. of 2 ent.)}\\
&\leq \frac{1}{\beta_{ceb}} [-H(Z|X) + H(Z|Y)] - [-H(Y|Z)] && \text{(Non-negativity of ent.)}\\
&= \int \{\frac{1}{\beta_{ceb}} \log \frac{e(z|x)}{p(z|y)} - \log p(y|z)\} p(x,y,z) \partial x \partial y \partial z && \text{(Definition of ent.)}\\
&\leq \int \{\frac{1}{\beta_{ceb}} \log \frac{e(z|x)}{b(z|y)} - \log c(y|z)\} p(x,y) e(z|x) \partial x \partial y \partial z && \text{(Variational approx.)}\\
&\approx \frac{1}{N} \sum_{n=1}^{N} \int \{\frac{1}{\beta_{ceb}} \log \frac{e(z|x^n)}{b(z|y^n)} - \log c(y^n|z)\} e(z|x^n) \partial z && \text{(Empirical data distrib.)}\\
&\approx \text{VCEB}_{\beta_{ceb}}(\theta = \{e,b,c\}), && \text{(Reparameterization trick)}
\end{aligned}
$$

where

$$
\text{VCEB}_{\beta_{ceb}}(\theta = \{e,b,c\}) = \frac{1}{N} \sum_{n=1}^{N} \{\frac{1}{\beta_{ceb}} D_{\text{KL}}(e(z|x^n) \| b(z|y^n)) - \mathbb{E}_{\epsilon} \log c(y^n | e(x^n, \epsilon)\}.
$$

As a reminder, Alemi et al. (2017) upper bounded: $\text{IB}_{\beta_{ib}}(Z) = \frac{1}{\beta_{ib}} I(X;Z) - I(Y;Z)$ by :

$$
\text{VIB}_{\beta_{ib}}(\theta = \{e,b,c\}) = \frac{1}{N} \sum_{n=1}^{N} \{\frac{1}{\beta_{ib}} D_{\text{KL}}(e(z|x^n) \| b(z)) - \mathbb{E}_{\epsilon} \log c(y^n | e(x^n, \epsilon)\}.
$$

In VIB, all features distribution $e(z|x)$ are moved towards the same class-agnostic distribution $b(z) \sim N(\mu, \sigma)$, independently of y. In VCEB, all $e(z|x)$ are moved towards the class conditional marginal $b^{\mu}(y) \sim N(b^{\mu}(y), b^{\sigma}(y))$. This is the **unique difference between VIB and VCEB**: VIB leads to a looser approximation with more bias than VCEB.

E.5 TRANSFORMING IBMI AND CEBMI INTO TRACTABLE LOSSES

In this section we derive the variational approximation of the IBMI criterion, defined by:

$$
\text{IBMI}_{\beta_{ib}, \delta_{mi}}(Z_1, Z_2) \equiv \text{IB}_{\beta_{ib}}(Z_1) + \text{IB}_{\beta_{ib}}(Z_2) + \delta_{mi} I(Z_1; Z_2).
$$

Mutual Information Estimation To estimate the MI component, we apply the same procedure than for CMI but without the categorical constraint, as in the seminal work of Belghazi et al. (2018). Let $\mathcal{B}_j$ and $\mathcal{B}_p$ be two random batches sampled respectively from the observed joint distribution $p(z_1, z_2) = p(e_1(z|x), e_2(z|x))$ and the product distribution $p(z_1)p(z_2) = p(e_1(z|x))p(e_2(z|x'))$, where x, x' are two images. We similarly train a network w that tries to discriminate these two distributions. With $f = \frac{w}{1-w}$, the MI estimation is:

$$
\hat{\mathcal{I}}_{DV}^{MI} = \frac{1}{|\mathcal{B}_j|} \sum_{(z_1,z_2) \in \mathcal{B}_j} \underbrace{\log f(z_1, z_2)}_{\text{Diversity}} - \log(\frac{1}{|\mathcal{B}_p|} \sum_{(z_1,z_2') \in \mathcal{B}_p} f(z_1, z_2')),
$$

and the final loss:

$$
\hat{\mathcal{L}}_{DV}^{MI}(e_1, e_2) = \frac{1}{|\mathcal{B}_j|} \sum_{(z_1,z_2) \in \mathcal{B}_j} \log f(z_1, z_2). \tag{8}
$$

IBMI Finally we train $\theta_1 = \{e_1, b_1, c_1\}$ and $\theta_2 = \{e_2, b_2, c_2\}$ jointly by minimizing:

$$\mathcal{L}_{IBMI}(\theta_1, \theta_2) = \text{VIB}_{\beta_{ib}}(\theta_1) + \text{VIB}_{\beta_{ib}}(\theta_2) + \delta_{mi}\hat{\mathcal{L}}^{MI}_{DV}(e_1, e_2). \tag{9}$$

CEBMI For ablation study, we also consider a model that would benefit from CEB tight approximation while having MI regularization:

$$\mathcal{L}_{CEBMI}(\theta_1, \theta_2) = \text{VCEB}_{\beta_{ceb}}(\theta_1) + \text{VCEB}_{\beta_{ceb}}(\theta_2) + \delta_{mi}\hat{\mathcal{L}}^{MI}_{DV}(e_1, e_2). \tag{10}$$

E.6 DICE FOR M MEMBERS

We trivially expand our framework for $M > 2$ using the same variational lower bounds and neural estimations. The training loss is:

$$\mathcal{L}_{DICE}(\theta_{1:M}) = \sum_{m=1}^{M} \text{VCEB}_{\beta_{ceb}}(\theta_m) + \frac{\delta_{cmi}}{(M-1)} \sum_{m=1}^{M} \sum_{m'=m+1}^{M} \hat{\mathcal{L}}^{CMI}_{DV}(e_m, e_{m'}). \tag{11}$$

E.7 CAN WE LEARN FEATURES INDEPENDENCE WITHOUT INFORMATION BOTTLENECK?

In the Markov chain $Z_1 \leftarrow X \rightarrow Z_2$, the chain rules provides:

$$I(X; Z_2|Z_1) = I(X, Z_1; Z_2) - I(Z_1; Z_2) = I(X; Z_2) - I(Z_1; Z_2),$$

therefore $I(Z_1; Z_2|Y) \leq I(X; Z_2|Y)$. By symmetry we have $I(Z_1; Z_2|Y) \leq I(X; Z_1|Y)$.

In first order, reducing $I(Z_1, X|Y)$ and $I(Z_2, X|Y)$ has an impact on $I(Z_1, Z_2|Y)$. That's the reason why CEB has higher diversity than a deterministic baseline. Our adversarial loss is a second order improvement, that cannot leverage a parameterized variational approximation. Moreover, without the IB component, there is no constraint on the predicted variance. We tried different approaches, with variance set to $\mathbf{1}$, with and without the reparametrization trick in classification and in adversarial training: we experimentally verify that learning only the adversarial loss without bottleneck losses (VIB or VCEB) yields unstable results. In conclusion, DICE benefits from both components.

E.8 IS THE THEORETIC FORMULATION STILL VALID IN THE BRANCH-BASED SETTING ?

Yes! It exactly remains the same ... except for the last assumption that "$Z_1 \leftarrow X \rightarrow Z_2$ is a Markov chain". In this case, we just need to consider that X is the latest shared features among all members.

F SOCIOLOGICAL ANALOGY

We showed that increasing diversity in features while encouraging the different learners to agree improves performance: the optimal diversity-accuracy trade-off was obtained with a large diversity in neural networks. To finish, we make a short discussion about the impact of diversity in our society. Decision making in group is better than individual decision as long as the members do not belong to the same cluster. Homogenization of the decision makers increases vulnerability to failures, whereas diversity among backgrounds sparks new discoveries (Muldoon, 2016): ideas should be shared and debated among members reflecting the diversity of the society's various components. Academia especially needs this diversity to promote the trust in research (Sierra-Mercado & Lázaro-Muñoz, 2018), to improve the quality of the findings (Swartz et al., 2019), productivity of the teams (Vasilescu et al., 2015) and even schooling's impact (Bowman, 2013).