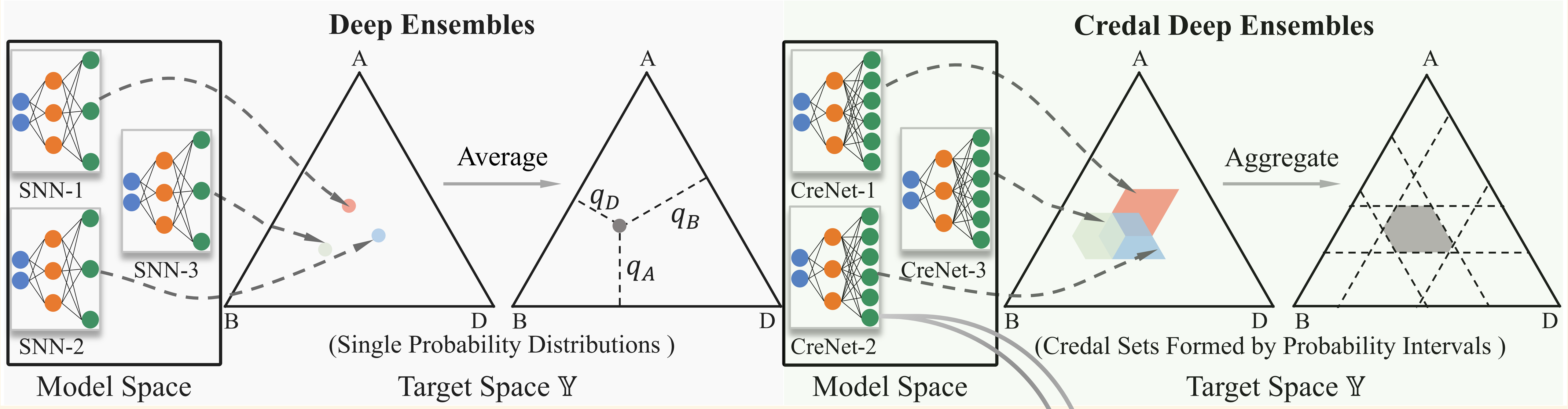
Credal Deep Ensembles for Uncertainty Quantification

Kaizheng Wang[1,4], Fabio Cuzzolin[3], Shireen Kudukkil Manchingal[3], Keivan Shariatmadar[2,4], David Moens[2,4], Hans Hallez[1]
[1]KU Leuven, Department of Computer Science, DistriNet [2]KU Leuven, Department of Mechanical Engineering, LMSD
[3]Oxford Brookes University, Visual Artificial Intelligence Laboratory [4]Flanders Make@KU Leuven

Concept & Methodology



Credal Set Prediction via Probability Intervals

- *Generating Probability Intervals via Interval SoftMax*

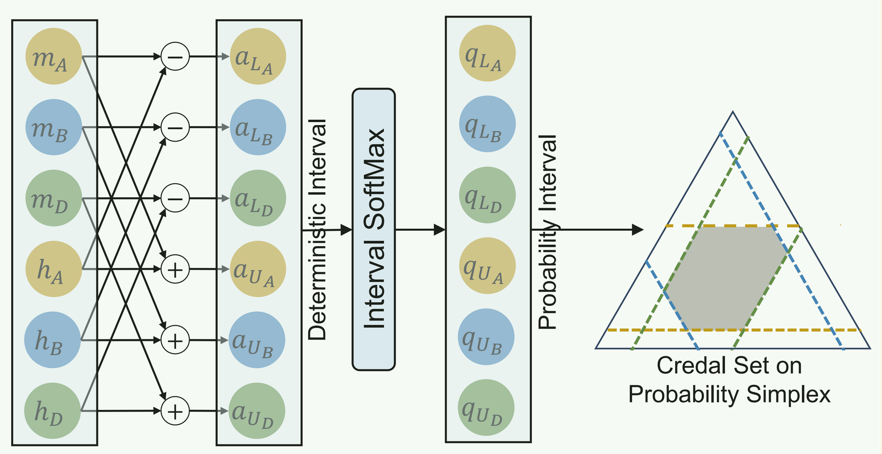
$$q_{L_i}=\frac{\exp(a_{L_i})}{\exp(a_{L_i})+\sum_{k\neq i}\exp\left(\frac{a_{L_k}+a_{L_k}}{2}\right)};\ q_{U_i}=\frac{\exp(a_{U_i})}{\exp(a_{U_i})+\sum_{k\neq i}\exp\left(\frac{a_{L_k}+a_{L_k}}{2}\right)}$$

Probability interval vector $[\boldsymbol{q}_L,\boldsymbol{q}_U]\coloneqq\{q_{L_i},q_{U_i}\}_i^C$ satisfies:

$$q_{L_i}\leq q_{U_i}\ \forall i=1,\dots,C;\ \sum_i^C q_{L_i}\leq 1\leq\sum_i^C q_{U_i}.$$

- *Defining a Non-empty Credal Set* $\mathbb{Q}$

$$\mathbb{Q}=\{\boldsymbol{q}\,|\,q_i\in[q_{L_i},q_{U_i}]\ \forall i;\ \textstyle\sum_i^C q_i=1\}$$



CreNet Final Layer Structure

Training Procedure

- *Vanilla Training Strategy*

$$\underset{\boldsymbol{\theta}\in\Theta}{\text{minimize}}\left\{\frac{1}{N}\sum_n^N\mathcal{L}((\boldsymbol{x}_n,\boldsymbol{t}_n),\boldsymbol{\theta})\right\}$$

$\{\boldsymbol{x}_n,\boldsymbol{t}_n\}_n^N$: training set; $\boldsymbol{\theta}$ trainable model parameter in the space Θ; $\mathcal{L}(\cdot,\cdot)$: an arbitrary loss function.

- *Distributionally Robust Optimization (DRO) using Adversarially Reweighted Learning*

$$\underset{\boldsymbol{\theta}\in\Theta}{\text{minimize}}\left\{\underset{\boldsymbol{w}\in\mathbb{S}}{\text{maximize}}\frac{1}{N}\sum_n^N w_n\mathcal{L}((\boldsymbol{x}_n,\boldsymbol{t}_n),\boldsymbol{\theta})\right\}$$

A minimax game between a learner and an adversary.
w_n: an adversarial assignment of weights, collected in $\boldsymbol{w}$; the set $\mathbb{S}$ of weight vectors varies across implementations.

- *CreNet Loss Design*

$$\mathcal{L}_{\text{cre}}=\frac{1}{N}\sum_n^N\text{CE}(\boldsymbol{q}_{U_n},\boldsymbol{t}_n)+\underset{\boldsymbol{w}\in\mathbb{S}}{\text{maximize}}\frac{1}{N}\sum_n^N w_n\text{CE}(\boldsymbol{q}_{L_n},\boldsymbol{t}_n)$$

CE: cross-entropy loss.

Vanilla Component

- Take the training distribution at face value.
- Encourage "optimistic/upper-bound" predictions.

DRO Component

- Weigh training outliers to simulate future differences in data distribution at test time.
- Encourage "pessimistic/lower-bound" predictions.

- *CreNet Loss Implementation*

Algorithm 1 CreNet Training Procedure

Input: Training dataset $\mathbb{D}=\{\boldsymbol{x}_n,\boldsymbol{t}_n\}_{n=1}^N$; Portion of samples per batch $\delta\in[0.5,1)$; Batch size η
while *enable training* **do**
1. Compute $\text{CE}(\boldsymbol{q}_{U_n},\boldsymbol{t}_n)$ and $\text{CE}(\boldsymbol{q}_{L_n},\boldsymbol{t}_n)$ for each sample
2. Sort the sample indices $(m_1,\dots,m_\eta)$ in descending order of $\text{CE}(\boldsymbol{q}_{L_n},\boldsymbol{t}_n)$
3. Define $\eta_\delta=\lfloor\delta\eta\rfloor$
4. Minimize $\mathcal{L}_{\text{CreNet}}=\frac{1}{\eta}\sum_{n=1}^{\eta}\text{CE}(\boldsymbol{q}_{U_n},\boldsymbol{t}_n)+\frac{1}{\delta\cdot\eta}\sum_{j=1}^{\eta_\delta}\text{CE}(\boldsymbol{q}_{L_{m_j}},\boldsymbol{t}_{m_j})$
end while

Class Prediction & Uncertainty Quantification

- *Maximax and Maximin Criteria for Class Prediction*

$$\hat{\imath}_{\text{min}}\coloneqq\underset{i}{\text{argmax}}\ q_{L_i}^*;\ \hat{\imath}_{\text{max}}\coloneqq\underset{i}{\text{argmax}}\ q_{U_i}^*$$

Output the class indices with the highest lower and upper reachable probability, respectively.

$$q_{L_i}^*=\max\left(q_{L_i},1-\textstyle\sum_{j\neq i}q_{U_j}\right);\ q_{U_i}^*=\min\left(q_{U_i},1-\textstyle\sum_{j\neq i}q_{L_j}\right)$$

- *Generalized Shannon Entropy for Uncertainty Quantification*

$$\overline{H}(\mathbb{Q})=\text{maximize}\textstyle\sum_i^C-q_i\log_2 q_i\ \text{ s.t. }\ \sum_i^C q_i=1;\ q_{L_i}^*\leq q_i\leq q_{U_i}^*$$

For $\underline{H}(\mathbb{Q})$, replace maximize by minimize.

Aleatoric uncertainty (AU) and epistemic uncertainty (EU) are measured by $\underline{H}(\mathbb{Q})$ and $\overline{H}(\mathbb{Q})-\underline{H}(\mathbb{Q})$, respectively.

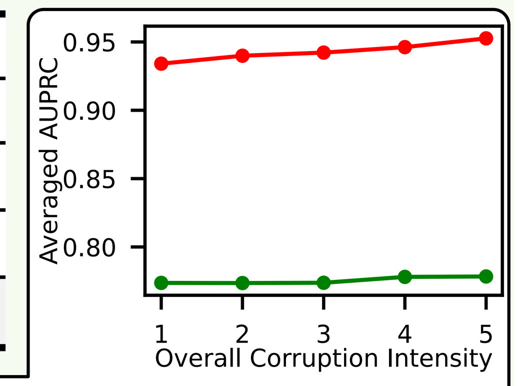
Experimental Validation

Table 1. OOD detection performance (%, ↑) using EU between on ResNet50 architecture.

	CIFAR10 (ID)				CIFAR100 (ID)				ImageNet (ID)	
	SVHN (OOD)		Tiny-ImageNet (OOD)		SVHN (OOD)		Tiny-ImageNet (OOD)		ImageNet-O (OOD)	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
DEs	$89.58_{\pm0.93}$	$92.29_{\pm1.00}$	$86.87_{\pm0.20}$	$83.02_{\pm0.16}$	$73.83_{\pm1.97}$	$84.96_{\pm1.25}$	$78.80_{\pm0.20}$	$74.68_{\pm0.27}$	$65.03_{\pm0.53}$	$62.77_{\pm0.38}$
CreDEs	$\mathbf{96.55}_{\pm0.25}$	$\mathbf{98.17}_{\pm0.17}$	$\mathbf{88.10}_{\pm0.26}$	$\mathbf{87.85}_{\pm0.35}$	$\mathbf{78.55}_{\pm1.15}$	$\mathbf{86.57}_{\pm0.65}$	$\mathbf{82.54}_{\pm0.26}$	$\mathbf{77.60}_{\pm0.44}$	$\mathbf{67.82}_{\pm0.06}$	$\mathbf{62.80}_{\pm0.12}$

Table 2. OOD detection performance (%, ↑) using EU between on different architecture.

	CIFAR10 (ID) [VGG16]				CIFAR10 (ID) [ViT Base]			
	SVHN (OOD)		Tiny-ImageNet (OOD)		SVHN (OOD)		Tiny-ImageNet (OOD)	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
DEs	$82.19_{\pm0.82}$	$87.52_{\pm0.81}$	$78.58_{\pm0.15}$	$73.28_{\pm0.23}$	$77.71_{\pm1.97}$	$88.73_{\pm0.32}$	$82.27_{\pm0.79}$	$78.85_{\pm0.81}$
CreDEs	$\mathbf{87.68}_{\pm0.73}$	$\mathbf{93.47}_{\pm0.67}$	$\mathbf{82.56}_{\pm0.28}$	$\mathbf{80.81}_{\pm0.52}$	$\mathbf{88.57}_{\pm2.08}$	$\mathbf{93.24}_{\pm1.25}$	$\mathbf{82.54}_{\pm0.26}$	$\mathbf{87.84}_{\pm0.52}$



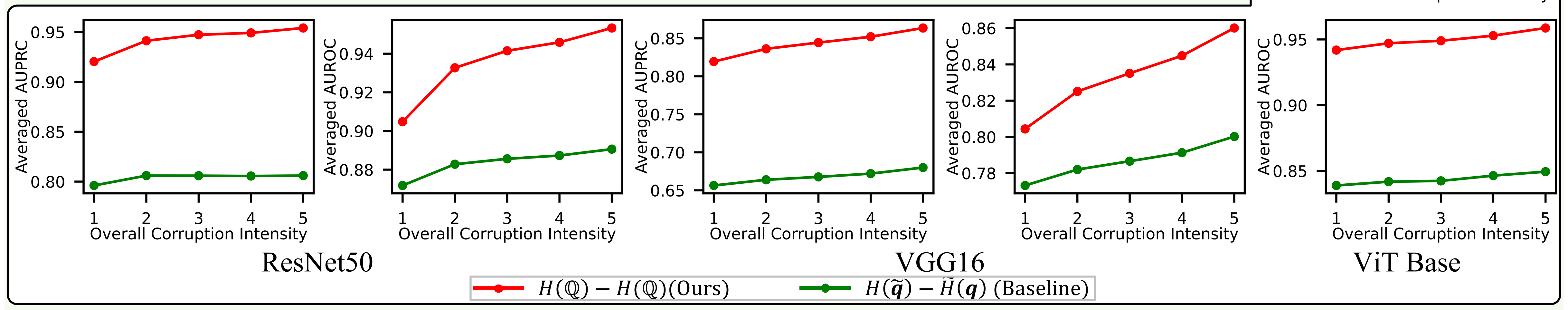


Figure 1. OOD detection (CIFAR10 vs CIFAR10-C) over increased corruption intensity on distinct architecture.

Table 3. Test ACC (%, ↑) and ECE (↓) on ResNet50 architecture.

		CIFAR10		CIFAR100		ImageNet	
		ACC (%)	ECE	ACC (%)	ECE	ACC (%)	ECE
DEs		$93.32_{\pm0.13}$	$0.013_{\pm0.001}$	$73.83_{\pm1.97}$	$0.039_{\pm0.003}$	$77.92_{\pm0.02}$	$0.242_{\pm0.001}$
CreDEs	$\hat{\imath}_{\text{min}}$	$\mathbf{93.73}_{\pm0.11}$	$\mathbf{0.009}_{\pm0.002}$	$\mathbf{79.54}_{\pm0.21}$	$\mathbf{0.027}_{\pm0.002}$	$\mathbf{78.41}_{\pm0.02}$	$0.593_{\pm0.001}$
	$\hat{\imath}_{\text{max}}$	$\mathbf{93.74}_{\pm0.11}$	$\mathbf{0.011}_{\pm0.002}$	$\mathbf{79.65}_{\pm0.19}$	$\mathbf{0.027}_{\pm0.002}$	$\mathbf{78.51}_{\pm0.02}$	$\mathbf{0.169}_{\pm0.000}$

Additional Findings

- Uncertainty quantification performance robust against training hyper parameter δ
- Improved total uncertainty estimation quality
- EU quantification quality robust against different measures like generalized Hartley measure
- Enhanced uncertainty quantification compared to deep ensembles that applied the DRO strategy or 'product of experts' strategy and several Bayesian neural network baselines
- Superior performance in a case study of active learning
- Marginal increase in inference complexity compared to deep ensembles

FULL PAPER