universite PARIS-SACLAY

SSFAM

A Symmetry-Aware Exploration of Bayesian Neural Network Posteriors

Olivier Laurent, Emanuel Aldea & Gianni Franchi◇
◇ gianni.franchi@ensta-paris.fr

ENSTA IP PARIS

ICLR

1 - Motivation & Contributions

Most successful deep learning **uncertainty quantification** methods — Deep Ensembles [1], SWA(G) [2], Laplace, Monte-Carlo Dropout [3], etc — seek to approximate the **Bayesian posterior** via **marginalization over the weights** [4]:

$$p(y \mid \boldsymbol{x}, \mathcal{D}) = \int_{\omega \in \Omega} p(y \mid \boldsymbol{x}, \omega) p(\omega \mid \mathcal{D}) d\omega$$

However, in modern deep neural networks, there exists a large or infinite number of equivalent weight configurations [5]:

- ❖ **Scaling** the weights by sequences of coefficients and their inverses leaves the function unchanged.
- ❖ **Reordering** the weights using sequences of permutation matrices and their inverses does not change the function.

→ Symmetries impact optimization, generalization via the loss landscape.

? What is the **impact of symmetries** on **Bayesian posteriors** and their estimation by **uncertainty quantification** methods?

Contributions

A - Express the **theoretical impact** of perm./scale **symmetries** on the **posterior**.

B - Evaluate & discuss the **impact** of symmetries on **UQ** methods.

C - The **min.** of the **weight decay** on scaling symmetries has a **unique** solution.

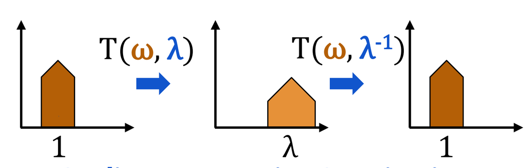
D - **"Checkpoints"**: **dataset** of medium-sized independently trained **models**.

Prior

We follow practitioners and work with *i.i.d.* **Gaussian priors**: weight decay.

2 - Notations & Theory

x, y: inputs and targets
D: data distribution
$\triangledown$, $\triangleright$: row-wise and col.-wise products



Scaling symmetries & activations

→ Scaling symmetries

For all $\boldsymbol{\theta} \in \mathbb{R}^{\cdot \times m}, \boldsymbol{\omega} \in \mathbb{R}^{m \times n}, \boldsymbol{\lambda} \in (\mathbb{R}_{>0})^m, \forall \boldsymbol{x} \in \mathbb{R}^n,\ (\boldsymbol{\lambda}^{-1} \triangledown \boldsymbol{\theta}) \times r(\boldsymbol{\lambda} \triangleright \boldsymbol{\omega x}) = \boldsymbol{\theta} \times r(\boldsymbol{\omega x})$

→ Permutation symmetries

For all $\boldsymbol{\theta} \in \mathbb{R}^{\cdot \times m}, \boldsymbol{\omega} \in \mathbb{R}^{m \times n}, \boldsymbol{\pi} \in P_m, \forall \boldsymbol{x} \in \mathbb{R}^n,\ \boldsymbol{\theta \pi}^\intercal \times r(\boldsymbol{\pi} \times \boldsymbol{\omega x}) = \boldsymbol{\theta} \times r(\boldsymbol{\omega x})$

Proposition:

With $\tilde{\Omega}$ the *r.v.* of the scaled and sorted weights, the final posterior is a continuous mixture of discrete mixtures of the "original" posterior $p(\tilde{\Omega} \mid \mathcal{D})$:

$$p(\Omega \mid \mathcal{D}) = |\Pi|^{-1} \int_{\Lambda \in \mathbb{\Lambda}} \sum_{\Pi \in \Pi} \mathcal{T}_p(\mathcal{T}_s(p(\tilde{\Omega} \mid \mathcal{D}), \Lambda), \Pi) p(\Lambda) d\Lambda$$

Corollary:

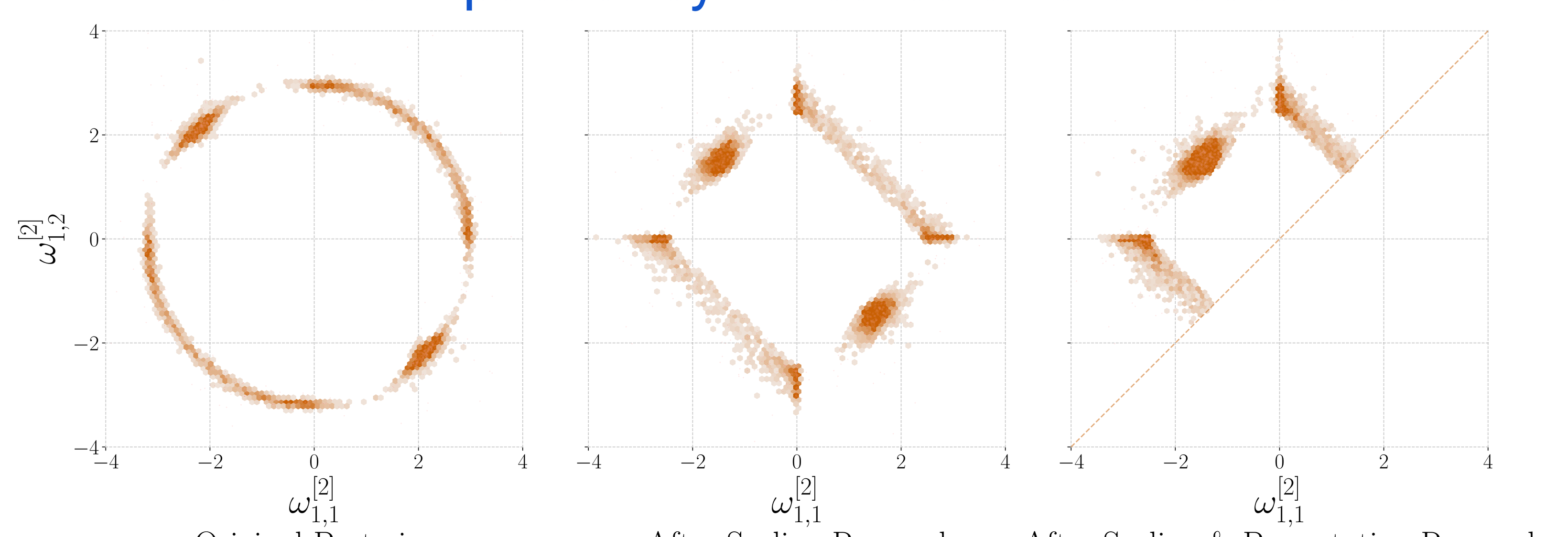
With independent and layer-wise constant initializations,

$$p(\omega_{i,j}^{[l]} | \mathcal{D}) = p(\omega_{0,j}^{[l]} | \mathcal{D}), \text{ for all i, j at layer l.}$$

→ The (possibly multivariate) posterior distribution of the weights of a given feature/channel is **constant**.

→ This corollary is not respected by most uncertainty quantification methods, including HMC (see 7-).

3 - The Impact of Symmetries on the Posterior



4 - Uncertainty, Performance & Posterior estimation

We report the performance of various methods that approx. the Bayesian posterior and compute the Maximum Mean Discrepancy (MMD) [6] with a *ground-truth posterior* of indépendent checkpoints (see 7-).

			Posterior quality			Calibration			Out-of-distribution detection		Diversity	
		Method	MMD ↓	NS ↓	Acc ↑	ECE ↓	ACE ↓	Brier ↓	AUPR ↑	FPR95 ↓	**IDMI ↓**	**OODMI ↑**
CIFAR100	One Mode	Dropout	4.5	7.5	74.2	14.7	3.2	38.8	76.4	47.7	5.7	9.1
		viBNN	9.0	10.2	57.9	24.6	3.0	63.7	60.9	79.1	2.7	4.2
		SWAG	6.7	7.2	70.9	2.3	1.2	38.9	86.2	48.0	2.4	6.3
		Laplace	5.7	7.0	75.1	**0.9**	0.9	34.6	81.3	42.4	27.6	63.3
		SGHMC	7.5	7.9	73.7	4.9	1.0	36.2	79.4	62.3	**0.2**	0.5
	Multi Mode	Dropout	0.7	4.5	**79.5**	4.3	1.0	29.2	78.2	48.1	20.5	46.3
		viBNN	6.1	5.6	66.5	2.8	2.0	45.3	71.9	71.7	45.5	81.1
		SWAG	5.0	5.4	72.8	1.5	1.1	36.9	**89.1**	50.6	6.5	19.7
		Laplace	0.6	4.3	78.9	6.9	0.8	30.3	82.9	**41.3**	44.1	**98.5**
		DE	**0.0**	**0.0**	**79.5**	1.6	**0.6**	**28.7**	81.1	45.6	22.5	58.0
TinyImageNet	One Mode	Dropout	9.5	4.9	63.2	16.4	2.4	53.9	48.8	81.1	8.3	8.4
		SWAG	9.1	3.9	66.4	10.5	0.7	46.2	61.9	57.7	3.0	4.5
		Laplace	5.5	6.1	33.1	6.0	3.6	77.1	48.8	77.7	200.7	228.0
		SGHMC	9.8	5.3	58.3	**2.6**	1.0	54.1	56.3	72.7	**0.24**	0.30
	M. Mode	Dropout	4.3	1.8	70.2	9.9	1.2	42.1	74.8	58.2	34.1	60.0
		SWAG	6.7	5.4	69.3	3.6	**0.6**	41.3	**96.5**	55.9	17.6	32.1
		Laplace	0.5	3.1	37.0	10.9	3.3	75.1	48.4	72.5	219.5	**254.7**
		DE	**0.0**	**0.0**	**70.3**	6.5	0.7	**40.9**	86.3	**50.2**	38.4	83.4

Comparison of popular methods approximating the Bayesian posterior - ResNet-18

A — Perf. & Aleatoric Uncertainty

→ Multi-mode methods obtain **better** scores in accuracy, calibration and Brier score *i.e.* **better aleatoric uncertainty estimation**.

B — Epistemic Uncertainty

→ Multi-mode methods consistently **perform better**.

→ **No** clear **correlation** with posterior quality estimation.

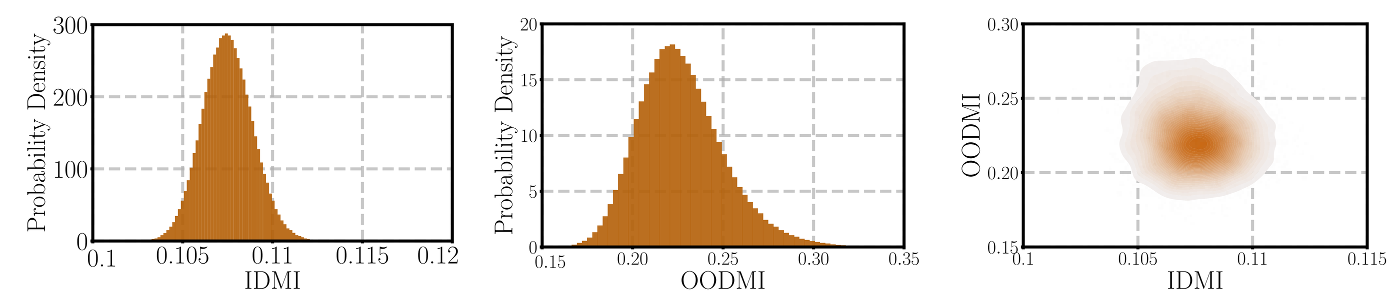
C — ID & OOD Diversity

→ Multi-mode methods exhibit **more diversity** either in- and out- of-distribution.

5 - Functional Collapse & Diversity

We take 1000 independent networks from "**Checkpoints**" and compute mutual information over the test set for each pair.

$$\mathbf{MI} = \mathcal{H}\left(\frac{1}{M}\sum_{m=1}^{M} P_{\boldsymbol{\omega}_m}(\hat{y}|\boldsymbol{x}, \mathcal{D})\right) - \frac{1}{M}\sum_{m=1}^{M} \mathcal{H}(P_{\boldsymbol{\omega}_m}(\hat{y}|\boldsymbol{x}, \mathcal{D}))$$



Pairwise mutual information - ResNet-18 - CIFAR-100/SVHN

→ Very **low** dispersion of the **in-distribution diversity**.

→ **Greater** dispersion of the **OOD diversity**.

→ **ID** & **OOD diversities** seem only very **weakly correlated**.

6 - Scaling Symmetries & Representation Cost

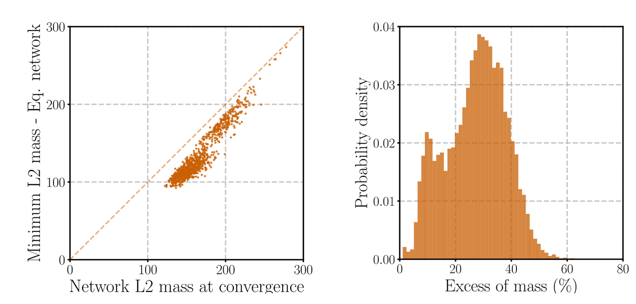
We show that the minimization of the L2 norm of the weights - the representation cost - over scaling symmetries is log-log strictly **convex**.

$$m^* = \min_{\Lambda \in \mathbb{\Lambda}} |\mathcal{T}_s(\tilde{\omega}, \Lambda)|_2^2$$

→ **Unique solution** found via convex optimization.

Gaussian-regularized training (weight decay) empirically **never** converges towards the minimal scaling coefficient for the scaled representation cost.

→ Negligible gradients vs. SGD **noise**.



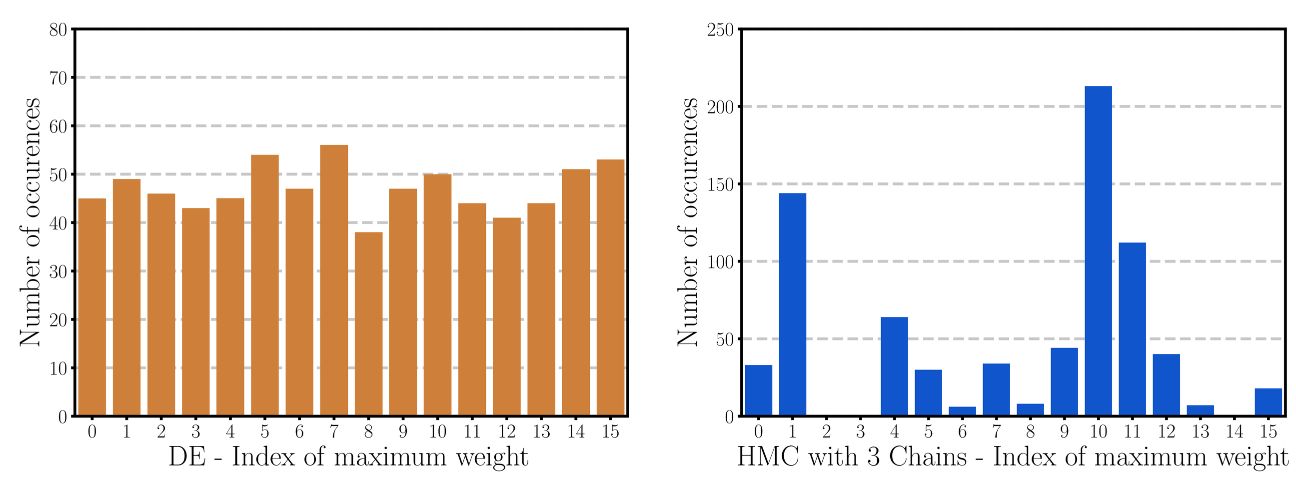
Mass profiles of simple ConvNets

⚠ Batch normalization → degenerate problem.

7 - How to Estimate the Posterior?

Multiple techniques exist to estimate the Bayesian posterior of DNNs: **variational inference Bayesian networks** [7], **Hamiltonian Monte Carlo** (HMC) [8], and its **stochastic** versions [9], **Laplace** methods, and **ensembles** [1, 4]. Izmailov et al. [10] have shown that HMC can be scaled to ResNet-20 and CIFAR-10 (albeit at very high memory expense). Although more theoretically grounded, HMC chains exhibit **autocorrelation** and their stochastic alternatives are not very reliable. Ensembles still favor local minima but we argue that it is a minor issue considering the *countless* number of minima in very high dimensions. *See the paper for a more detailed discussion.*

Corollary: $p(\omega_{i,j}^{[l]} | \mathcal{D}) = p(\omega_{0,j}^{[l]} | \mathcal{D})$, for all i, j at layer l.



→ The indices of the maximum weights of HMC are **not** uniform.

"Checkpoints" @🤗

Easy to download models & scripts:

- ❖**1,024** ResNet-20 FRN/SiLU – CIFAR-10
- ❖**2,048** ResNet-18 – CIFAR-10
- ❖**9,216** ResNet-18 – CIFAR-100
- ❖**2,048** ResNet-18 – TinyImageNet

Checkpoints

TorchUncertainty

- Open-source library for leveraging predictive **uncertainty quantification** techniques. Includes classification, regression, segmentation & pixel-wise regression.
- *viBNNs, MIMO, Masksembles, Packed-Ensembles, SWA(G), Dropout, LP-BNN, Checkpoint & Snapshot Ensembles, Temperature Scaling, MC BatchNorm, metrics, layers, losses, etc...*
- **Pre-trained weights** on Hugging Face 🤗.

TorchUncertainty

★ **STAR THE LIBRARY TO HELP US** ★

Key References

[1] Balaji Lakshminarayanan, et al. Simple and scalable predictive uncertainty estimation using DE. In NeurIPS, 2017.
[2] Wesley J. Maddox et al. A simple baseline for bayesian uncertainty in deep learning. In NeurIPS, 2019.
[3] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In ICML, 2016.
[4] Andrew G Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. In NeurIPS, 2020.
[5] Robert Hecht-Nielsen. On the algebraic structure of feedforward network weight spaces. In Advanced Neural Computers, 1990.
[6] Le Song. Learning via hilbert space embedding of distributions. University of Sydney, 2008.
[7] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In ICML, 2015.
[8] Radford M. Neal. MCMC using Hamiltonian dynamics. Handbook of markov chain monte carlo, 2011.
[9] Ruqi Zhang, et al. Cyclical stochastic gradient MCMC for bayesian deep learning. In ICLR, 2020.
[10] Pavel Izmailov et al. What are Bayesian neural network posteriors really like? In ICML, 2021.