Challenges in Bayesian Neural Networks: *Balancing Priors and Learning Biases*

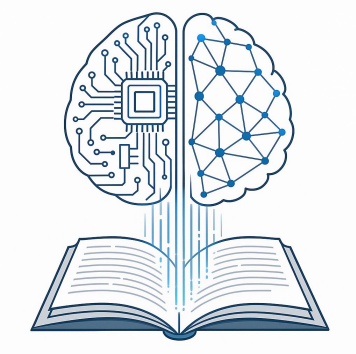
2026-03-02

Tomasz Kuśmierczyk

UNIWERSYTET
JAGIELLOŃSKI
W KRAKOWIE

Acknowledgements

https://bayes.ii.uj.edu.pl



This research is part of the project No. 2022/45/P/ST6/02969 co-funded by the National Science Centre and the European Union Framework Programme for Research and Innovation Horizon 2020 under the Marie Skłodowska-Curie grant agreement No. 945339.

NARODOWE CENTRUM NAUKI

POLONEZ BIS

Background on Bayesian learning

Bayes Theorem

$$p(A|B) = \frac{p(B|A) \cdot p(A)}{p(B)}$$

Bayes Theorem

$$p(A|B) = \frac{p(B|A) \cdot p(A)}{p(B)}$$

Let's rename $A \to \theta$, $B \to \mathcal{D}$:

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta) \cdot p(\theta)}{p(\mathcal{D})}$$

Bayesian Neural Networks

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta) \cdot p(\theta)}{p(\mathcal{D})}$$

Model Parameters as Random Variables:

- In a neural network (NN), we have weight parameters $\theta = \{w_1, w_2, \ldots, w_m\}$ (possibly millions).
- Bayesian approach: place a prior on θ and condition on data D to get the posterior $p(\theta \mid D)$.

Bayes Theorem

the standard MLE objective

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta) \cdot p(\theta)}{p(\mathcal{D})}$$

- **Prior** $p(\theta)$**:** Initial belief on parameters before seeing data
- **Likelihood** $p(\mathcal{D}|\theta)$**:** Probability of data given parameters of the model
- **Posterior** $p(\theta|\mathcal{D})$**:** Updated belief after seeing (more) data

Predictions in BNNs

- Bayesian Model Averaging (average of all possible models weighted by the posterior):

$$\underbrace{p(y|x,\mathcal{D})}_{\text{posterior predictive}} = \int \underbrace{p(y|\theta,x)}_{\text{likelihood}} \underbrace{p(\theta|\mathcal{D})}_{\text{posterior}} d\theta$$

Statistics of $p(y|x,\mathcal{D})$ can be obtained using the Monte-Carlo estimates by two-step sampling:

- sample $\theta \sim p(\theta|\mathcal{D})$
- for the fixed θ, sample $y \sim p(y|\theta,x)$

Statistics of the predictive distribution in BNNs

$$\mathbb{E}_{p(y|x,\mathcal{D})}[y] \approx \frac{1}{S_\theta}\frac{1}{S_y}\sum_{\theta\sim p(\theta|\mathcal{D})}\sum_{y\sim p(y|\theta,x)} y$$

In the same way we can get estimates of Variance, Skewness etc.

Why Bayesian? → Epistemic uncertainty

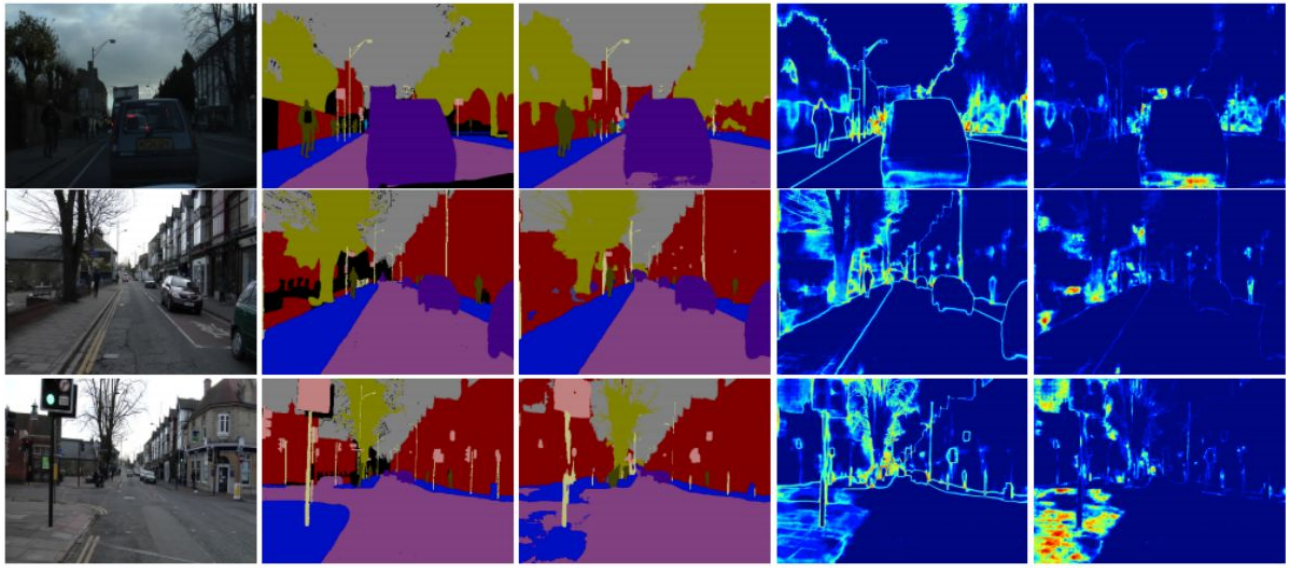
- uncertainty caused by not knowing enough yet
- can be reduced by getting more information or data

Why Bayesian? - Uncertainty decomposition

$$\mathrm{Var}(y \mid x, \mathcal{D}) = \underbrace{\mathbb{E}_{p(\theta|\mathcal{D})}[\mathrm{Var}(y \mid \theta, x)]}_{\text{Aleatoric Uncertainty}} + \underbrace{\mathrm{Var}_{p(\theta|\mathcal{D})}[\mathbb{E}(y \mid \theta, x)]}_{\text{Epistemic Uncertainty}}$$

$$H(y \mid x, \mathcal{D}) = \underbrace{\mathbb{E}_{p(\theta|\mathcal{D})}[H(y \mid \theta, x)]}_{\text{Aleatoric Uncertainty}} + \underbrace{I(y; \theta \mid \mathcal{D})}_{\text{Epistemic Uncertainty}}$$

Epistemic uncertainty

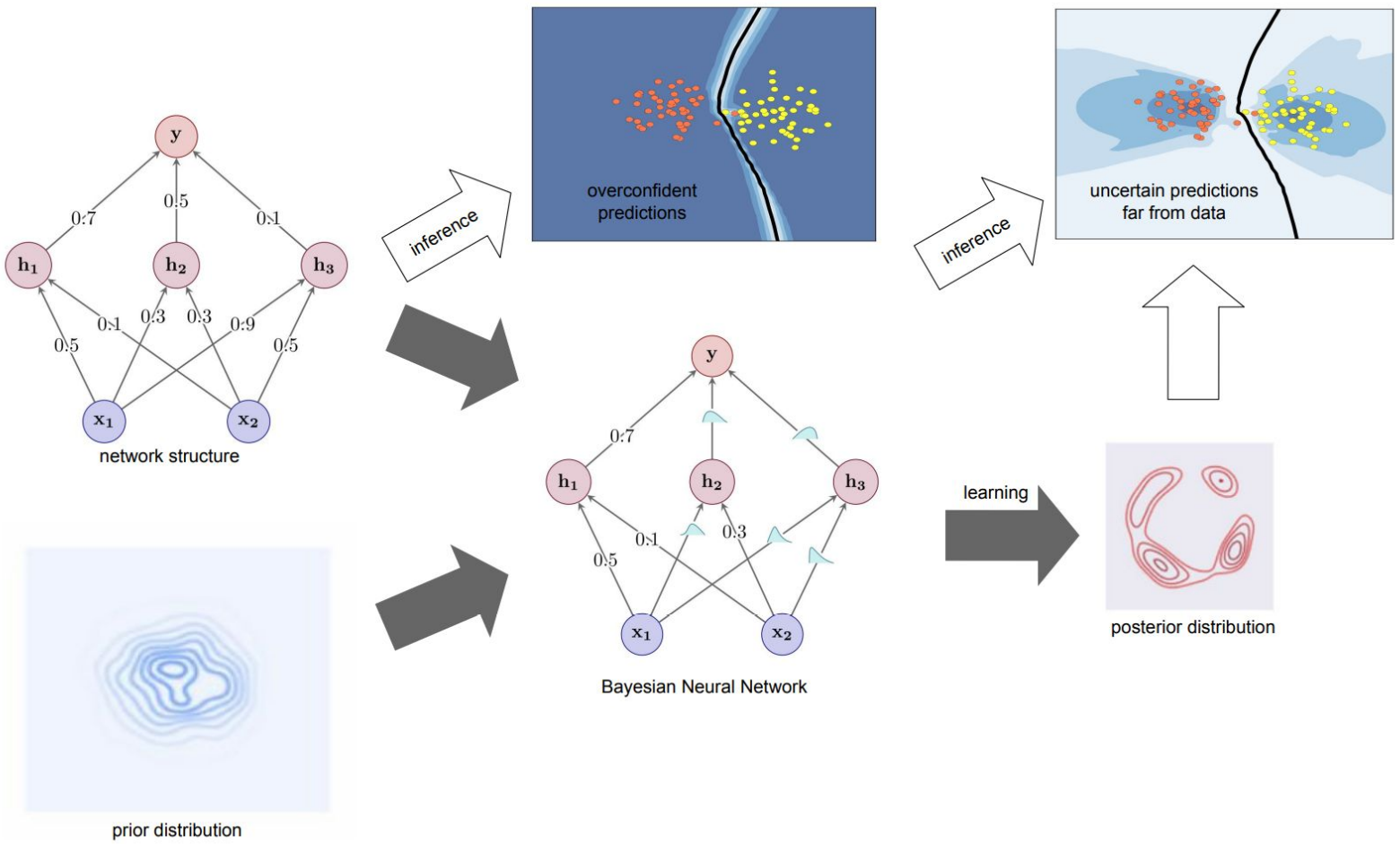


(a) Input Image (b) Ground Truth (c) Semantic Segmentation (d) Aleatoric Uncertainty (e) Epistemic Uncertainty

Figure 1: **Illustrating the difference between aleatoric and epistemic uncertainty** for semantic segmentation on the CamVid dataset [8]. *Aleatoric* uncertainty captures noise inherent in the observations. In (d) our model exhibits increased aleatoric uncertainty on object boundaries and for objects far from the camera. *Epistemic* uncertainty accounts for our ignorance about which model generated our collected data. This is a notably different measure of uncertainty and in (e) our model exhibits increased epistemic uncertainty for semantically and visually challenging pixels. The bottom row shows a failure case of the segmentation model when the model fails to segment the footpath due to increased epistemic uncertainty, but not aleatoric uncertainty.

Source: Kendall et al.: *What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?* NIPS2017

Motivation:
Prior Biases, Calibration, OOD detection, Robustness



Overview of the Challenges

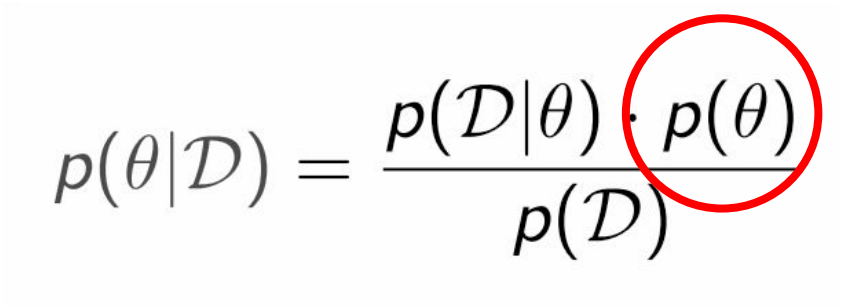
Design:

- likelihood and network structure
- priors $p(\theta)$

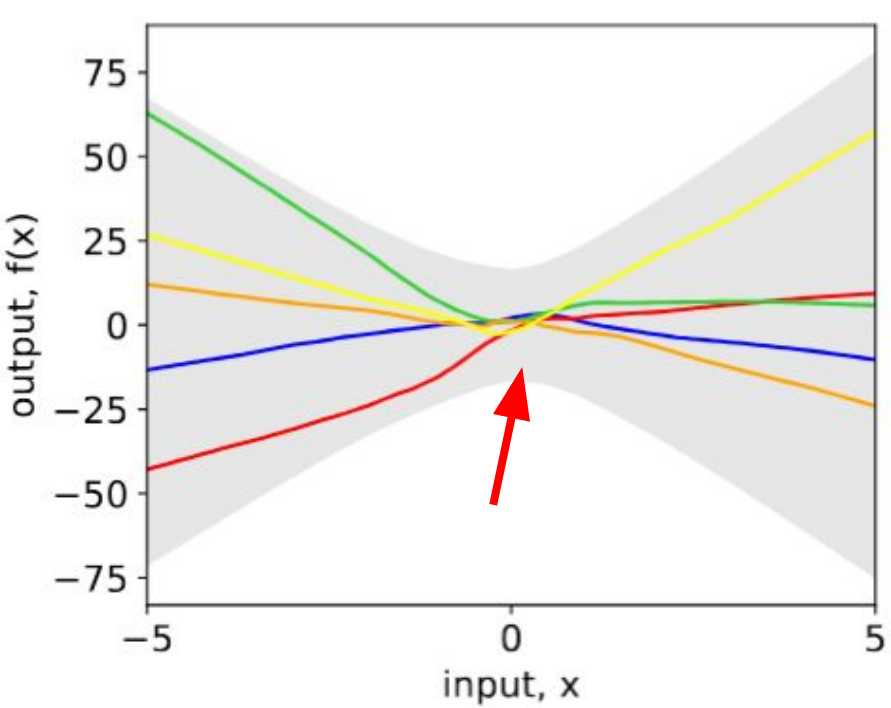
Learning:

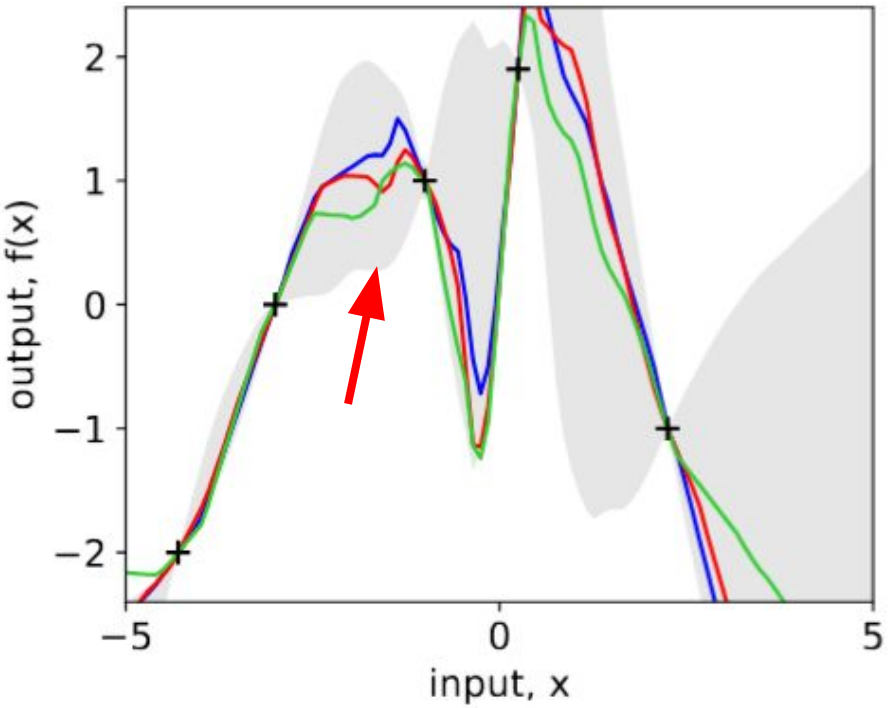
- posterior $p(\theta|\mathcal{D}) = \frac{p(\mathcal{D},\theta)}{p(\mathcal{D})}$
- evidence $p(\mathcal{D}) = \int p(\mathcal{D}|\theta)p(\theta)d\theta$
- posterior predictive $p(y|\mathcal{D}) = \int p(y|\theta)p(\theta|\mathcal{D})d\theta$
- model selection = hyperparameters

Interpretable Priors


$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta) \cdot p(\theta)}{p(\mathcal{D})}$

Motivation: Standard Gaussian priors



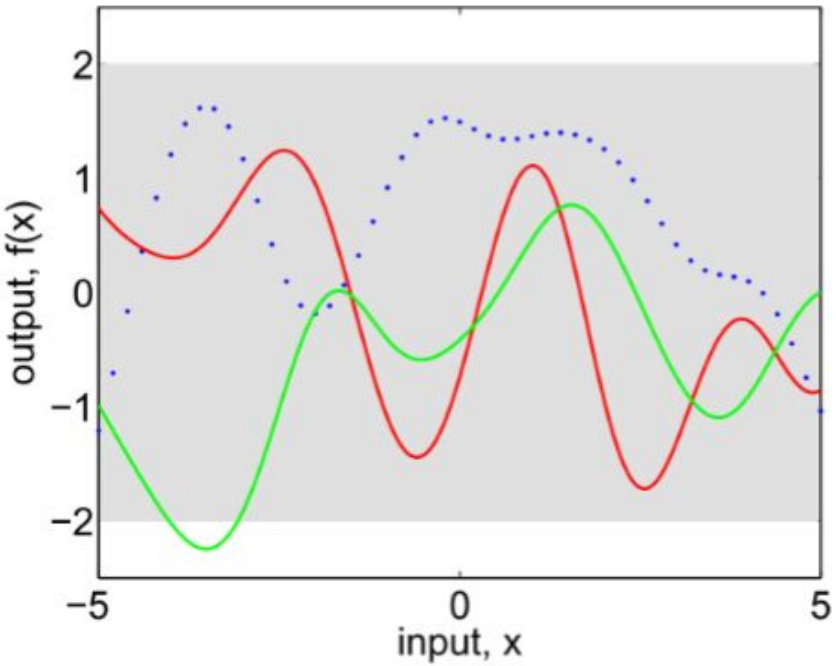


Adapted from *GPs for ML (Carl Edward Rasmussen, Christopher K. I. Williams, 2006)*

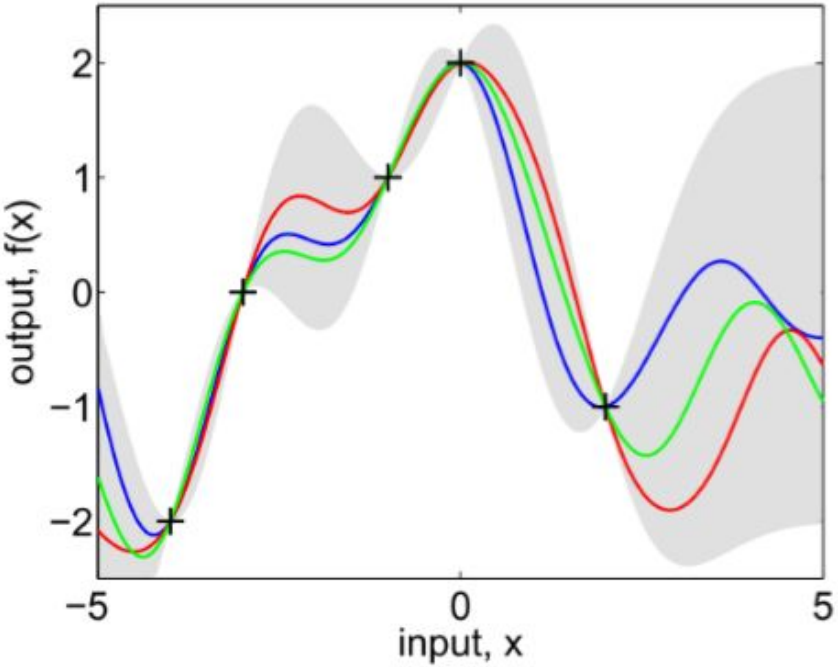
Gaussian processes:

smoothness and interpretability via kernel design

$$f(x) \sim \mathsf{GP}(m(x), \kappa(x, x'))$$



(a), prior



(b), posterior

Pretraining of BNN priors using GP reference

1. Reparameterize priors and activation:

   $p(w, b|\lambda) = N(w|0, diag(\sigma_w))N(b|0, diag(\sigma_b))$, $\phi(\cdot|\eta)$

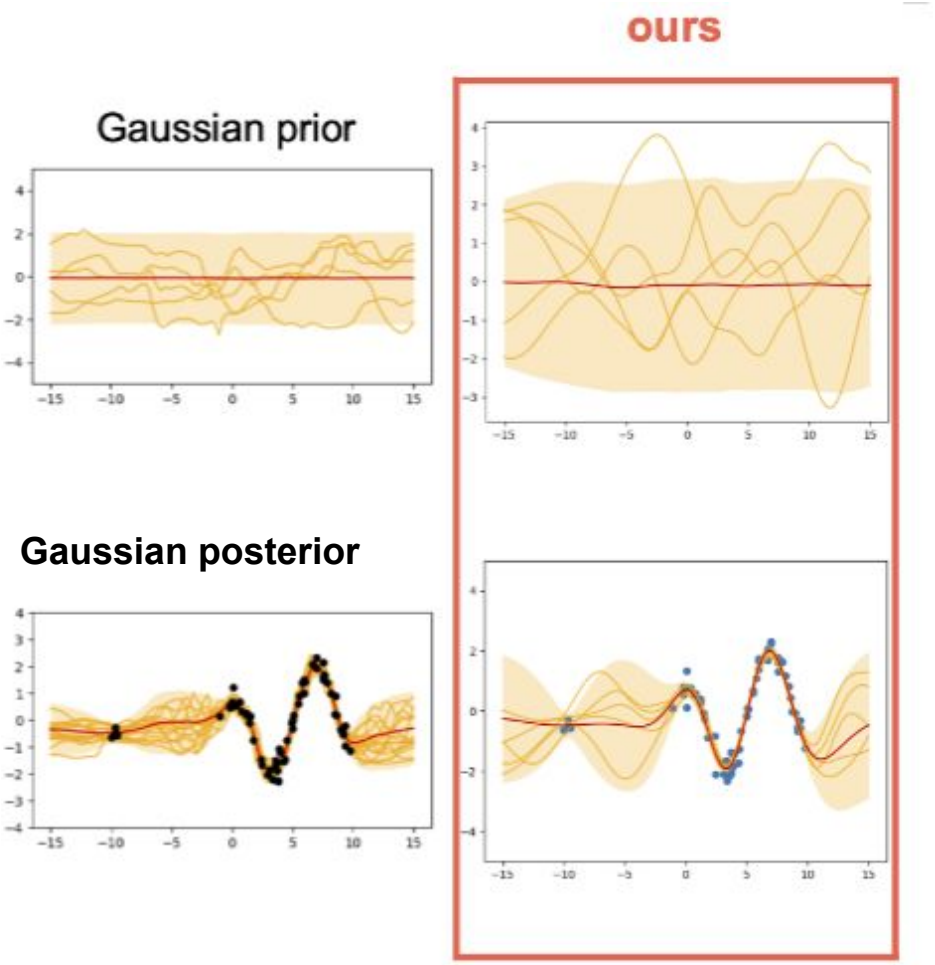
2. Optimize a distributional divergence between a GP and a BNN:

   $\lambda^* = \text{argmin}_\lambda \frac{1}{S}\Sigma_{X\sim p_X} D(p_{nn}(f^l(X|\lambda)), p_{gp}(f^l(X)))$, where $\lambda = \{\sigma_w, \sigma_b, \eta\}$

3. Use closed-form 2-Wasserstein divergence between two Gaussians:

   $D = ||\mu_1 - \mu_2||_2^2 + \text{Tr}\left(\Sigma_1 + \Sigma_2 - 2\sqrt{\sqrt{\Sigma_1}\Sigma_2\sqrt{\Sigma_1}}\right)$

BNN with pre-trained GP priors

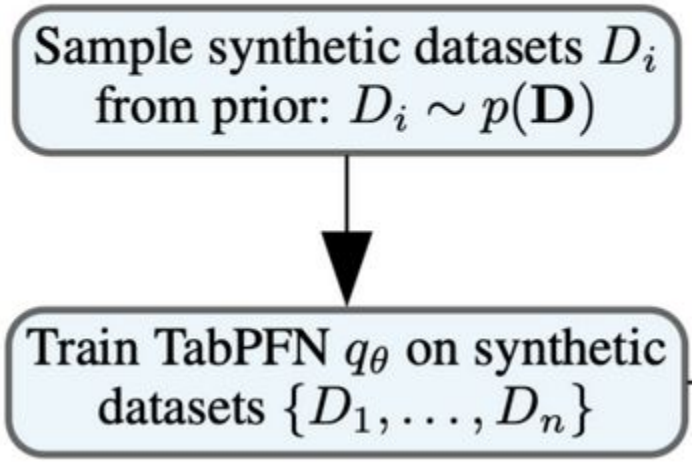


M. Sendera*, A. Sorkhei, T. Kuśmierczyk*: Revisiting the Equivalence of Bayesian Neural Networks and Gaussian Processes: On the Importance of Learning Activations. UAI 2025.

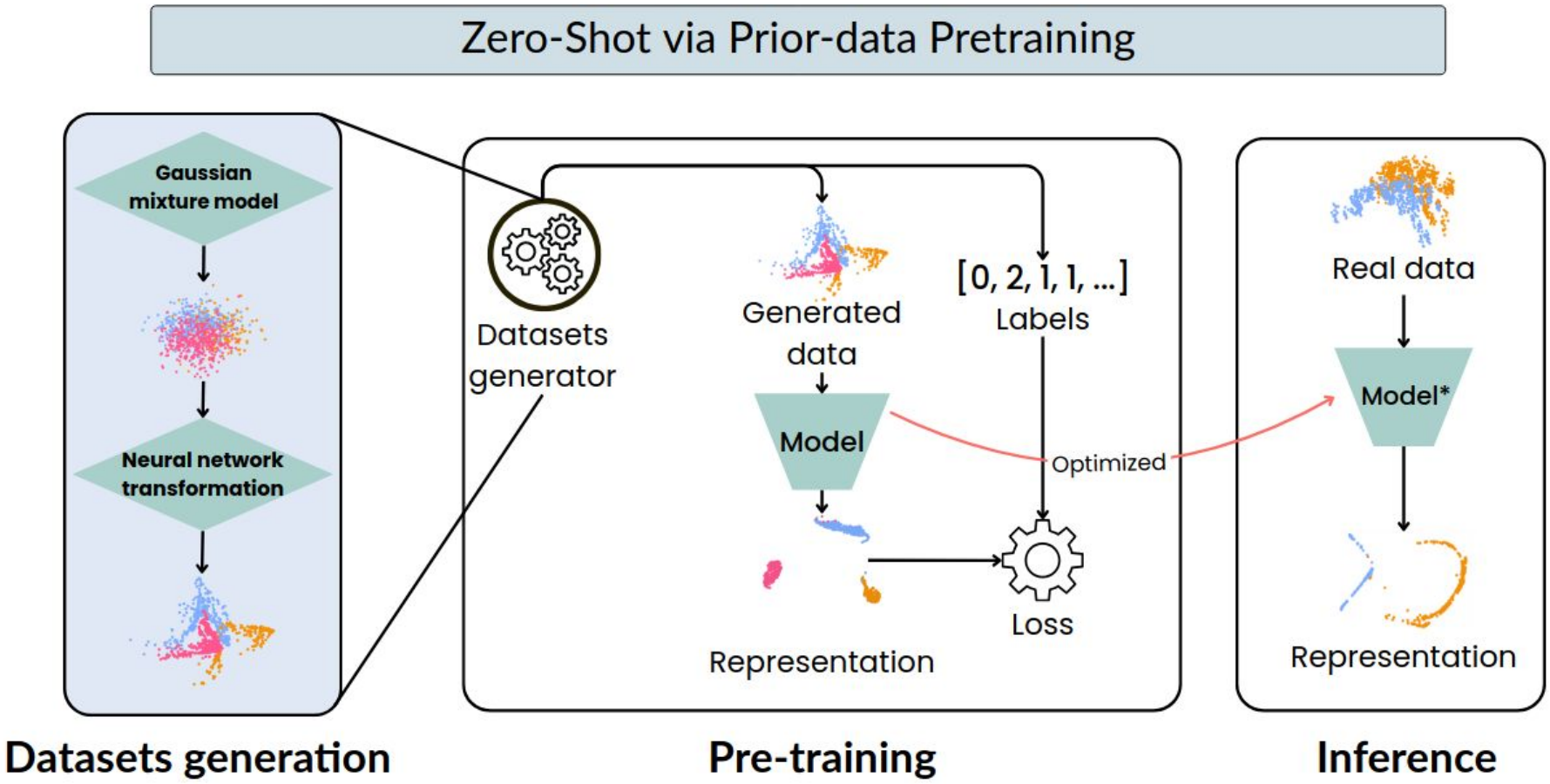
Synthetic data pre-training (for ICL): TabPFN

Create many synthetic datasets with the **desired properties**

=> pre-train the model



Transfer of priors



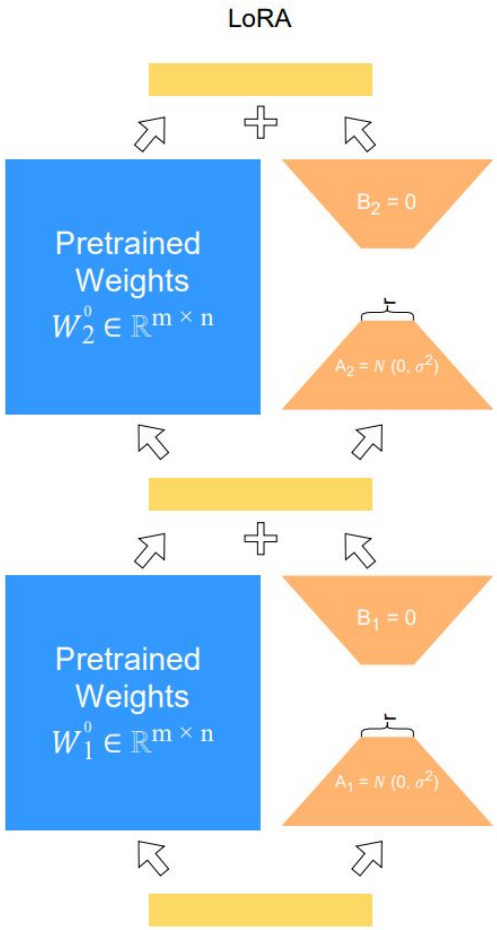
Patryk Marszałek, Tomasz Kuśmierczyk*, Witold Wydmański, Jacek Tabor, Marek Śmieja*: ZEUS: Zero-shot Embeddings for Unsupervised Separation of Tabular Data. NeurIPS 2025

Efficient Posteriors for Bayesian Fine-tuning

Selected: Posterior learning challenges

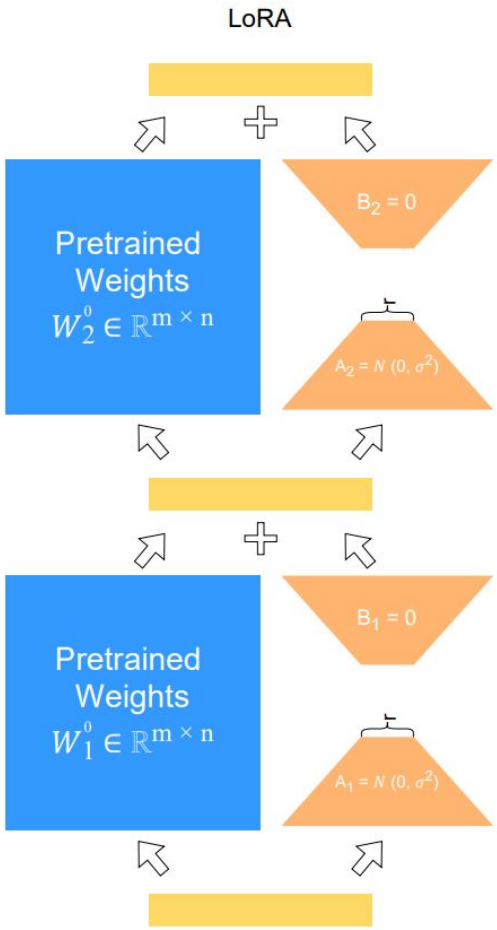
- Scalability limits for deep architectures and large datasets due to high computational and memory cost.
- Intractable exact posterior; reliance on approximations
  - ➢ even then costly

Fine-tuning of deep pre-trained models



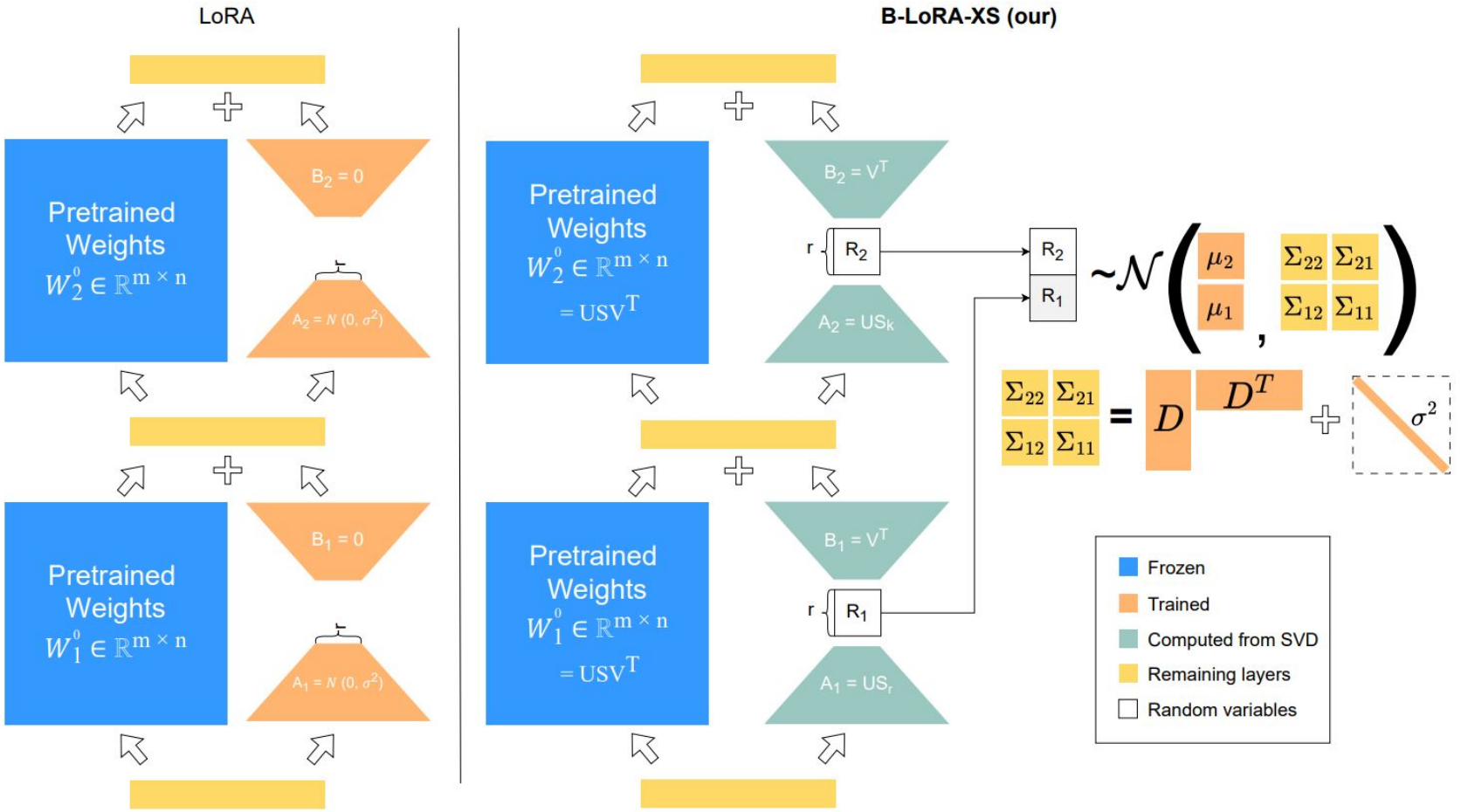
	LoRA
Standard	No Uncertainty $n \times r \cdot (d_{\text{in}} + d_{\text{out}})$

Fine-tuning of deep pre-trained models



	LoRA
Standard	No Uncertainty $n \times r \cdot (d_{\text{in}} + d_{\text{out}})$
Bayesian	Parameter Inefficient $n \times r \cdot (d_{\text{in}} + d_{\text{out}}) \cdot (k + 2)$

Learning in projected subspace



P. Marszałek*, K. Bałazy, J. Tabor, T. Kuśmierczyk*: Minimal Ranks, Maximum Confidence: Parameter-efficient Uncertainty Quantification for LoRA. EMNLP 2025

Learning in projected subspace (SVD-based variant)

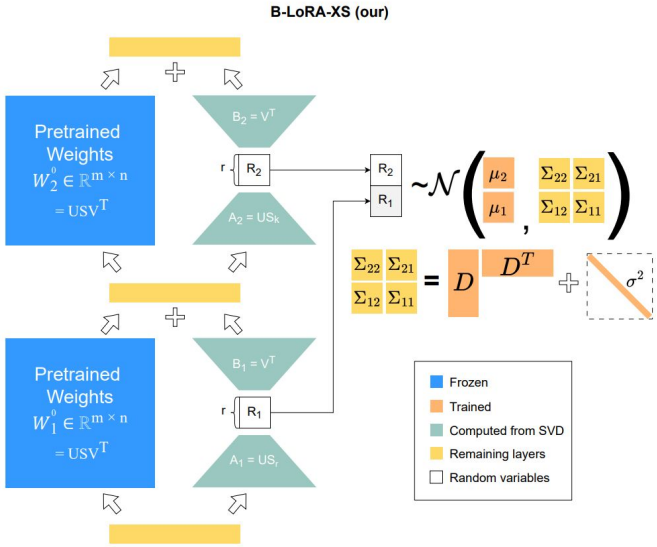
- Compute **truncated SVD of** W^0: $W^0 \approx U_r S_r V_r^T$.
- Freeze projections $A = U_r S_r,\ B = V_r^T$.
- Insert small **adapter** R:

$$\Delta W = ARB, \quad R \in \mathbb{R}^{r \times r}.$$

- Learn **Bayesian posterior** over R

$$p(R \mid D) \approx \mathcal{N}(\mu_R, \Sigma_R),$$

with Σ_R low-rank (via SWAG).



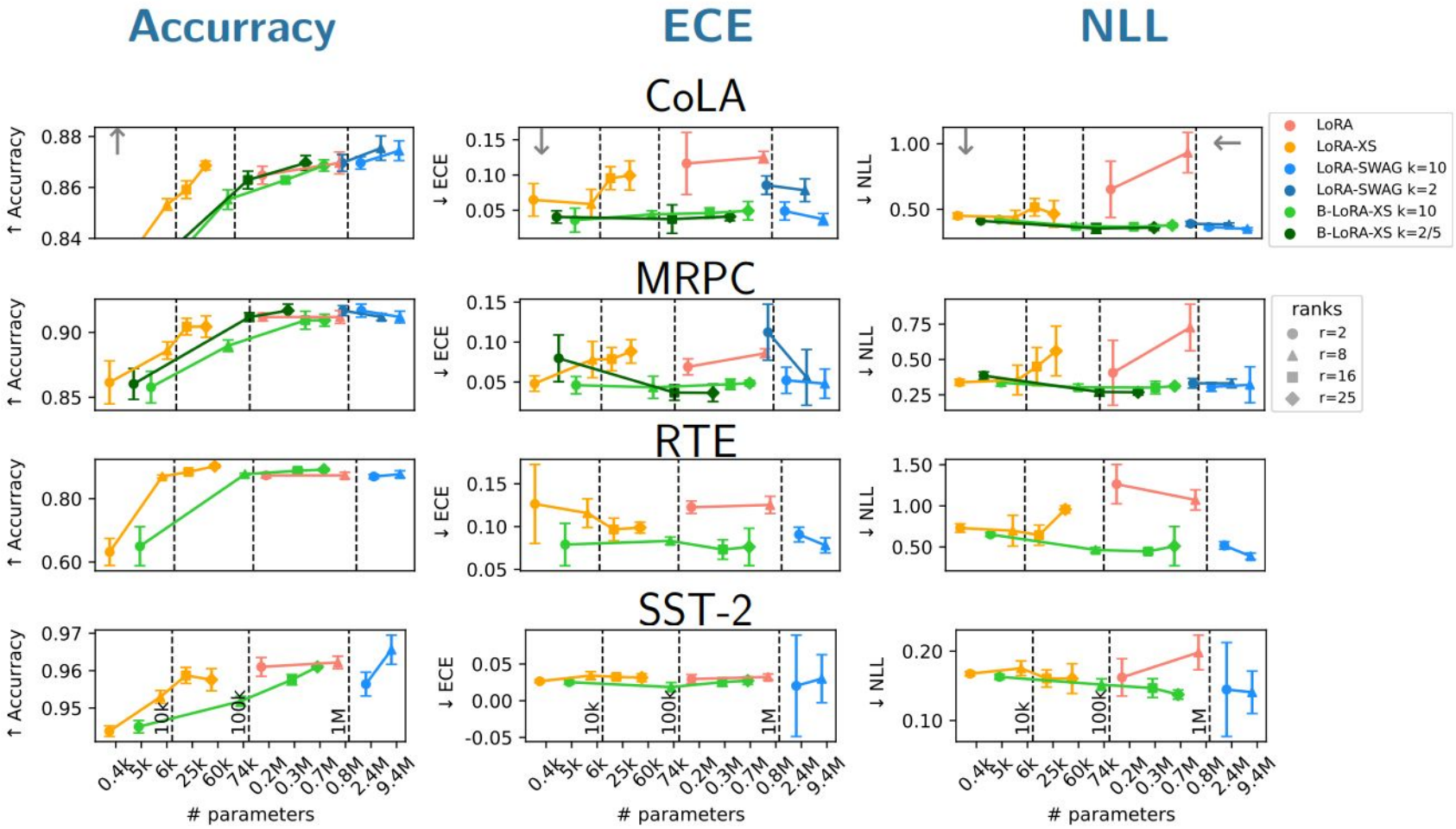
P. Marszałek*, K. Bałazy, J. Tabor, T. Kuśmierczyk*: Minimal Ranks, Maximum Confidence: Parameter-efficient Uncertainty Quantification for LoRA. EMNLP 2025

Profit

	LoRA	SVD-projected
Standard	No Uncertainty $n \times r \cdot (d_{\text{in}} + d_{\text{out}})$	No Uncertainty $n \times r^2$
Bayesian	Parameter Inefficient $n \times r \cdot (d_{\text{in}} + d_{\text{out}}) \cdot (k+2)$	B-LoRA-XS (Our) $(n \times r^2) \cdot (k+2)$

P. Marszałek*, K. Bałazy, J. Tabor, T. Kuśmierczyk*: Minimal Ranks, Maximum Confidence: Parameter-efficient Uncertainty Quantification for LoRA. EMNLP 2025

Improved calibration



P. Marszałek*, K. Bałazy, J. Tabor, T. Kuśmierczyk*: Minimal Ranks, Maximum Confidence: Parameter-efficient Uncertainty Quantification for LoRA. EMNLP 2025

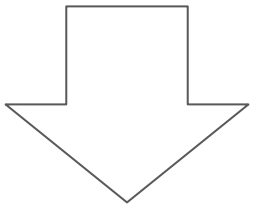
Improved OOD detection

	MRPC	
method	W_1	AUROC
B-LoRA-XS (total)	0.16	0.73
LoRA-SWAG (total)	0.05	0.44
LoRA-XS (total)	0.13	0.70
LoRA (total)	0.04	0.67
B-LoRA-XS (epistemic)	0.04	0.58
LoRA-SWAG (epistemic)	0.04	0.43

Efficient Inference

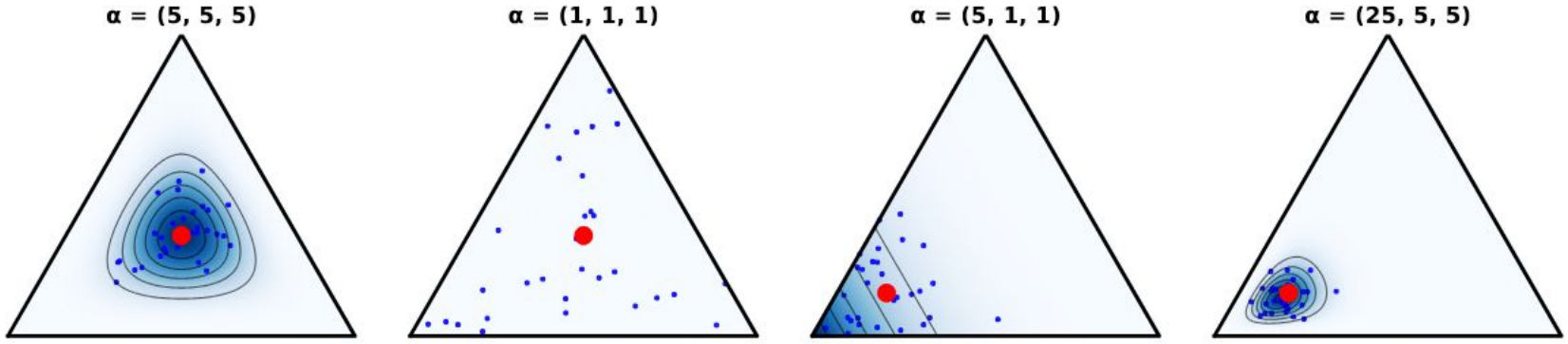
Sampling cost

$$\mathbb{E}_{p(y|x,\mathcal{D})}[y] \approx \frac{1}{S_\theta}\frac{1}{S_y}\sum_{\theta\sim p(\theta|\mathcal{D})}\sum_{y\sim p(y|\theta,x)} y$$



$$\mathbb{E}_{p(y|x,\mathcal{D})}[y] \approx \frac{1}{S_\theta}\sum_{\theta\sim p(\theta|\mathcal{D})} p(y|\theta,x)\, y$$

Predictive → Dirichlet distribution on outputs



Distillation objective

$$\mathcal{L}_{\text{Dirichlet}} = -\frac{1}{M}\sum_{i=1}^{M}\Big[\log\Gamma(\alpha_0^{(i)}) - \sum_{c=1}^{K}\log\Gamma(\alpha_c^{(i)}) + \sum_{n=1}^{N} w_n \sum_{c=1}^{K}(\alpha_c^{(i)} - 1)\log p(y = c \mid \theta_n, x^{(i)})\Big].$$

Student output parameters

Teacher samples

+ Early stopping on training data NLL

Lakshmana Sri Harsha Nemani, PK Srijith, Tomasz Kuśmierczyk: Efficient Uncertainty in LLMs through Evidential Knowledge Distillation. (submitted)

Performance does not drop

Dataset	Model	Epoch	Accuracy	ECE	NLL	Brier
Amazon	BayesPE	-	**0.959**	0.021	0.160	**0.035**
	Dirichlet	3.667	**0.958** ± 0.001	**0.011** ± 0.007	**0.132** ± 0.004	**0.035** ± 0.001
	Softmax	1.000	0.957 ± 0.000	0.013 ± 0.000	0.138 ± 0.001	**0.034** ± 0.000
SST2	BayesPE	-	**0.955**	0.029	0.165	**0.037**
	Dirichlet	4.333	**0.954** ± 0.000	**0.017** ± 0.005	**0.142** ± 0.009	**0.037** ± 0.001
	Softmax	1.333	0.952 ± 0.001	0.025 ± 0.001	0.147 ± 0.004	0.037 ± 0.000
YouTube	BayesPE	-	0.875	0.031	0.295	0.091
	Dirichlet	4.333	**0.900** ± 0.017	0.097 ± 0.023	0.294 ± 0.022	**0.079** ± 0.010
	Softmax	4.667	0.892 ± 0.001	**0.015** ± 0.006	**0.279** ± 0.006	0.084 ± 0.002
Yahoo	BayesPE	-	0.593	0.194	2.173	0.061
	Dirichlet	1.000	**0.610** ± 0.003	**0.042** ± 0.001	**1.385** ± 0.008	**0.055** ± 0.000
	Softmax	1.000	**0.609** ± 0.003	0.123 ± 0.071	1.780 ± 0.345	0.057 ± 0.002

Lakshmana Sri Harsha Nemani, PK Srijith, Tomasz Kuśmierczyk: Efficient Uncertainty in LLMs through Evidential Knowledge Distillation. (submitted)

Performance does not drop

dataset	model	Total			Epistemic			Aleatoric		
		Avg	W_1	AUROC	Avg	W_1	AUROC	Avg	W_1	AUROC
SST2	BayesPE	0.05	0.01	0.40	**0.02**	**0.04**	0.14	0.03	0.02	0.39
	Dirichlet	**0.13**	**0.07**	**0.86**	**0.02**	**0.04**	**0.50**	**0.11**	**0.06**	**0.85**
	Softmax	0.05	0.01	0.56	-	-	-	-	-	-
YouTube	BayesPE	0.25	0.19	0.86	**0.08**	0.05	0.67	0.17	0.12	0.84
	Dirichlet	**0.60**	**0.55**	**0.98**	0.03	**0.06**	**0.77**	**0.57**	**0.52**	**0.97**
	Softmax	-	0.22	0.89	-	-	-	-	-	-
Yahoo	BayesPE	0.52	0.47	0.90	**0.13**	**0.09**	0.75	0.40	0.34	0.89
	Dirichlet	**2.16**	**2.10**	**1.00**	0.08	**0.09**	**0.86**	**2.07**	**2.02**	**1.00**
	Softmax	-	0.43	0.83	-	-	-	-	-	-

Lakshmana Sri Harsha Nemani, PK Srijith, Tomasz Kuśmierczyk: Efficient Uncertainty in LLMs through Evidential Knowledge Distillation. (submitted)

Questions?