BLoB: Bayesian Low-Rank Adaptation by Backpropagation for Large Language Models

Yibin Wang*[1], Haizhou Shi*[1], Ligong Han[1 2], Dimitris N. Metaxas[1], Hao Wang[1]

*Equal Contribution

[1]Rutgers University [2]MIT-IBM Watson AI Lab

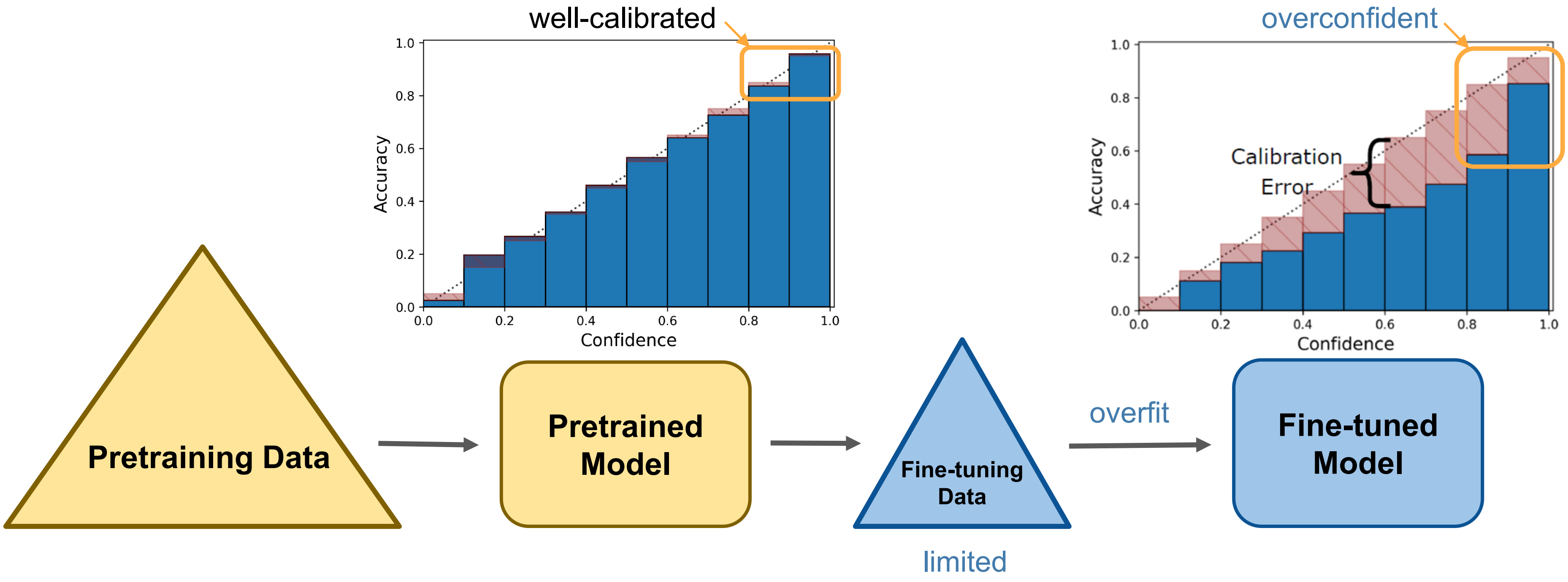
NeurIPS 2024

RUTGERS

MIT-IBM Watson AI Lab.

Motivation

Accurately estimating response confidence (or uncertainty) is crucial to trustworthy LLMs.



Motivation

- Accurately estimating response confidence (or uncertainty) is crucial to trustworthy LLMs.
- Bayesian neural networks provide a natural way to estimate uncertainty and calibrate model, especially in a data-limited scenario.

$$\underbrace{P(\boldsymbol{y}|\boldsymbol{x},\mathcal{D})}_{\text{predictive uncertainty}} = \int P(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{W})\underbrace{P(\boldsymbol{W}|\mathcal{D})}_{\text{posterior distribution}}d\boldsymbol{W}$$

Variational Bayesian Networks approximate the true posterior using a variational distribution.

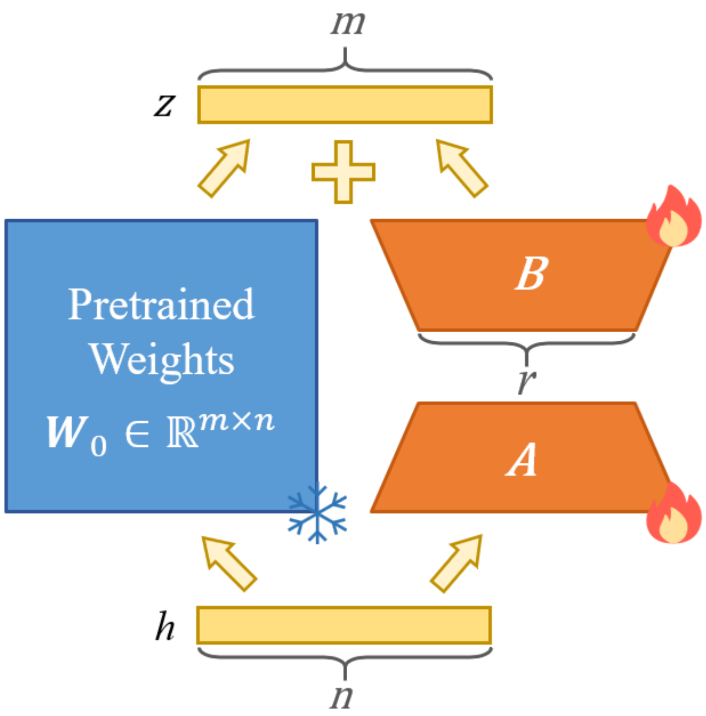
$$\underbrace{q(\boldsymbol{W}|\boldsymbol{\theta})}_{\text{variational distribution}} \xrightarrow{\text{approximate}} \underbrace{P(\boldsymbol{W}|\mathcal{D})}_{\text{true posterior distribution}}$$

- However, introducing additional trainable parameters $\boldsymbol{\theta}$ is impractical for large models.
- Parameter-Efficient Fine-Tuning (PEFT) can significantly relieve the burden.

Combining Bayesian Neural Networks and PEFT

- **Low-Rank Adaptation (LoRA)[1]**

LoRA decomposes each update matrix $\Delta\mathbf{W} \in \mathbf{R}^{m \times n}$ into the product of two low-rank matrices $\mathbf{B}$ and $\mathbf{A}$, where $\mathbf{B} \in \mathbf{R}^{m \times r}$ and $\mathbf{A} \in \mathbf{R}^{r \times n}$. ($r \ll \min\{m, n\}$)



[1] Hu, Edward J., et al. "LoRA: Low-Rank Adaptation of Large Language Models." *International Conference on Learning Representations.*

Combining Bayesian Neural Networks and PEFT

- **Bayes By Backprop (BBB)**

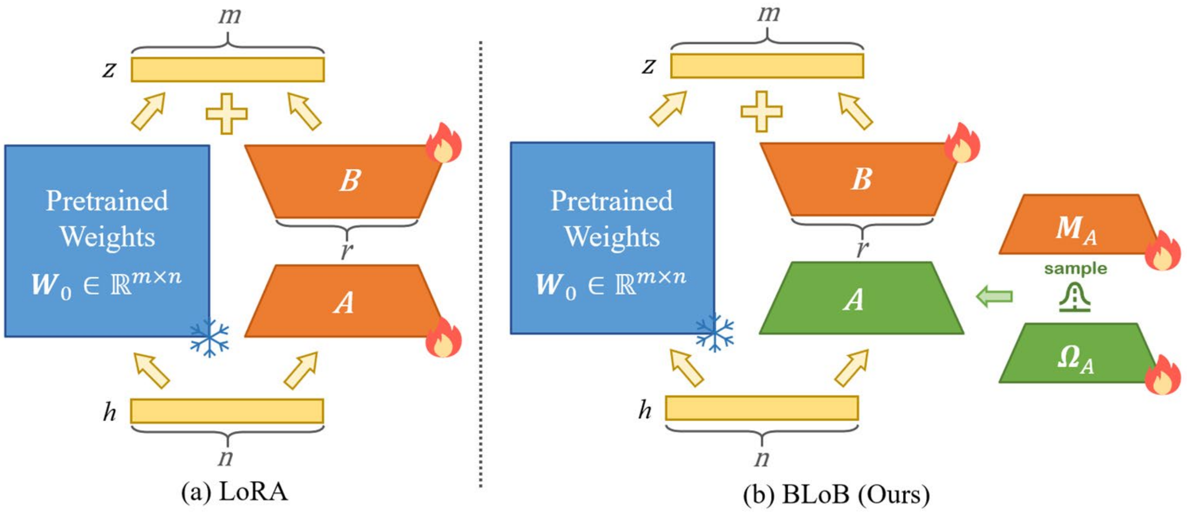
Bayes By Backprop (BBB)[2] parameterizes the variational distribution $q(\mathbf{W}|\boldsymbol{\theta})$ as a diagonal Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}_q^2)$, and minimizes the following variational free energy:

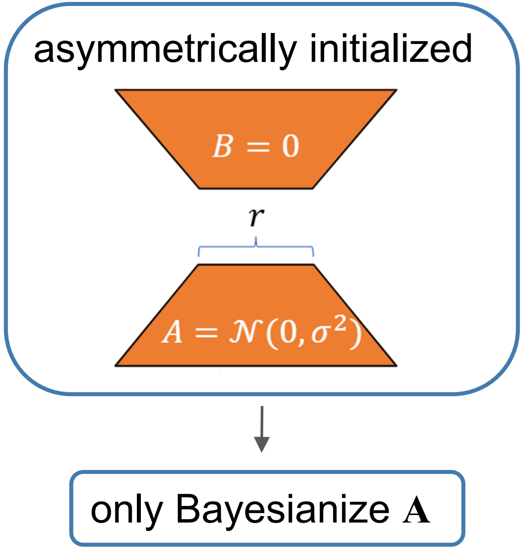
$$\mathcal{F}(\mathcal{D}, \boldsymbol{\theta}) \approx \underbrace{-\frac{1}{K}\sum_{k=1}^{K} \log P(\mathcal{D}|\boldsymbol{W}_k)}_{\text{data likelihood}} + \underbrace{\frac{1}{K}\sum_{k=1}^{K} [\log q(\boldsymbol{W}_k|\boldsymbol{\theta}) - \log P(\boldsymbol{W}_k)]}_{\text{equivalent to minimize } \mathrm{KL}[q(\boldsymbol{W}|\boldsymbol{\theta}) \,\|\, P(\boldsymbol{W})]},$$

[2] Blundell, Charles, et al. "Weight uncertainty in neural network." *International conference on machine learning*. PMLR, 2015.

Bayesian Low-Rank Adaptation by Backpropagation (BLoB)

- **Asymmetric LoRA Bayesianization**





- reduce sampling noise & improve convergence speed
- reduce additional memory cost by 50%
- is equivalent to finding a posterior estimate for the full-weight matrix with a low-rank structure

$$q(\boldsymbol{A}|\boldsymbol{\theta} = \{\boldsymbol{M}, \boldsymbol{\Omega}\}) = \prod_{ij} \mathcal{N}(A_{ij}|M_{ij}, \Omega_{ij}^2) \qquad q(\mathrm{vec}(\boldsymbol{W})|\boldsymbol{B}, \boldsymbol{\theta}) = \mathcal{N}(\mathrm{vec}(\boldsymbol{W})|\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$$

Bayesian Low-Rank Adaptation by Backpropagation (BLoB)

- **Asymmetric LoRA Bayesianization: From Posterior to Prior**

We assume the prior distribution to be a low-rank Gaussian, with its covariance matrix parameterized by a rank-r' matrix $\widetilde{\boldsymbol{R}} \in \mathbf{R}^{(mn)\times r'}$

$$P(\text{vec}(\boldsymbol{W})) = \mathcal{N}(\text{vec}(\boldsymbol{W})|\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p),$$
$$\text{where} \quad \boldsymbol{\mu}_p = \text{vec}(\boldsymbol{W}_0),$$
$$\boldsymbol{\Sigma}_p = \widetilde{\boldsymbol{R}}\widetilde{\boldsymbol{R}}^\top.$$

Then we can optimize the KL divergence in the low-rank space, with the Gaussian prior distribution $P(\boldsymbol{A}) = \prod_{ij} \mathcal{N}(A_{ij}|0, \sigma_p^2)$

$$\text{KL}[q(\text{vec}(\boldsymbol{W})|\boldsymbol{B}, \boldsymbol{\theta})\|P(\text{vec}(\boldsymbol{W}))] = \text{KL}[q(\boldsymbol{A}|\boldsymbol{\theta})\|P(\boldsymbol{A})],$$

if $\widetilde{\boldsymbol{R}} = [\sigma_p \boldsymbol{I}_n \otimes \boldsymbol{R}]$, *where* $\boldsymbol{R}$ *satisfies* $\boldsymbol{R}\boldsymbol{R}^\top = \boldsymbol{B}\boldsymbol{B}^\top$.

Bayesian Low-Rank Adaptation by Backpropagation (BLoB)

- **BLoB: Final Algorithm**

$$\mathcal{F}(\mathcal{D}, \boldsymbol{B}, \boldsymbol{\theta}) = -\mathbb{E}_{q(\boldsymbol{A}|\boldsymbol{\theta})}[\log P(\mathcal{D}|\boldsymbol{A}, \boldsymbol{B})] + \mathrm{KL}[q(\boldsymbol{A}|\boldsymbol{\theta}) \parallel P(\boldsymbol{A})]$$

Training

$$\mathbb{E}_{q(\boldsymbol{W}|\boldsymbol{\theta})}[P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{W})] \approx \frac{1}{N}\sum_{n=1}^{N} P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{W}_n), \quad \boldsymbol{W}_n \sim q(\boldsymbol{W}|\boldsymbol{\theta}).$$

Inference

Experimental Result

$$\mathbb{E}_{q(\boldsymbol{W}|\boldsymbol{\theta})}[P(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{W})] \approx \frac{1}{N}\sum_{n=1}^{N} P(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{W}_n), \quad \boldsymbol{W}_n \sim q(\boldsymbol{W}|\boldsymbol{\theta}).$$

Metric	Method	Datasets					
		WG-S [82]	ARC-C [18]	ARC-E [18]	WG-M [82]	OBQA [65]	BoolQ [17]
ACC (↑)	MLE	68.99±0.58	69.10±2.84	85.65±0.92	74.53±0.66	81.52±0.25	86.53±0.28
	MAP	68.62±0.71	67.59±0.40	86.55±0.55	75.61±0.71	81.38±0.65	86.50±0.41
	MCD [29]	69.46±0.62	68.69±1.30	86.21±0.46	**76.45±0.04**	81.72±0.10	87.29±0.13
	ENS [51, 8, 103]	69.57±0.66	66.20±2.01	84.40±0.81	75.32±0.21	81.38±0.91	87.09±0.11
	BBB [11]	56.54±7.87	68.13±1.27	85.86±0.74	73.63±2.44	82.06±0.59	**87.21±0.22**
	LAP [116]	69.20±1.50	66.78±0.69[1]	80.05±0.22	75.55±0.36	82.12±0.67	86.95±0.09
	BLoB (N=0)	**70.89±0.82**	**70.83±1.57**	**86.68±0.60**	74.55±1.94	**82.73±0.41**	86.80±0.23
	BLoB (N=5)	66.30±0.62	67.34±1.15	84.74±0.33	72.89±1.25	81.79±0.94	86.47±0.15
	BLoB (N=10)	69.07±0.34	68.81±1.09	85.56±0.35	73.69±0.17	81.52±0.74	86.99±0.24
ECE (↓)	MLE	29.83±0.58	29.00±1.97	13.12±1.39	20.62±0.74	12.55±0.46	3.18±0.09
	MAP	29.76±0.87	29.42±0.68	12.07±0.55	23.07±0.14	13.26±0.82	3.16±0.23
	MCD [29]	27.98±0.44	27.53±0.80	12.20±0.56	19.55±0.47	13.10±0.11	3.46±0.16
	ENS [51, 8, 103]	28.52±0.55	29.16±2.37	12.57±0.58	20.86±0.43	15.34±0.27	9.61±0.24
	BBB [11]	21.81±12.95	26.23±1.47	12.28±0.58	15.76±4.71	11.38±1.07	3.74±0.10
	LAP [116]	**4.15±1.12**	16.25±2.61[1]	33.29±0.57	7.40±0.27	8.70±1.77	**1.30±0.33**
	BLoB (N=0)	20.62±0.83	20.61±1.16	9.43±0.38	11.23±0.69	8.36±0.38	2.46±0.07
	BLoB (N=5)	10.89±0.83	11.22±0.35	6.16±0.23	4.51±0.35	**3.40±0.57**	1.63±0.35
	BLoB (N=10)	9.35±1.37	**9.59±1.88**	**3.64±0.53**	**3.01±0.12**	3.77±1.47	1.41±0.19
NLL (↓)	MLE	3.17±0.37	2.85±0.27	1.17±0.13	0.95±0.07	0.73±0.03	0.32±0.00
	MAP	2.46±0.34	2.66±0.11	0.90±0.05	1.62±0.29	0.75±0.01	0.33±0.00
	MCD [29]	2.79±0.53	2.67±0.15	1.00±0.14	1.02±0.03	0.77±0.03	0.31±0.00
	ENS [51, 8, 103]	2.71±0.08	2.46±0.22	0.82±0.03	1.25±0.03	1.06±0.04	0.57±0.02
	BBB [11]	1.40±0.55	2.23±0.04	0.91±0.06	0.84±0.15	0.66±0.05	**0.31±0.00**
	LAP [116]	**0.60±0.00**	1.03±0.04[1]	0.88±0.00	0.57±0.01	0.52±0.01	**0.31±0.00**
	BLoB (N=0)	0.91±0.10	1.19±0.02	0.56±0.01	0.60±0.01	0.56±0.02	0.32±0.00
	BLoB (N=5)	0.68±0.01	0.90±0.01	0.46±0.02	0.56±0.01	0.53±0.01	0.32±0.00
	BLoB (N=10)	0.63±0.01	**0.78±0.02**	**0.40±0.01**	**0.54±0.00**	**0.50±0.01**	**0.31±0.00**

N = 10

- best uncertainty estimation performance

N = 0

- only use the mean of variational distribution
- best accuracy at the expense of calibration