FY2024/2024年度

Approximate Bayesian Inference Team
Mohammad Emtiyaz Khan
近似ベイズ推論チーム

RIKEN Center for Advanced Intelligence Project

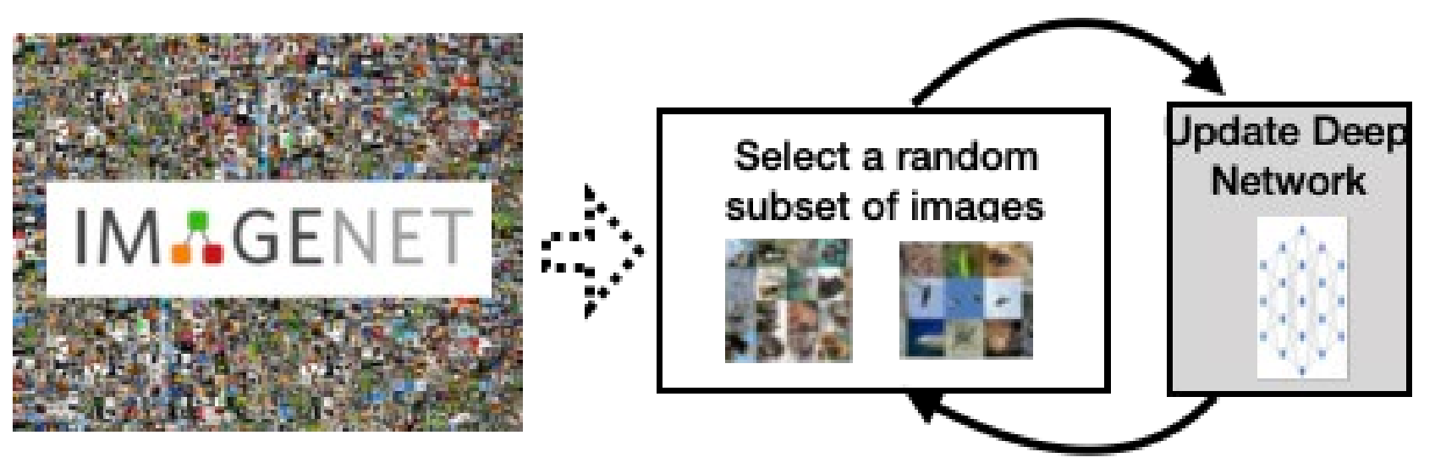
Overview and Goals

Goal: to design low-cost AI systems that can learn and improve continually throughout their lives, just like humans and animals. Currently, deep learning requires a large amount of data which is costly, rigid, and cannot quickly adapt. We aim to fix this with a new principle which Bayes-duality.

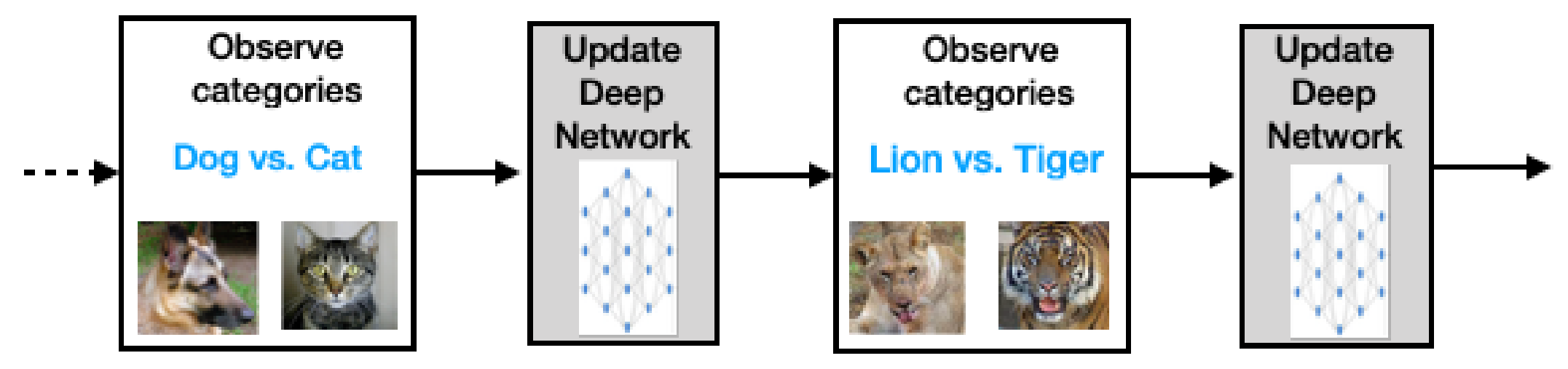
Summary for the year 2024 (paper number shown in green boxes)

1. We obtain state-of-the-art result at GPT-2 level with a new variational algorithm called Improved Variational Online Newton (IVON).
2. IVON also improves performance for LoRA fintetuning over standard methods such as AdamW.
3. We propose new uncertainty-based methods to understand and improve model-merging in Large Language Models (LLMs).
4. We propose new conformal prediction methods to address challenging cases, such as, heteroscedastic, multimodal, or skewed distributions.
5. We participated in a position paper to emphasize the importance of Bayesian methods for large-scale AI based on deep learning.

Standard Deep Learning

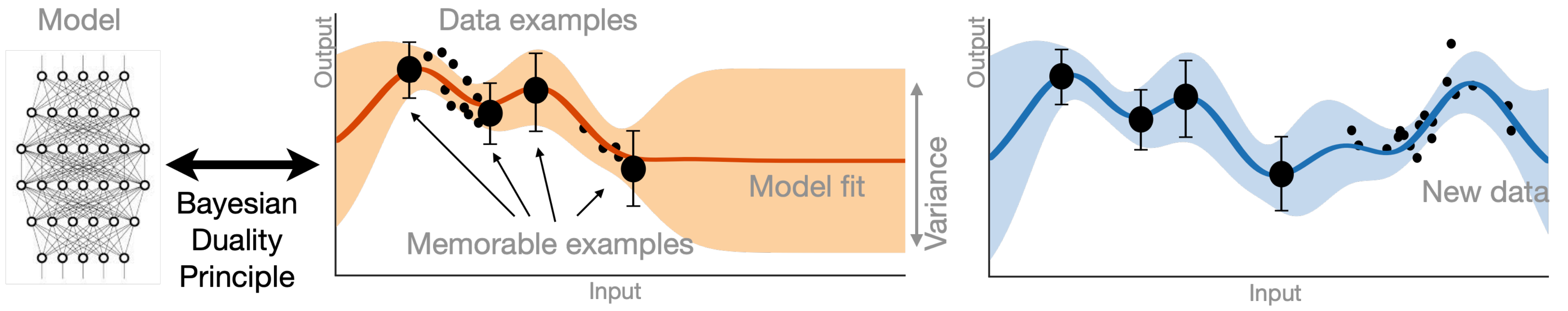


Continual Lifelong Learning



Bayes-duality relates *models parameters (left)* to the *data examples* (small dots at the right). The principle enables us to identify a few *memorable examples* (big black circle).

The memorable examples can be reused later during training with the new data. This avoids forgetting of the past.



IVON

Problem & Contribution [1]: Existing variational-learning algorithms do not work as well as Adam at large scale (e.g., GPT level) while also keeping the cost the same. We propose IVON that fixes this issue and obtains state-of-the-art results on GPT-2.

RMSprop

```
$\hat{g} \leftarrow \hat{\nabla}\ell(\theta)$
$\hat{h} \leftarrow \hat{g}^2$
$h \leftarrow (1-\rho)h + \rho\hat{h}$
$\theta \leftarrow \theta - \alpha(\hat{g} + \delta m)/(\sqrt{h} + \delta)$
```

IVON

```
$\hat{g} \leftarrow \hat{\nabla}\ell(\theta)$ where $\theta \sim \mathcal{N}(m, \sigma^2)$
$\hat{h} \leftarrow \hat{g} \cdot (\theta - m)/\sigma^2$
$h \leftarrow (1-\rho)h + \rho\hat{h} \; + \rho^2(h - \hat{h})^2/(2(h+\delta))$
$m \leftarrow m - \alpha(\hat{g} + \delta m)/(h + \delta)$
$\sigma^2 \leftarrow 1/(N(h + \delta))$
```

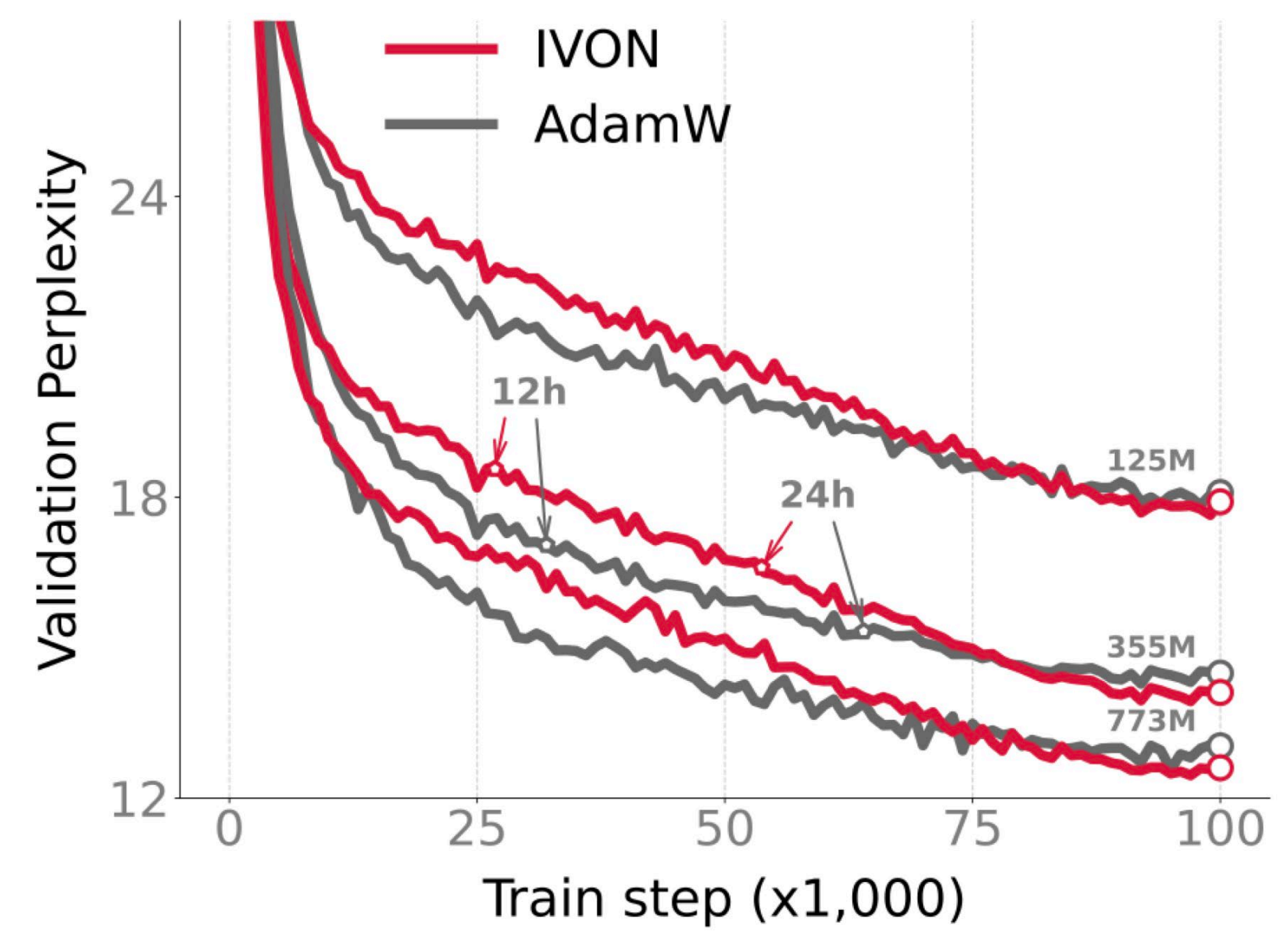
Plug-and-Play code

PyTorch

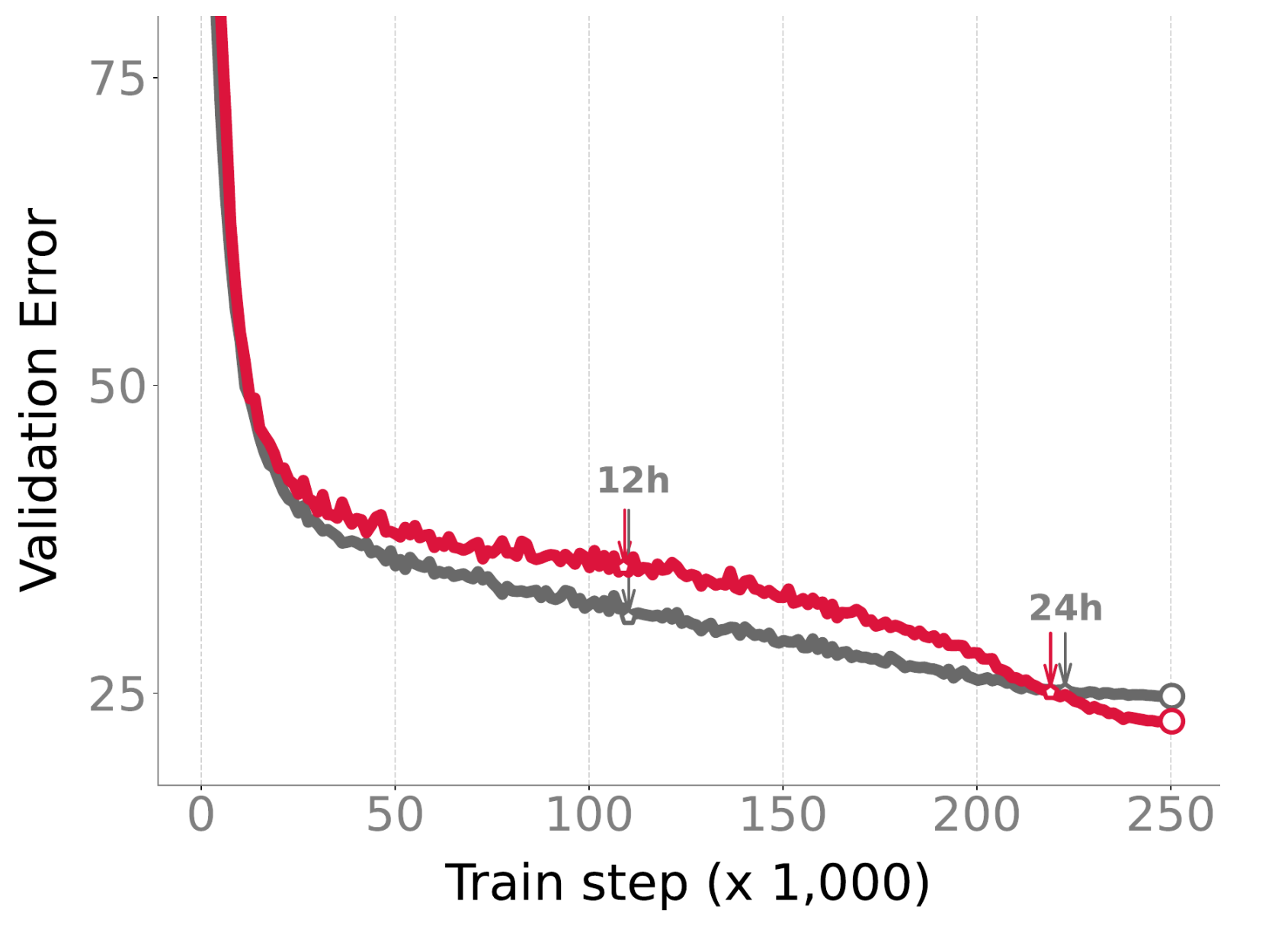
Code (just 2 lines modification)

```
for inputs, targets in dataloader:
 for _ in range(num_mc_samples):
  with optimizer.sampled_params(train=True):
   optimizer.zero_grad()
   outputs = model(inputs)
   loss = loss_fn(outputs, targets)
   loss.backward()
 optimizer.step()
```

IVON on GPT-2 (better perplexity than AdamW)



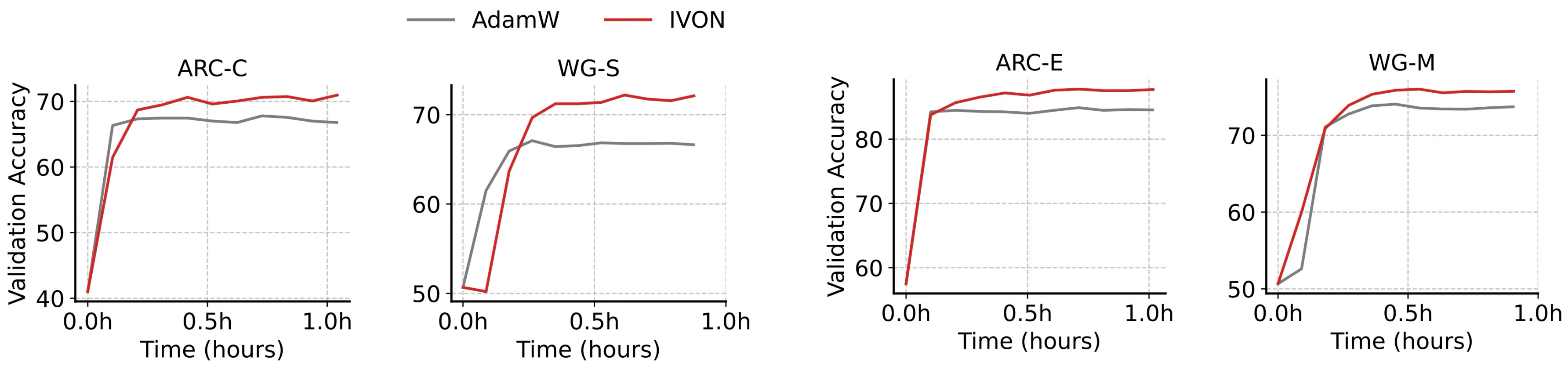
IVON on ImageNet (better accuracy)



IVON obtained first position on NeurIPS 2023 challenge (the team won $3000)

Rank	Method	CIFAR-10		MedMNIST		UCI
		Agree ↑	TVD ↓	Agree ↑	TVD ↓	W_2 ↓
1	Multi-IVON[†]	**78.7%**	**0.198**	88.4%	0.099	**0.094**
2	Multi-SWAG	77.8%	0.219	**89.0%**	**0.098**	0.166
3	SAE	77.3%	0.210	87.5%	0.107	0.116
	Multi-IVON (Alg. 1)	78.2%	0.204	89.1%	0.097	0.075

IVON works well for LoRA finetuning on Llama 2 (7 billion parameters)



Metrics	Methods	WG-S	ARC-C	ARC-E	WG-M	OBQA	BoolQ	Average
ACC ↑	AdamW*	$66.5_{0.4}$	$66.7_{0.5}$	$84.9_{0.2}$	$73.5_{0.4}$	$78.9_{0.7}$	$85.8_{0.1}$	76.1
	+ MC Drop	$66.7_{0.4}$	$67.3_{0.5}$	$84.8_{0.4}$	$73.7_{0.2}$	$79.3_{0.5}$	$85.9_{0.2}$	76.3
	+ LA (KFAC)	$66.6_{0.3}$	$66.0_{1.4}$	$84.3_{0.4}$	$73.2_{0.3}$	$78.6_{0.9}$	$85.7_{0.2}$	75.7
	+ LA (diag)	$66.2_{0.3}$	$61.2_{1.9}$	$81.8_{0.5}$	$73.3_{0.3}$	$79.7_{0.8}$	$85.7_{0.2}$	74.7
	+ SWA*	$69.7_{0.6}$	$67.2_{1.3}$	$85.2_{0.1}$	$75.6_{0.2}$	$79.8_{0.5}$	$85.5_{0.1}$	77.2
	+ SWAG	$69.4_{0.6}$	$68.4_{1.3}$	$85.1_{0.2}$	$75.2_{0.4}$	$80.1_{0.1}$	$85.2_{0.2}$	77.2
	IVON@mean*	(+5.6) $\mathbf{72.1}_{0.5}$	(+3.2) $69.9_{0.7}$	(+2.6) $\mathbf{87.5}_{0.6}$	(+3.1) $\mathbf{76.6}_{0.5}$	(+2.0) $80.9_{0.6}$	(+0.3) $86.1_{0.2}$	(+2.8) 78.9
	IVON	(+5.7) $\mathbf{72.2}_{0.5}$	(−0.4) $66.3_{0.6}$	(+0.8) $85.7_{0.3}$	(+2.9) $\mathbf{76.4}_{0.6}$	(+1.5) $80.4_{0.4}$	(−0.1) $85.7_{0.2}$	(+1.7) 77.8

1. *Y. Shen*, N. Daheim*, B. Cong, P. Nickl, G.M. Marconi, C. Bazan, R. Yokota, I. Gurevych, D. Cremers, M.E. Khan, T. Möllenhoff.* Variational Learning is Effective for Large Deep Networks. *ICML, 2024.*
2. *B. Cong, N. Daheim, Y. Shen, D. Cremers, R. Yokota, M.E. Khan, T. Möllenhoff.* Variational Low-Rank Adaptation using IVON. *NeurIPS Workshop on Fine-Tuning in Modern ML (FITML), 2024.*

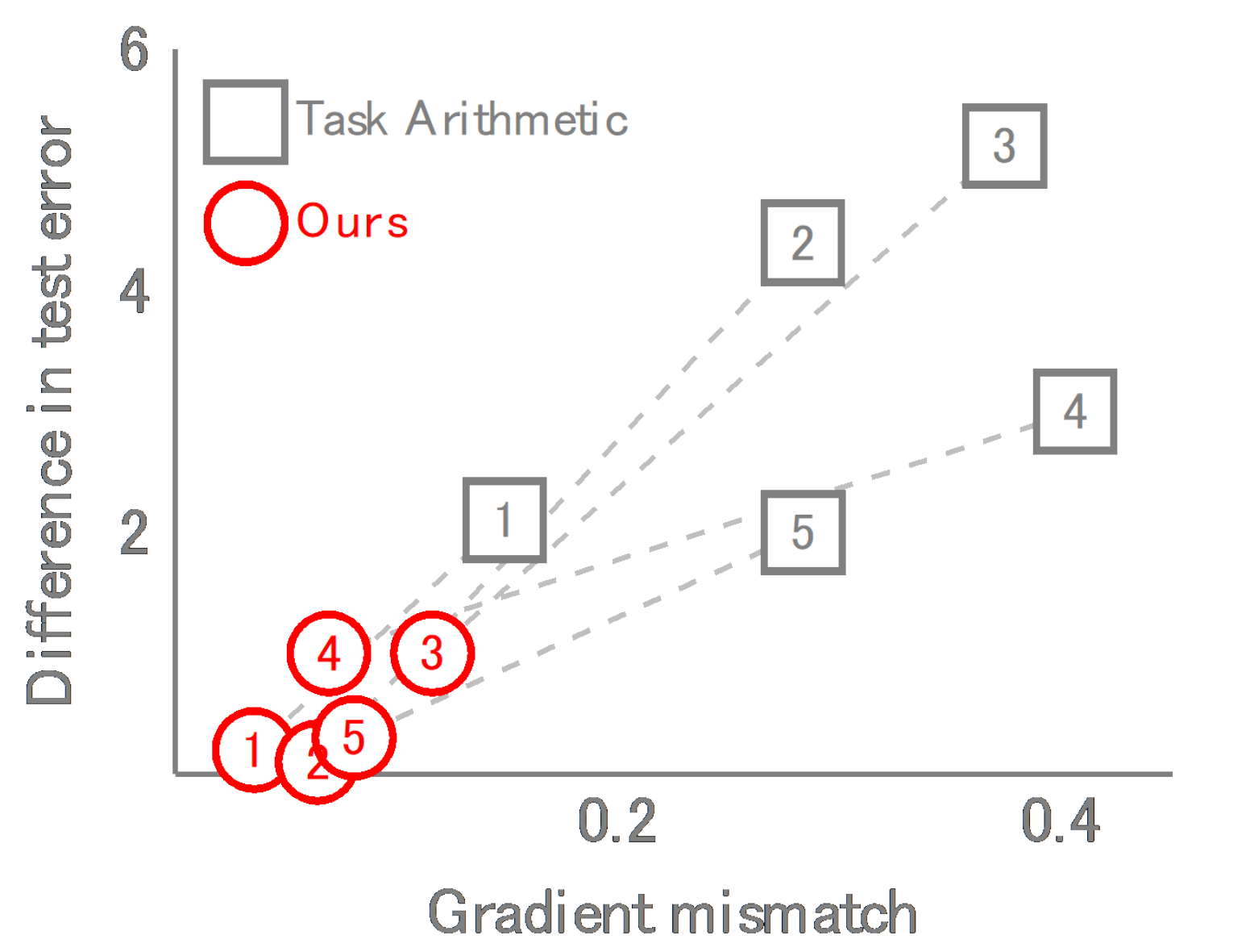
Model Merging

Problem: LLMs can be merged with finetuned models by a simple parameter addition. This works surprisingly well, but why?

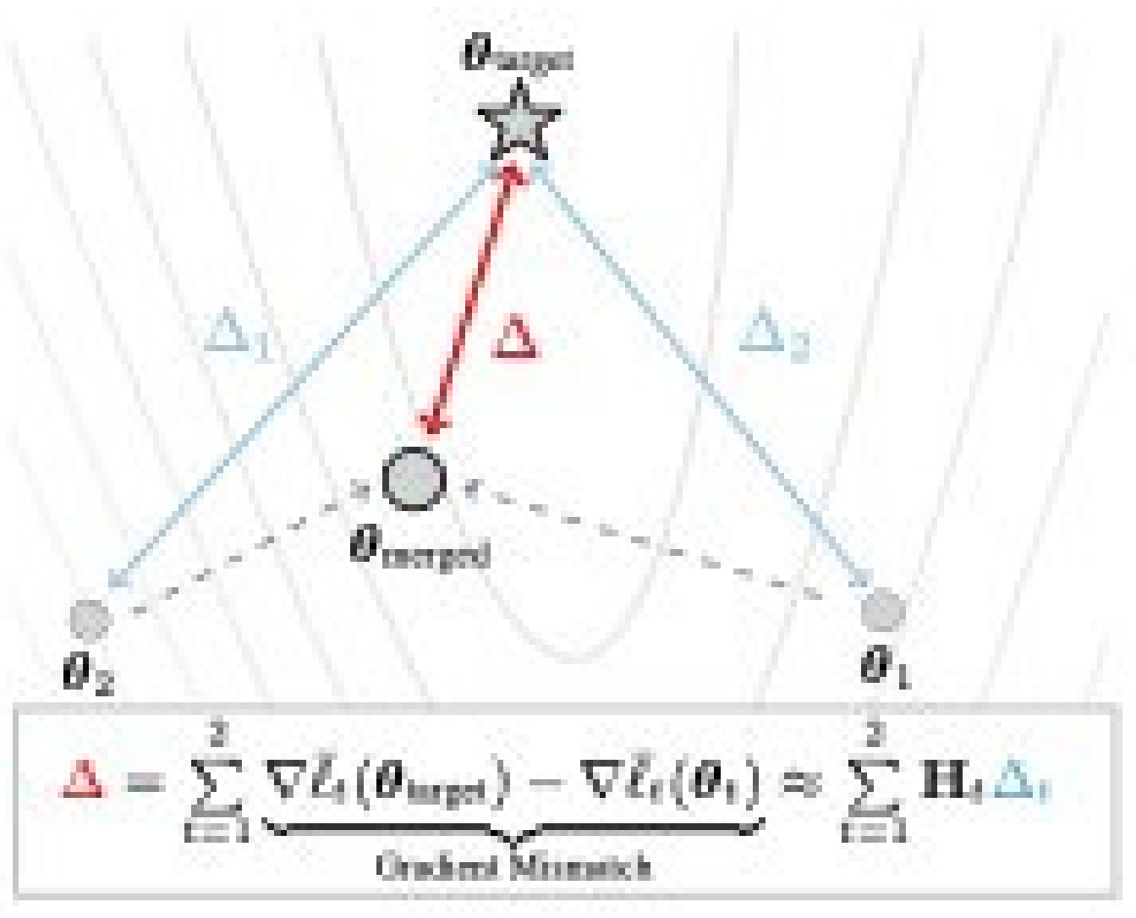
$$\begin{tikzcd} & \theta_{1+2} & \\ \theta_1 \arrow[ur] & & \theta_2 \arrow[ul] \\ \mathscr{D}_1 \arrow[u] & & \mathscr{D}_2 \arrow[u] \end{tikzcd}$$

Contribution: We connect the inaccuracy of model merging to mismatch in the gradients. When gradient mismatch is small, parameter addition works well. To reduce large gradient mismatch, we propose a Hessian-based method to reduce the error.

Gradient mismatch correlates with test error



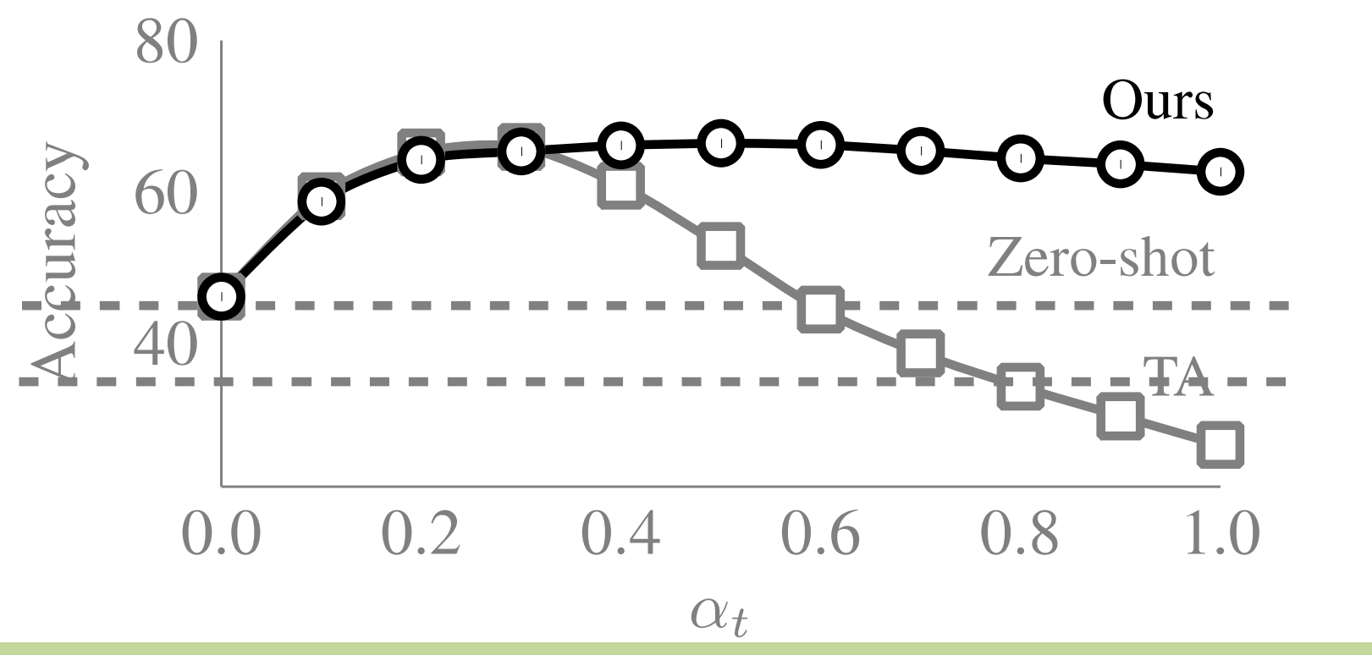
Gradient mismatch and its approximation



θ_{target}
Δ_1
Δ
Δ_2
θ_{merged}
θ_2
θ_1
$\Delta = \sum_{t=1}^{2} \underbrace{\nabla \bar{\ell}_t(\theta_{\text{target}}) - \nabla \bar{\ell}_t(\theta_t)}_{\text{Gradient Mismatch}} \approx \sum_{t=1}^{2} \mathbf{H}_t \Delta_t$

Result: Our method improves performance and is less sensitive to hyperparameters.

Model	θ	Toxicity		Fluency
		100·Avg.	Num. Toxic	PPL(↓)
GPT2_{117M}	θ_{LLM}	11.2	15.4 %	24.9
	TA	9.8	13.1 %	30.3
	ours	**9.6** (↓0.2)	**12.8 %** (↓0.3)	**26.9** (↓3.4)
$\text{GPT-J}_{1.3B}$	θ_{LLM}	11.9	16.6 %	12.6
	TA	10.7	14.5 %	**12.7**
	ours	**10.2** (↓0.5)	**14.0 %** (↓0.5)	12.8 (↓0.1)

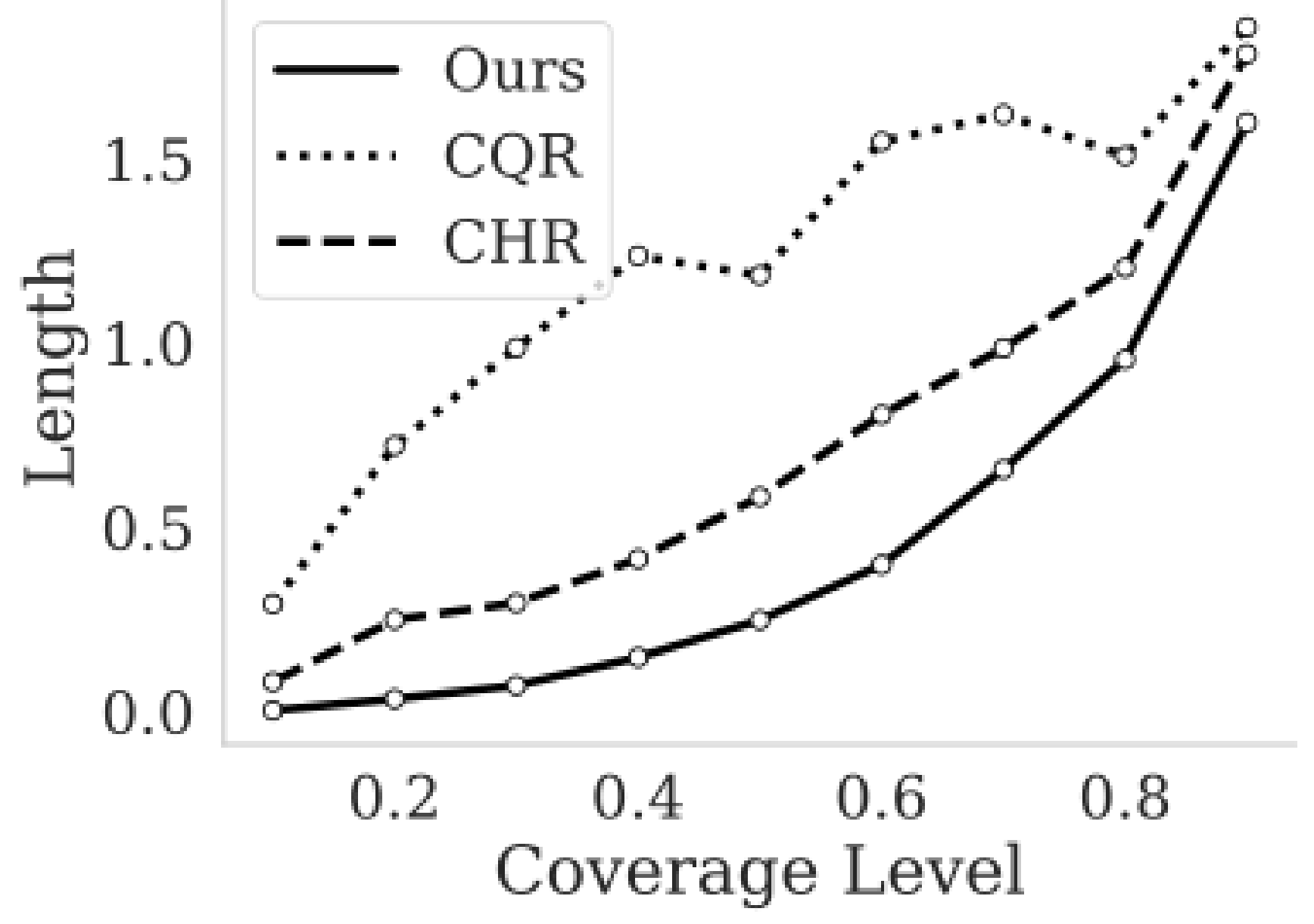


3. *N. Daheim, T. Möllenhoff, E. Ponti, I. Gurevych, M. E. Khan,* Model Merging by Uncertainty-Based Gradient Matching, *ICLR 2024*.

A New Conformal Prediction Method

Problem: Conformal Prediction (CP) aims to estimate uncertainty, but can be challenging to use for regression, especially with heteroskedastic, multimodal, or skewed distributions.

Contribution: We circumvent these challenges by converting regression to a classification problem and then use CP for classification. This gives surprisingly good results on many practical problems.



Code: pip install R2CCP (also available as part of TorchCP)

TorchCP

TorchCP: A Python toolbox for Conformal Prediction in Deep Learning.

4. *E. Guha, S. Natarajan, T. Möllenhoff, M. E. Khan, E. Ndiaye.* Conformal Prediction via Regression-as-Classification. *ICLR 2024.*

Position: Bayes is Needed in the Age of Large-Scale AI

Summary: We argue that Bayesian deep learning can improve the capabilities of deep learning, while acknowledging many of its challenges with and highlighting new research addressing the obstacles.

5. *M. E. Khan (among many authors),* Position paper: Bayesian Deep Learning in the Age of Large-Scale AI, *ICML 2024.*