Adapting Foundation Models via Training-free Dynamic Weight Interpolation

NAVER AI LAB · WISCONSIN UNIVERSITY OF WISCONSIN–MADISON

Changdae Oh[1*] Yixuan Li[1] Kyungwoo Song[2†] Sangdoo Yun[3†] Dongyoon Han[3†]

[1] University of Wisconsin—Madison [2] Yonsei University [3] NAVER AI Lab

*Work done during an internship at NAVER AI Lab, †Corresponding author

paper code

Problem define

Task of interest

1. **Robust fine-tuning** aims to achieve strong out-of-distribution (OOD) generalization while being adapted to in-distribution (ID) samples
2. **Multi-task learning** pursues establishing a unified framework that can solve multiple tasks

Model merging via weight interpolation

Allows us to construct an edited model by mixing the characteristics of individuals

$$\theta_\lambda = (1-\lambda)\theta_0 + \lambda\theta_1$$

- Existing methods usually conduct static interpolation resulting in a single fixed model
- Existing dynamic interpolation methods commonly require additional non-trivial training

Motivation

Q1) Could the finer granular merging achieve better performance?

Q2) How could we determine the proper interpolation coefficients per sample?

Method	Model Weight	IN	IN-V2	IN-R	IN-A	IN-S	ObjNet	Avg
		Acc. Under Distribution Shifts						
ZS (Radford et al., 2021)	θ_0	63.4	55.9	69.3	31.4	42.3	43.5	48.5
FT (Wortsman et al., 2022b)	θ_1	78.4	67.2	59.3	24.7	42.2	42.0	47.9
WiSE-FT (Wortsman et al., 2022b)	$(1-\lambda)\theta_0+\lambda\theta_1$	79.1	68.4	65.4	29.4	46.0	45.9	51.0
Dynamic Interpolation† (domain)	$(1-\lambda^*(\mathcal{X}))\theta_0+\lambda^*(\mathcal{X})\theta_1$	79.1	68.5	72.9	36.3	48.5	48.9	55.0
Dynamic Interpolation† (sample)	$(1-\lambda^*(x))\theta_0+\lambda^*(x)\theta_1$	**83.4**	**74.4**	**77.9**	**42.9**	**53.4**	**54.6**	**60.6**

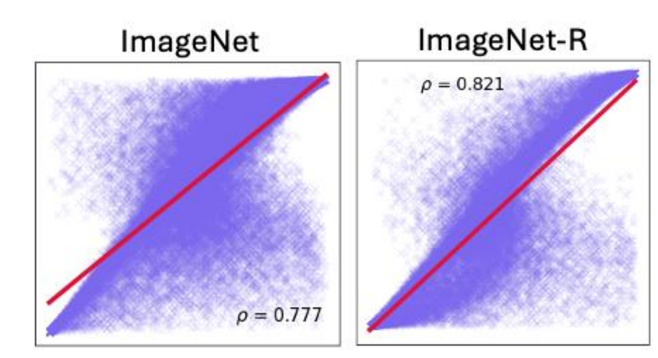
$$\lambda(x) = \frac{\exp(-l(f(x;\theta_1),y))}{\exp(-l(f(x;\theta_0),y))+\exp(-l(f(x;\theta_1),y))}$$

ratio of model expertise

X-entropy with the true labels works well, but we can not access labels during test-time

Entropy as a measure of model expertise

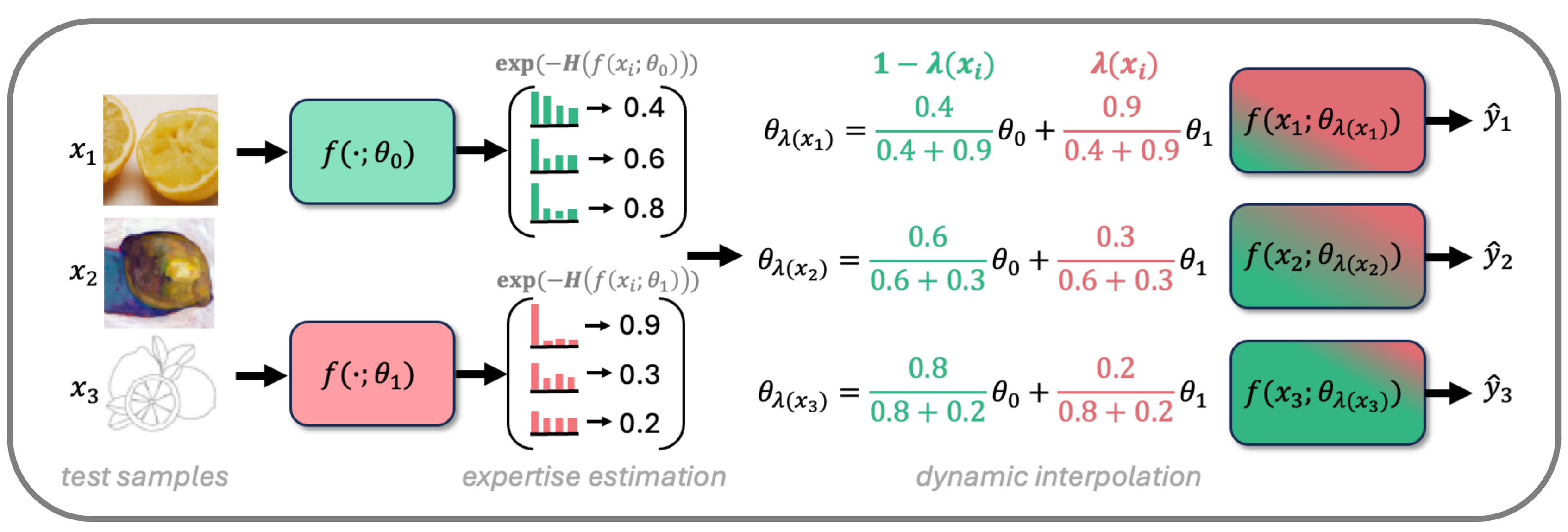
$$\lambda(x) = \frac{\exp(-H(f(x;\theta_1)))}{\exp(-H(f(x;\theta_0)))+\exp(-H(f(x;\theta_1)))}$$



Entropy ratios are strongly correlated with X-entropy ratios even under distribution shifts!

Our proposal: Training-free Dynamic Weight Interpolation (`DaWin`)

1) Dynamic weight interpolation via entropy ratio

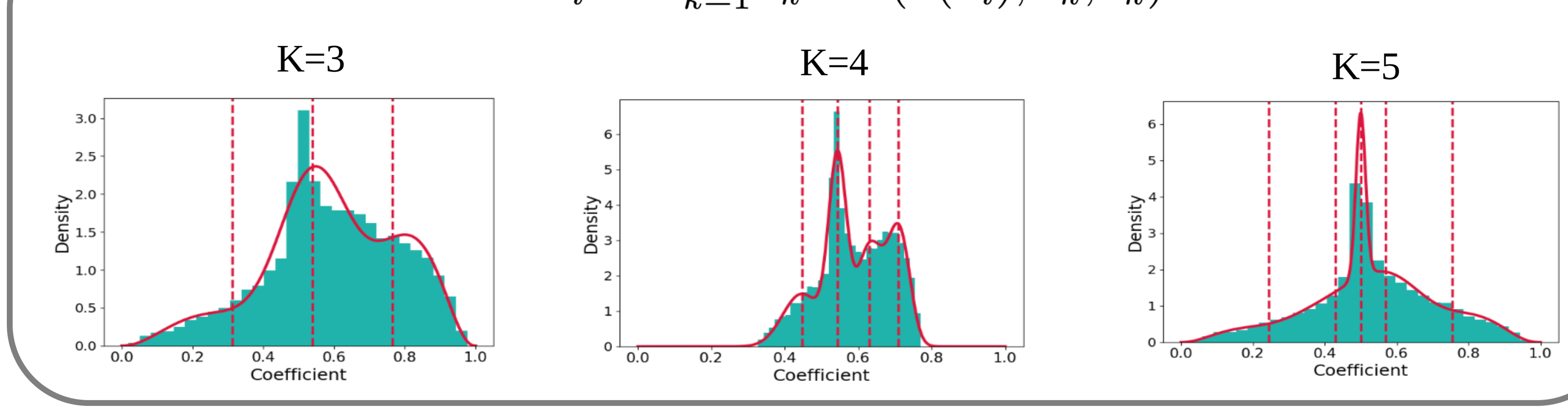


- For test-time incoming samples, we gather prediction **entropy** from individual models to **construct ratios of model expertise**
- We use this ratio as our per-sample interpolation coefficient to conduct **training-free dynamic weight interpolation**.

2) Efficient dynamic interpolation by mixture modeling

- We adopt Beta mixture model and Dirichlet mixture model based on the number of models to be merged

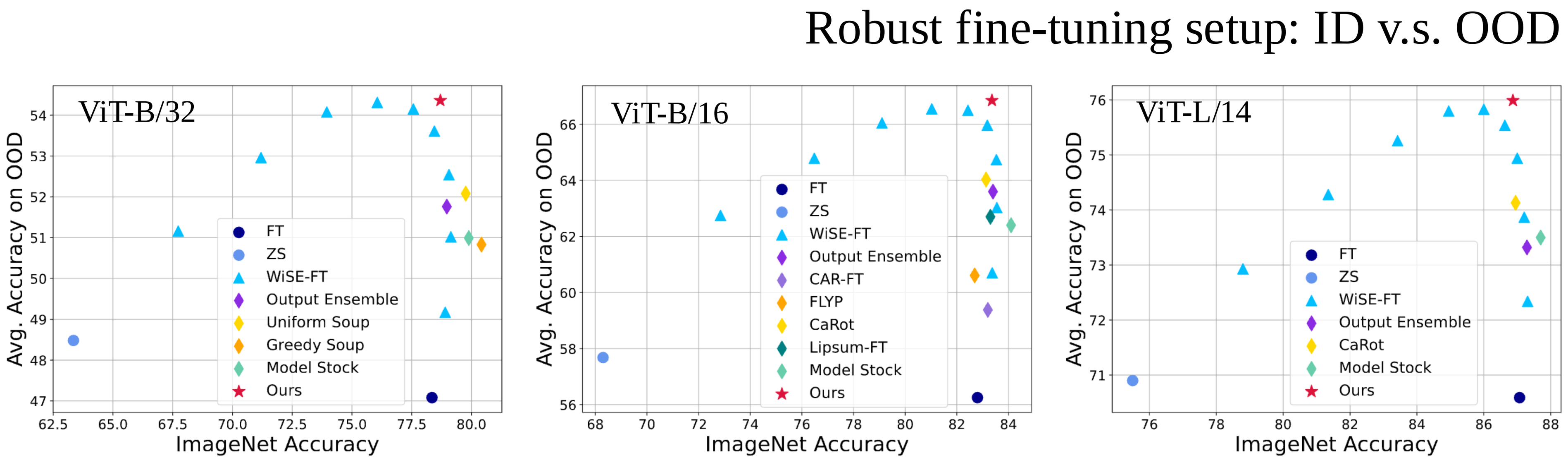
$$\Lambda_i \sim \Sigma_{k=1}^{K} \pi_k \mathrm{Beta}(\lambda(x_i); a_k, b_k)$$



- Reduce computational complexity induced by merging operations from **N** (number of entire samples in batch) to **K** (number of clusters)!

Result & Discussion

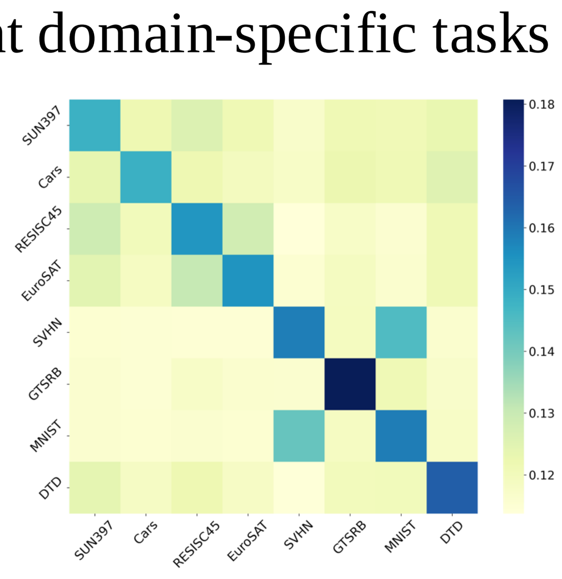
Robust fine-tuning setup: ID v.s. OOD Accuracy trade-off



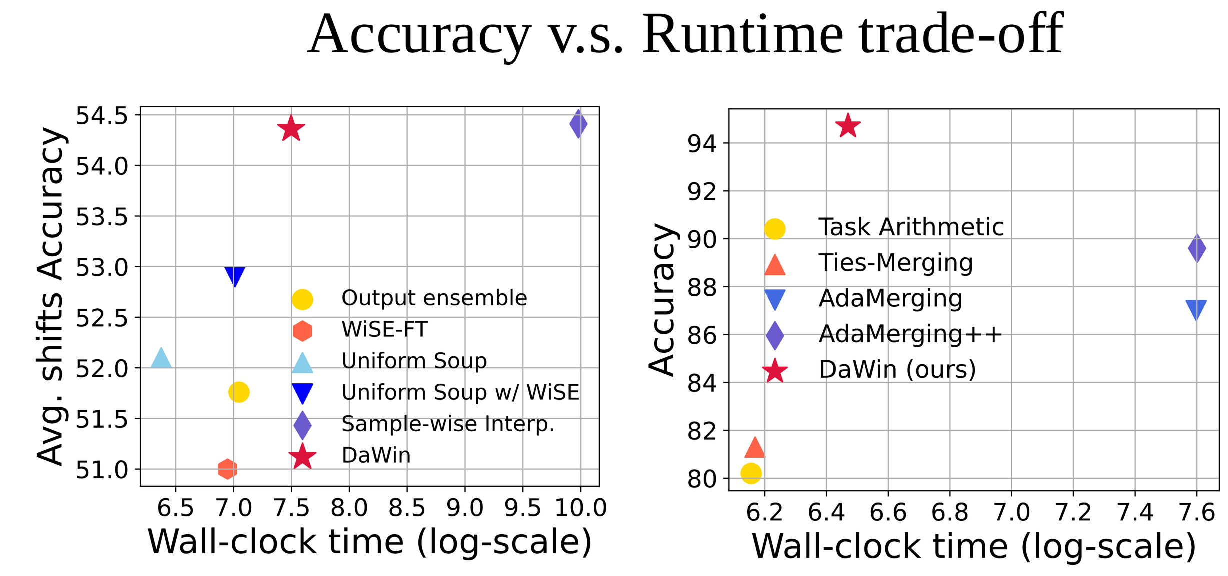
Method	Cost (T)	Cost (I)	ImageNet Acc (ID)	Avg. Acc on OOD
ZS	-	$\mathcal{O}(1)$	63.35	48.48
FT	1	$\mathcal{O}(1)$	78.35	47.08
Output ensemble	1	$\mathcal{O}(M)$	78.97	51.76
WiSE-FT (Wortsman et al., 2022b)	1	$\mathcal{O}(H)$	79.14	51.02
Uniform Soup (Wortsman et al., 2022a)	48	$\mathcal{O}(1)$	79.76	52.08
Greedy Soup (Wortsman et al., 2022a)	48	$\mathcal{O}(1)$	**80.42**	50.83
Model Stock (Jang et al., 2024)	2+α	$\mathcal{O}(1)$	79.89	50.99
`DaWin` w/o mixture modeling	1	$\mathcal{O}(N+M)$	78.71	**54.41**
`DaWin`	1	$\mathcal{O}(K+M)$	78.70	54.36

Multi-task learning setup: average accuracy across eight domain-specific tasks

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	**Avg.**
Pre-trained	63.2	59.6	60.2	45.2	31.6	32.6	48.3	44.4	48.1
Jointly fine-tuned	73.9	74.4	93.9	98.2	95.8	98.9	99.5	77.9	88.9
Individuals†	75.3	77.7	96.1	99.8	97.5	98.7	99.7	79.4	90.5
Weight Average (Ilharco et al., 2022)	65.3	63.3	71.4	72.6	64.2	52.8	87.5	50.1	65.9
Fisher Merging (Matena & Raffel, 2022)	68.6	69.2	70.7	66.4	72.9	51.1	87.9	59.9	68.3
RegMean (Jin et al., 2023)	65.3	63.5	75.6	78.6	78.1	67.4	93.7	52.0	71.8
Task Arithmetic (Ilharco et al., 2023)	55.3	54.9	66.7	75.9	80.2	69.7	97.3	50.1	68.8
Ties-Merging (Yadav et al., 2023)	65.0	64.3	74.7	75.7	81.3	69.4	96.5	54.3	72.6
AdaMerging (Yang et al., 2024b)	64.2	68.0	79.2	93.0	87.0	92.0	97.5	58.8	80.0
AdaMerging++ (Yang et al., 2024b)	65.8	68.4	82.0	93.6	89.6	89.0	98.3	60.2	80.9
Pareto Merging (Chen & Kwok, 2024)	**71.4**	**74.9**	87.0	97.1	92.0	96.8	98.2	61.1	84.8
`DaWin`	66.2	66.7	**91.3**	**99.2**	**94.7**	**98.1**	**99.5**	**74.6**	**86.3**



Accuracy v.s. Runtime trade-off

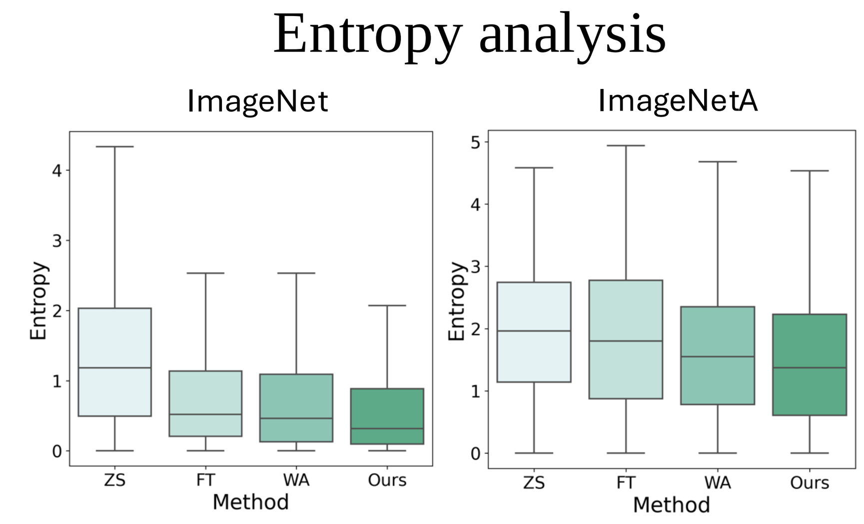


Applications: *dynamic output ensemble & classifier selection*

	Model	FT	DCS	DOE	`DaWin`
		Method			
ID	B/32	78.35	78.59	**78.71**	**78.71**
ID	B/16	82.80	82.15	83.24	**83.38**
ID	L/14	**87.07**	86.53	**87.07**	86.88
OOD	B/32	47.08	52.87	52.71	**54.41**
OOD	B/16	56.25	64.90	64.85	**66.85**
OOD	L/14	70.59	74.71	75.14	**76.01**

Ablation on expertise measure

Entropy analysis



Theoretical analysis

"*`DaWin` produces interpolation coefficients biased towards the true expert models*"

Lemma 5.1

$$\lambda_{j\in\mathcal{J}}(x) \geq \frac{1}{M} \quad \textit{where } \mathcal{J} = \{i \mid \arg\max_c [f(x;\theta_i)]_c = y\}$$
$$\textit{if } H(f(x;\theta_{j\in\mathcal{J}})) \leq H(f(x;\theta_{k\notin\mathcal{J}})) \textit{ for all } j \textit{ and } k.$$