Adapting Foundation Models via Training-free Dynamic Weight Interpolation

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

Abstract

Adapting foundation models on downstream tasks should ensure robustness against distribution shifts without the need to retrain the whole model. Although existing weight interpolation methods are simple yet effective, we argue their static nature limits downstream performance while achieving efficiency. In this work, we propose `DaWin`, a training-free **d**yn**a**mic **w**eight **in**terpolation method that leverages the prediction entropy of each unlabeled test sample to dynamically assess model expertise for per-sample interpolation coefficients. Unlike previous works that typically rely on additional training to learn such coefficients, our approach requires no training. Then, we propose a mixture modeling approach that greatly reduces inference overhead raised by dynamic interpolation. We validate `DaWin` on the large-scale visual recognition benchmarks, spanning 14 tasks across robust fine-tuning – ImageNet and its 5 distribution shift benchmarks – and multi-task learning with eight classification tasks. Results demonstrate that `DaWin` achieves significant performance gain in considered settings, with minimal computational overhead. We further discuss `DaWin`'s analytic behavior to explain its empirical success.

1 Introduction

Recent studies have shown that while fine-tuning improves a foundation model's performance on specific downstream tasks, it may damage the model's generalizability and robustness [70]. To address this issue, robust fine-tuning methods [70] have been developed to adapt models to in-distribution (ID) while maintaining out-of-distribution (OOD) generalization. Some approaches integrate regularization into the learning objective [64, 51], while others focus on preserving the knowledge of the pre-trained model by modifying tuning procedure [35, 38, 17]. Notably, *weight interpolation* approaches allow for the integration of knowledge from multiple models via simple interpolation or averaging and have proven effective on various settings [70, 69, 26, 72, 76, 29]

Weight interpolation approaches are appealing because they are easily applied to any fine-tuned model as a post-hoc plug-in method, delivering competitive performance. Most existing works [70, 69] focus on creating a single merged model using a static global interpolation coefficient λ for *all* test samples: $(1-\lambda)\theta_0 + \lambda\theta_1$, where θ_0 and θ_1 are the weights of two individual models. While these methods efficiently achieve strong performance, we argue that the optimal coefficients vary across input data samples, leaving notable potential for improving the performance of the interpolation-based approaches. Recent studies on dynamic merging [7, 44, 63] explore sample-wise interpolation with $(1-\lambda(x))\theta_0 + \lambda(x)\theta_1$, which introduce extra learnable modules to replace the global coefficient with sample-wise coefficient $\lambda(x)$ from a given sample x. Yet, these methods require additional training and careful design of the router modules to determine $\lambda(x)$, which brings non-trivial complexity.

In this work, we propose a **d**yn**a**mic **w**eight **in**terpolation framework, `DaWin`, that performs sample-wise interpolation *without requiring any additional training*. To begin with, we conduct a pilot study to investigate the upper-bound performance of dynamic interpolation methods. Specifically,

[*]Work done during an internship at NAVER AI Lab. † Equal correspondence.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

we simulate an oracle sample-wise weight interpolation by leveraging ground truth test labels for sample-wise model expertise estimation via the cross-entropy (X-entropy) loss. Given a test sample, a model yielding a smaller X-entropy is of larger importance during model merging, reflecting the expertise of the corresponding selected model. We observe that fine-grained dynamic interpolation with suitable expertise measures significantly outperforms static interpolation. Motivated by this, we design an entropy ratio-based score function that can act as a reliable alternative to the X-entropy ratio to robustly determine the sample-wise interpolation coefficients across different samples from diverse domains. Furthermore, to resolve the computation overhead induced by sample-wise interpolation operation during inference time, we further devise a mixture modeling-based [45] coefficient clustering method that dramatically reduces the computation.

Contribution: (1) We present an intuitive upper bound performance analysis of oracle dynamic interpolation methods and reveal that a variant of X-entropy ratio is a reliable metric to serve as per-sample interpolation coefficients. (2) We devise a practical method, `DaWin`, that approximates oracle dynamic interpolations by leveraging prediction entropy over unlabeled test samples. (3) Extensive validation shows that `DaWin` consistently improves classification accuracy on distribution shift and multi-task learning setups while not remarkably increasing the inference time, and we provide theoretical analyses to explain the empirical success of `DaWin`.

2 Preliminary: background and pilot study

Background. Given domain $\mathcal{X}$ of input x and a label space $\mathcal{Y} = \{1, ..., C\}$, let $f(\cdot; \theta_0)$ and $f(\cdot; \theta_1)$ denote parametric models map input x to $C-1$ dimensional simplex. They are individually trained on the same or different datasets but have identical architecture. For example, $f(\cdot; \theta_0)$ could represent a pre-trained model such as CLIP [55], while $f(\cdot; \theta_1)$ is the fine-tuned counterpart on a particular downstream task. Model merging approach [70, 69] constructs a merged model $f(\cdot; \theta_\lambda)$, which achieves a better trade-off between ID and OOD performance than the individual models by interpolating in the weight space $\theta_\lambda = (1-\lambda)\theta_0 + \lambda\theta_1$. We will use the terms interpolation and merging interchangeably. Throughout the paper, we use the term *static interpolation* to denote methods that induce a single merged model corresponding to a single interpolation coefficient used for all test samples. In contrast, *dynamic interpolation* refers to methods that generate multiple merged models, with the interpolation coefficients varying depending on the sample x or domain $\mathcal{X}$.

Goal and hypothesis. We begin by conducting a pilot study to understand the benefits of dynamic interpolation, which adapts interpolation coefficients at a finer granularity (such as sample level, $\lambda(x)$), as opposed to using global coefficient λ for all samples [70, 69]. To explore the upper limits of these methods, we experiment with ground truth labels as an *oracle* to estimate upper-bound performance and understand the maximum potential of model interpolation methods. We hypothesize that (1) Fine-grain interpolation, which adapts coefficients at a finer level (such as the sample level), can lead to substantial improvements compared to static interpolation. (2) Per-sample interpolation coefficients can effectively estimated with X-entropy-based model expertise measurement.

Setup. We assess the top-1 accuracy of several approaches on ImageNet-1K (ID) [58] and distribution-shifted (OOD) benchmarks [57, 23, 24, 67, 2] by fine-tuning the CLIP ViT-B/32 [55] on ID. Besides individual models (zero-shot; ZS and fine-tuned; FT), we include a representative static interpolation method, WiSE-FT [70], which determines the interpolation coefficient λ grid search on the ID validation set. Then, we implement two oracle interpolation methods: Dynamic Interp† per domain (d) and per sample (s). For the domain-wise interpolation, oracle coefficients $\lambda^*(\mathcal{X})$ are determined by grid search over all test samples within each domain $\mathcal{X}$, such as art or sketch. In contrast, for sample-wise interpolation coefficients $\lambda^*(x)$, we use negative X-entropy to measure models' expertise on a specific input x, which is computed as $\lambda^*(x) = \exp(-l(f(x;\theta_1), y))/(\exp(-l(f(x;\theta_0), y)) + \exp(-l(f(x;\theta_1), y)))$.

Table 1: **Pilot experiments for upper-bound analysis**. We evaluate zero-shot (ZS), fine-tuned (FT), and several interpolation methods with CLIP ViT-B/32 on ImageNet (ID) and its five distribution shifts (OOD). The domain-wise coefficients are found by grid search over each test set, and sample-wise coefficients are determined by the X-entropy ratio of individual models. † denotes *oracles* that use ground truth labels.

Method	Model Weight	ID	OOD
ZS [55]	θ_0	63.4	48.5
FT [70]	θ_1	78.4	47.9
WiSE-FT [70]	$(1-\lambda)\theta_0 + \lambda\theta_1$	79.1	51.0
Dynamic Interp.d†	$(1-\lambda^*(\mathcal{X}))\theta_0 + \lambda^*(\mathcal{X})\theta_1$	79.1	55.0
Dynamic Interp.s†	$(1-\lambda^*(x))\theta_0 + \lambda^*(x)\theta_1$	**83.4**	**60.6**

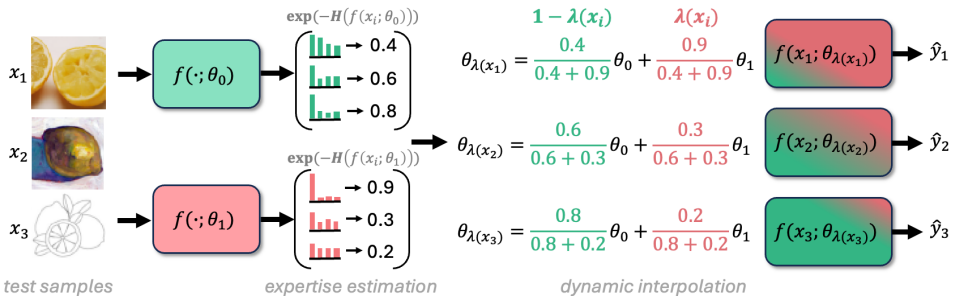

Figure 1: **Framework overview**. DaWin first estimates per sample model expertise and then produces coefficients based on the relative expertise of models to conduct dynamic interpolation.

Results and interpretations. We verify our hypothesis in Table 1. Here, the domain-wise and sample-wise dynamic interpolation methods perform far better than individual models or static interpolation. Moreover, compared with the domain-wise interpolation method, sample-wise interpolations show much higher performance in all cases. This supports our hypotheses that fine-grain interpolation leads to better downstream accuracy, and the negative X-entropy can be used as a reliable estimator of model expertise. We conclude that ***determining proper interpolation coefficients on a sample-wise basis dramatically elevates the achievable performance of model merging-based approaches***.

3 Method

Revisiting Entropy as a Measure of Model Expertise. While the pilot study in the previous section provides encouraging results, those oracle methods cannot work in practice. This is because we do not have access to the ground truth labels for incoming test samples. This leads us to the question: *how can we reliably estimate the interpolation coefficient solely based on the test input x?* There is extensive literature which adopts **entropy as a proxy of X-entropy** [18, 5, 60, 66, 54]. The rationale behind this is grounded in the observation that the entropy strongly correlates with X-entropy, even under distribution shifts [66] those models have not explicitly trained on. This work presents the first attempt to leverage the sample-wise entropy to measure each model's expertise to determine the interpolation coefficients given test samples. We claim that entropy is well-correlated with X-entropy and can thus be used to estimate model expertise. Figure 2 shows scatter plots with fitted regression lines of the entropy and X-entropy ratios with Pearson correlation coefficients on two datasets. Specifically, we plot $\frac{H(f(x;\theta_0))}{H(f(x;\theta_0))+H(f(x;\theta_1))}$ and $\frac{l(f(x;\theta_0),y)}{l(f(x;\theta_0),y)+l(f(x;\theta_1),y)}$ for entire test samples and compute Pearson correlation coefficient between them, where H denotes the entropy. Here, we observe strong correlations [59] in both datasets. This implies that entropy is a reasonable proxy for the X-entropy for computing model expertise and, ultimately, sample-wise interpolation coefficients.

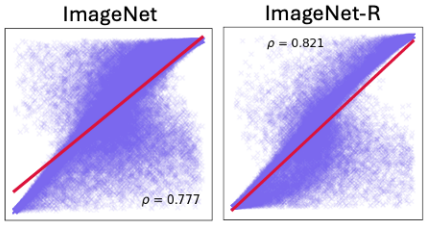

Figure 2: **Correlation between entropy ratio and X-entropy ratio**. Results imply that the entropy ratio strongly correlates with the X-entropy ratio both on ID and OOD datasets.

Dynamic Interpolation via Entropy Ratio. After confirming a strong correlation between the entropy and X-entropy, we propose a new dynamic weight interpolation, DaWin, by defining sample-wise interpolation coefficients $\lambda(x)$ as below:

$$\lambda(x) = \frac{\exp(-H(f(x;\theta_1)))}{\exp(-H(f(x;\theta_0))) + \exp(-H(f(x;\theta_1)))}. \tag{1}$$

where H represents the entropy over the output probability distribution of $f(\cdot;\theta)$ given input x. The value of $\lambda(x)$ approaches 1 when model $f(\cdot;\theta_1)$ exhibits lower entropy (greater expertise), whereas it approaches 0 if the entropy of model $f(\cdot;\theta_1)$ becomes higher. Figure 1 illustrates the overall framework. Intuitively, this approach is analogous to classifier selection methods [14, 33], which aims to dynamically select suitable classifier(s) given a test sample. However, DaWin differs in terms of its motivation, expertise metric, and goal as we pursue finding the interpolation coefficients rather than selecting the best model (See Table 6). We discuss the connection between DaWin and the selection-based method in App. A.5. Note that our sample-wise expertise estimations **do not require any hyperparameters** in contrast to prior works [70, 27]. However, performing interpolation on every sample can substantially increase the inference-time computation, with the

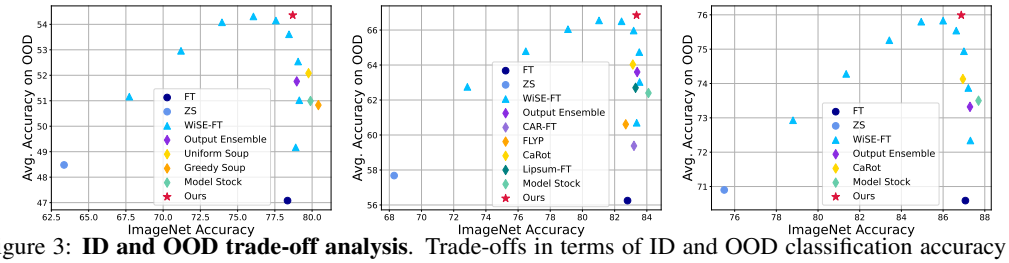

Figure 3: **ID and OOD trade-off analysis**. Trade-offs in terms of ID and OOD classification accuracy on ImageNet distribution shift benchmarks with CLIP ViT-{B/32, B/16, L/14} from left to right. The blue curve is the hyperparameter sweep $\{0.1, 0.2, ..., 0.9\}$ of WiSE-FT. `DaWin` achieves performance beyond the Pareto-optimal points given two models, ZS and FT. (See Figure 7 for more).

cost scaling linearly with the number of model parameters [44]. To ensure practicality while pursuing outstanding downstream performance, we devise an efficient dynamic interpolation by mixture modeling and conduct mixture component-wise interpolation. Meanwhile, if we assume we can access all test samples simultaneously, we can refine the coefficient computation by introducing per-domain expertise offset terms and use this technique by default (see App. A.1) for omitted details.

4 Experiments

Setup. We validate `DaWin` on **robust fine-tuning** [70] and **multi-task learning** [26] setups with focusing on the top-1 classification accuracy (Acc). For robust fine-tuning, we use ImageNet-1K and its five OOD variants, ImageNet-V2, ImageNet-R, ImageNet-A, ImageNet-Sketch, and ObjectNet, to evaluate robustness under distribution shifts and adopt CLIP ViT-{B/32, B/16, L/14} [55] as zero-shot (ZS) models to ensure a fair comparison with [70, 69], and fine-tuned (FT) checkpoints for each backbone from Jang et al. [29]. For multi-task learning, we follow the conventional protocol [26] using CLIP ViT-B/32 on 8 datasets: SUN397, Cars, RESISC45, EuroSAT, SVHN, GTSRB, MNIST, and DTD. See App. A.2 for omitted details on implementation and baseline methods.

Better performance trade-off between ID and OOD. In Figure 3, we present the accuracy for ID (x-axis) and OOD average (y-axis) across three backbone models, along with baseline performance. Achieving superior ID/OOD trade-offs beyond or even comparable to the blue curve from the WiSE-FT hyperparameter sweep is challenging. However, `DaWin` provides a remarkably better performance trade-off compared to baselines. It is worth noting that `DaWin` does not require any hyperparameter tuning. Instead, it dynamically generates interpolation coefficients solely based on the trained models' output. This distinguishes `DaWin` from other hyperparameter-intensive fine-tuning such as CaRot [51] or existing interpolation methods [70]. Tab. 2 provides detailed accuracies and costs. Compared with Uniform and Greedy Soups and Model Stock, which require more than two models to ensure diversity among fine-tuned models [69] or periodic interpolations during training [29], `DaWin` operates with only a single fine-tuned model while achieving significantly better accuracy trade-off with slightly increased inference cost. These trends hold across other backbones (see Tab. 4 and 5), verifying `DaWin`'s generality in terms of modeling scale. Although `DaWin` compromises the inference-time efficiency for accuracy, given that existing works typically require hyperparameter tuning in practice [26], relative computation overhead is minor (See Tab. 5).

Table 2: Accuracy on ImageNet (ID) and its OOD for CLIP ViT-B/32. Cost (T, I) denote the number of training required to build the final model and the number of inference evaluations, respectively. Given M models, N denotes the number of test samples, and H and K denote the size of the hyperparameter grid and the number of mixture components, respectively.

Method	Cost (T)	Cost (I)	ID	OOD
ZS	-	$\mathcal{O}(1)$	63.3	48.4
FT	1	$\mathcal{O}(1)$	78.3	47.0
Output ensemble	1	$\mathcal{O}(M)$	78.9	51.7
WiSE-FT [70]	1	$\mathcal{O}(H)$	79.1	51.0
Uniform Soup [69]	48	$\mathcal{O}(1)$	79.7	52.0
Greedy Soup [69]	48	$\mathcal{O}(1)$	**80.4**	50.8
Model Stock [29]	2+α	$\mathcal{O}(1)$	79.8	50.9
`DaWin` **(sample-wise)**	1	$\mathcal{O}(N+M)$	78.7	**54.4**
`DaWin`	1	$\mathcal{O}(K+M)$	78.7	54.3

Multi-task adaptation capability. We now evaluate existing merging approaches and our `DaWin` on multi-task learning benchmarks with CLIP ViT-B/32. Following the standard evaluation protocol [74], we have M tasks and models individually fine-tuned on each task (where $M = 8$). We do not know where each test sample arises from during evaluation and cannot choose true experts per domain. *While both AdaMerging and `DaWin` use unlabeled testset, `DaWin` produces dynamic merged models given tasks or samples, whereas AdaMerging induces a single merged model for all tasks by default.*

Table 3: **Multi-task learning performance.** We use CLIP ViT-B/32 for evaluation across eight benchmark tasks. † denotes optimal-like performance using the ground truth domain indicator to select models for each domain. We report the layer-wise method for AdaMerging [74] here, which surpasses those of the task-wise approach.

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
Individuals†	75.3	77.7	96.1	99.8	97.5	98.7	99.7	79.4	90.5
Fisher Merging [48]	68.6	69.2	70.7	66.4	72.9	51.1	87.9	59.9	68.3
RegMean [31]	65.3	63.5	75.6	78.6	78.1	67.4	93.7	52.0	71.8
Task Arithmetic [27]	55.3	54.9	66.7	75.9	80.2	69.7	97.3	50.1	68.8
Ties-Merging [72]	65.0	64.3	74.7	75.7	81.3	69.4	96.5	54.3	72.6
AdaMerging [74]	64.2	68.0	79.2	93.0	87.0	92.0	97.5	58.8	80.0
AdaMerging++ [74]	65.8	68.4	82.0	93.6	89.6	89.0	98.3	60.2	80.9
Pareto Merging [6]	**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**

Here, we modify the interpolation formula in Sec. 3 into task arithmetic formulation, *i.e.*, given weights of pre-trained model θ_0 and fine-tuned models $\{\theta_j\}_{j=1}^M$, a dynamic interpolation is defined as $\theta_{\lambda(x)} = \theta_0 + \lambda_0 \sum_{j=1}^M \lambda_j(x)\tau_j$ where $\tau_j = \theta_j - \theta_0$, $\lambda_j(x)$, and λ_0 denote the task vector [27], weight for j-th task vector, and scaling term (set to 0.3 as Ilharco et al. [27]), respectively. In Table 3, DaWin greatly outperforms advanced averaging [31] and adaptive merging [6] methods that require tough training, and approaching a ground truth expert selection method, e.g., Individuals†. This verifies the versatility of DaWin, which is beneficial for adapting the model on multiple tasks as well as a single-task setup. Figure 4 shows the average of estimated sample-wise coefficients (y-axis) per dataset (x-axis). DaWin assigns the highest weights to true experts (diagonal) per task and leverages relevant experts by reflecting task-wise similarity.

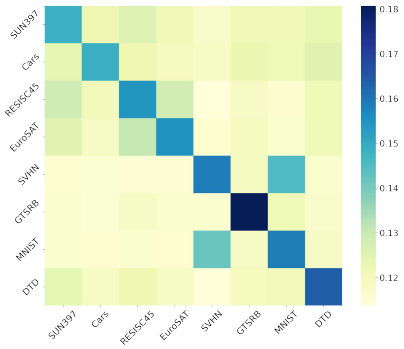

Figure 4: **Merging coefficient visualization.** We visualize DaWin's coefficients across 8 datasets (columns) with 8 fine-tuned models (rows).

Accuracy and runtime trade-off. While we focus on improving accuracy through model merging approaches, it is crucial to ensure the extra computation demands of DaWin during inference are manageable. Fig. 5 presents trade-offs between the accuracy and wall-clock time for various merging methods. For methods requiring hyperparameter tuning, e.g., WiSE-FT and Task Arithmetic, the wall-block time (logarithm of sec.) reflects a cumulative time for evaluations across all hyperparameters. For methods like DaWin or AdaMerging, we include the time required for additional workloads. In both settings, DaWin shows favorable trade-offs that outperform the most efficient one in terms of Acc. while its runtime is far less than computation-heavier methods such as AdaMerging or per sample interpolation without mixture model (Sample-wise Interp). Thus, DaWin provides a high-performing solution that can be flexibly adapted considering a given cost budget.

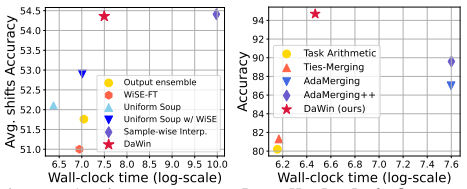

Figure 5: **Accuracy and wall-clock inference time with CLIP ViT-B/32**. We compared search-based, optimization-based, and our merging methods regarding accuracy and wall-clock inference time on robust fine-tuning (left) and multi-task learning evaluation on SVHN dataset (right).

5 Conclusion

This work has presented the first training-free dynamic weight interpolation method, DaWin, by estimating the sample-wise model expertise with entropy to produce reliable interpolation coefficients. We further proposed mixture modeling approaches to enhance computational efficiency. We then extensively evaluated DaWin on 14 benchmark classification tasks with three different backbone models in terms of ID/OOD performance trade-off in the robust fine-tuning scenario and the overall accuracy in the multi-task learning scenario. The consistent performance gains compared with state-of-the-art baselines empirically validated DaWin's effectiveness, and this empirical success was further explained by discussing an analytic behavior of DaWin (Please refer to App. A.5).

Acknowledgments

We thank the researchers and interns at NAVER AI Lab – Byeongho Heo, Taekyung Kim, Sanghyuk Chun, Sehyun Kwon, Yong-Hyun Park, and Jaeyoo Park – for their invaluable feedback. Most of the experiments are based on the NAVER Smart Machine Learning (NSML) platform [62]. We also appreciate on the constructive comments from Max Khanov, Hyeong Kyu Choi, and Seongheon Park at University of Wisconsin–Madison. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2022-00143911, AI Excellence Global Innovative Leader Education Program) and the National Research Foundation of Korea (NRF) grant funded by the Korea government(MSIT)(RS-2024-00457216, 2022R1A4A3033874)

References

[1] Kazi Md Rokibul Alam, Nazmul Siddique, and Hojjat Adeli. A dynamic ensemble learning algorithm for neural networks. *Neural Computing and Applications*, 32(12):8675–8690, 2020.

[2] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[3] Benjamin Biggs, Arjun Seshadri, Yang Zou, Achin Jain, Aditya Golatkar, Yusheng Xie, Alessandro Achille, Ashwin Swaminathan, and Stefano Soatto. Diffusion soup: Model merging for text-to-image diffusion models. *arXiv preprint arXiv:2406.08431*, 2024.

[4] Alceu S. Britto, Robert Sabourin, and Luiz E.S. Oliveira. Dynamic selection of classifiers—a comprehensive review. *Pattern Recognition*, 47(11):3665–3680, 2014. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2014.05.003. URL https://www.sciencedirect.com/science/article/pii/S0031320314001885.

[5] Olivier Chapelle and Alexander Zien. Semi-supervised classification by low density separation. In *International workshop on artificial intelligence and statistics*, pages 57–64. PMLR, 2005.

[6] Weiyu Chen and James Kwok. You only merge once: Learning the pareto set of preference-aware model merging. *arXiv preprint arXiv:2408.12105*, 2024.

[7] Feng Cheng, Ziyang Wang, Yi-Lin Sung, Yan-Bo Lin, Mohit Bansal, and Gedas Bertasius. Dam: Dynamic adapter merging for continual video qa learning. *arXiv preprint arXiv:2403.08755*, 2024.

[8] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

[9] Caroline Choi, Yoonho Lee, Annie Chen, Allan Zhou, Aditi Raghunathan, and Chelsea Finn. Autoft: Robust fine-tuning by optimizing hyperparameters on ood data. *arXiv preprint arXiv:2401.10220*, 2024.

[10] Alexandra Chronopoulou, Matthew E Peters, Alexander Fraser, and Jesse Dodge. Adaptersoup: Weight averaging to improve generalization of pretrained language models. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2054–2063, 2023.

[11] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, pages 3606–3613, 2014.

[12] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pages 7480–7512. PMLR, 2023.

[13] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pages 3259–3269. PMLR, 2020.

[14] Giorgio Giacinto and Fabio Roli. Dynamic classifier selection based on multiple classifier behaviour. *Pattern Recognition*, 34(9):1879–1881, 2001. ISSN 0031-3203. doi: https://doi.org/10.1016/S0031-3203(00)00150-3. URL https://www.sciencedirect.com/science/article/pii/S0031320300001503.

[15] Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vlad Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. Arcee's mergekit: A toolkit for merging large language models. *arXiv preprint arXiv:2403.13257*, 2024.

[16] IJ Good. Explicativity: a mathematical theory of explanation with statistical applications. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 354(1678):303–330, 1977.

[17] Sachin Goyal, Ananya Kumar, Sankalp Garg, Zico Kolter, and Aditi Raghunathan. Finetune like you pretrain: Improved finetuning of zero-shot vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19338–19347, 2023.

[18] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 2004.

[19] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[20] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

[21] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

[22] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

[23] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[24] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[25] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

[26] Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. *Advances in Neural Information Processing Systems*, 35:29262–29277, 2022.

[27] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023.

[28] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

[29] Dong-Hwan Jang, Sangdoo Yun, and Dongyoon Han. Model stock: All we need is just a few fine-tuned models. *European Conference on Computer Vision (ECCV)*, 2024.

[30] Ruochen Jin, Bojian Hou, Jiancong Xiao, Weijie Su, and Li Shen. Fine-tuning linear layers only is a simple yet effective way for task arithmetic. *arXiv preprint arXiv:2407.07089*, 2024.

[31] Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*, 2023.

[32] Haotian Ju, Dongyue Li, and Hongyang R Zhang. Robust fine-tuning of deep neural networks with hessian-based generalization guarantees. In *International Conference on Machine Learning*, pages 10431–10461. PMLR, 2022.

[33] Albert H.R. Ko, Robert Sabourin, and Alceu Souza Britto, Jr. From dynamic classifier selection to dynamic ensemble selection. *Pattern Recognition*, 41(5):1718–1731, 2008. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2007.10.015. URL https://www.sciencedirect.com/science/article/pii/S0031320307004499.

[34] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV workshops*, pages 554–561, 2013.

[35] Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022.

[36] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[37] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*. Atlanta, 2013.

[38] Yoonho Lee, Annie S Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical fine-tuning improves adaptation to distribution shifts. In *The Eleventh International Conference on Learning Representations*, 2023.

[39] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, 2021.

[40] Ziyue Li, Kan Ren, XINYANG JIANG, Yifei Shen, Haipeng Zhang, and Dongsheng Li. SIMPLE: Specialized model-sample matching for domain generalization. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=BqrPeZ_e5P.

[41] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2023.

[42] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

[43] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*, 2024.

[44] Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Dangyang Chen, and Yu Cheng. Twin-merging: Dynamic integration of modular expertise in model merging. *arXiv preprint arXiv:2406.15479*, 2024.

[45] Zhanyu Ma and Arne Leijon. Bayesian estimation of beta mixture models with variational inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2160–2173, 2011.

[46] Xiaofeng Mao, Yufeng Chen, Xiaojun Jia, Rong Zhang, Hui Xue, and Zhao Li. Context-aware robust fine-tuning. *International Journal of Computer Vision*, 132(5):1685–1700, 2024.

[47] Daniel Marczak, Bartłomiej Twardowski, Tomasz Trzciński, and Sebastian Cygert. Magmax: Leveraging model merging for seamless continual learning. *European Conference on Computer Vision (ECCV)*, 2024.

[48] Michael S Matena and Colin A Raffel. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716, 2022.

[49] Nithin Gopalakrishnan Nair, Jeya Maria Jose Valanarasu, and Vishal M Patel. Maxfusion: Plug&play multi-modal generation in text-to-image diffusion models. *arXiv preprint arXiv:2404.09977*, 2024.

[50] Giung Nam, Byeongho Heo, and Juho Lee. Lipsum-ft: Robust fine-tuning of zero-shot models using random text guidance. In *The Twelfth International Conference on Learning Representations*, 2024.

[51] Changdae Oh, Hyesu Lim, Mijoo Kim, Dongyoon Han, Sangdoo Yun, Jaegul Choo, Alexander Hauptmann, Zhi-Qi Cheng, and Kyungwoo Song. Towards calibrated robust fine-tuning of vision-language models. *Advances in Neural Information Processing Systems*, 2024.

[52] Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. *Advances in Neural Information Processing Systems*, 36, 2024.

[53] Karl Pearson. Method of moments and method of maximum likelihood. *Biometrika*, 28(1/2):34–59, 1936. ISSN 00063444, 14643510. URL http://www.jstor.org/stable/2334123.

[54] Viraj Prabhu, Shivam Khare, Deeksha Kartik, and Judy Hoffman. Sentry: Selective entropy optimization via committee consistency for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8558–8567, 2021.

[55] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[56] Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Advances in Neural Information Processing Systems*, 36, 2023.

[57] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning (ICML)*, 2019.

[58] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[59] Patrick Schober, Christa Boer, and Lothar A Schwarte. Correlation coefficients: Appropriate use and interpretation. *Anesth. Analg.*, 126(5):1763–1768, May 2018.

[60] Rui Shu, Hung Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. In *International Conference on Learning Representations*, 2018.

[61] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *IJCNN*, pages 1453–1460. IEEE, 2011.

[62] Nako Sung, Minkyu Kim, Hyunwoo Jo, Youngil Yang, Jingwoong Kim, Leonard Lausen, Youngkwan Kim, Gayoung Lee, Donghyun Kwak, Jung-Woo Ha, et al. Nsml: A machine learning platform that enables you to focus on your models. *arXiv preprint arXiv:1712.05902*, 2017.

[63] Anke Tang, Li Shen, Yong Luo, Nan Yin, Lefei Zhang, and Dacheng Tao. Merging multi-task models via weight-ensembling mixture of experts. In *Forty-first International Conference on Machine Learning*, 2024.

[64] Junjiao Tian, Zecheng He, Xiaoliang Dai, Chih-Yao Ma, Yen-Cheng Liu, and Zsolt Kira. Trainable projected gradient method for robust fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7836–7845, 2023.

[65] Junjiao Tian, Yen-Cheng Liu, James S Smith, and Zsolt Kira. Fast trainable projection for robust fine-tuning. *Advances in Neural Information Processing Systems*, 36, 2023.

[66] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021.

[67] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[68] Ke Wang, Nikolaos Dimitriadis, Guillermo Ortiz-Jimenez, François Fleuret, and Pascal Frossard. Localizing task information for improved model merging and compression. *ICML*, 2024.

[69] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR, 2022.

[70] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7959–7971, 2022.

[71] Jianxiong Xiao, Krista A Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *IJCV*, 119:3–22, 2016.

[72] Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36, 2023.

[73] Enneng Yang, Li Shen, Zhenyi Wang, Guibing Guo, Xiaojun Chen, Xingwei Wang, and Dacheng Tao. Representation surgery for multi-task model merging. *ICML*, 2024.

[74] Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. In *International Conference on Learning Representations*, 2024.

[75] LIN Yong, Lu Tan, Yifan HAO, Ho Nam Wong, Hanze Dong, WEIZHONG ZHANG, Yujiu Yang, and Tong Zhang. Spurious feature diversification improves out-of-distribution generalization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=d6H4RBi7RH.

[76] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*, 2024.

[77] Netzer Yuval. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[78] Zhanpeng Zhou, Yongyi Yang, Xiaojiang Yang, Junchi Yan, and Wei Hu. Going beyond linear mode connectivity: The layerwise linear feature connectivity. *Advances in Neural Information Processing Systems*, 36:60853–60877, 2023.

A Appendix

We provide the following items in this Appendix:

- (A.1) Algorithm and additional details on DaWin
- (A.2) Additional details on the experiment setup
- (A.3) Additional empirical results
- (A.4) Related works
- (A.5) Theoretical analysis
- (A.6) Limitation and future work

A.1 Algorithm and additional details on DaWin

Algorithm 1: Procedure for training-free dynamic weight interpolation (DaWin)

Input: Test samples $X = \{x_i\}_{i=1}^N \in \mathbb{R}^{N \times D}$, models $f(\cdot;\theta_0)$ and $f(\cdot;\theta_1)$ of the same architecture, entropy function $H(\cdot)$, and the number of mixture component K

for $i = 1, \dots, N$ **do**
  $(H_{i,0}, H_{i,1}) \leftarrow (H(f(x_i;\theta_0)), H(f(x_i;\theta_1)))$ `// Per-sample model expertise`
  if *offset adjustment* **then**
    $\lambda(x_i) \leftarrow \frac{\exp(-H_{i,1}) + O(X)/T(X)}{\exp(-H_{i,0}) + \exp(-H_{i,1}) + O(X)}$
  else
    $\lambda(x_i) \leftarrow \frac{\exp(-H_{i,1})}{\exp(-H_{i,0}) + \exp(-H_{i,1})}$ `// Per-sample interp. coefficient`
  end
end
bmm $\leftarrow$ `BetaMixture(`$\{\lambda(x_i)\}_{i=1}^N$`;`K`).fit()`
$\{m_i\}_{i=1}^N \leftarrow$ `bmm.predict(`X`)` `// Membership inference on test samples`
for $k = 0, \dots, K-1$ **do**
  $X_k \leftarrow X[\mathbb{I}(m_i == k), :]$
  $\lambda(X_k) \leftarrow$ `bmm[`k`].mean()`
  $\theta_{\lambda(X_k)} \leftarrow (1 - \lambda(X_k))\theta_0 + \lambda(X_k)\theta_1$ `// Cluster-wise dynamic interp.`
end

X-entropy ratio and likelihood ratio. Let $f(x;\theta) = p_\theta(y|x)$ be a classifier parameterized with θ that models the ground truth conditional probability distribution $p(y|x)$. Given that we observe one-hot encoded target labels so that $l(f(x;\theta), y) = -\log(p_\theta(y|x))$, then we can rewrite the oracle sample-wise coefficient in Sec 2 as follows, $\lambda^*(x) = p_{\theta_1}(y|x)/(p_{\theta_0}(y|x) + p_{\theta_1}(y|x))$, which can be interpreted as a posterior of Bernoulli distribution in the *noise-contrastive estimation* [21], i.e., $\lambda^*(x) = p_{\theta_1}(y|x)/(p_{\theta_0}(y|x) + p_{\theta_1}(y|x))$, which models the probability whether a data (x, y) comes from the distribution $p_{\theta_1}(y|x)$.

Sample-wise entropy valley hypothesis. One of the most popular hypotheses that explain the success of weight interpolations is *linear mode connectivity* (LMC; Frankle et al. [13]), which ensures that interpolated models are also laid on good solution space as well as low-loss converged individual models. Although entropy is a concave function that denies the LMC in nature, we believe that a similar statement can be claimed, which we refer to *sample-wise entropy valley* as follows:

i) There exists λ such that $H(f(x; \lambda\theta_0 + (1-\lambda)\theta_1) \leq \min\{H(f(x;\theta_0)), H(f(x;\theta_1))\}$.

ii) By observing the existence of sample-wise linear feature connectivity [78] conditioned on LMC, the λ obeyed to (i) is varying depending on the input data sample x.

As empirical evidence, Figure 12 shows that the interpolated model actually induces smaller entropy than individuals (i) on average, and DaWin achieves lower entropy than static interpolation (ii). Given the strong correlation between entropy and X-entropy, this hypothesis advocates the benefit of DaWin than static interpolation methods.

Incorporating domain-wise expertise. While the eq equation 1 yields sample-wise interpolation coefficients solely based on the per-sample expertise estimation, if we can access the whole test samples simultaneously, we can refine the coefficient computation by introducing domain-wise offset terms. These terms are automatically estimated by the per-domain entropy of each model. Given the significant performance improvements with domain-wise dynamic interpolation (*c.f.* Table 1), we expect domain-wise expertise per model can be a complimentary benefit to compute the interpolation coefficient. Therefore, we modify the sample-wise coefficient term in Eq. 1 as below:

$$\lambda(x) = \frac{\exp(-H(f(x;\theta_1))) + O(X)/T(X)}{\exp(-H(f(x;\theta_0))) + \exp(-H(f(x;\theta_1))) + O(X)}, \tag{2}$$

$$O(X) = \frac{1}{2}\left(\frac{\text{std}(H(f(X;\theta_0)))}{\text{mean}(H(f(X;\theta_0)))} + \frac{\text{std}(H(f(X;\theta_1)))}{\text{mean}(H(f(X;\theta_1)))}\right) \quad T(X) = \frac{H(f(X;\theta_0)) + H(f(X;\theta_1))}{H(f(X;\theta_0))},$$

where $O(X)$ and $T(X)$ denote the per-domain offset (formulated as an average coefficient of variation) and relative domain expertise terms, respectively. By incorporating domain-wise expertise, DaWIN offers more stable dynamic interpolation coefficients, correcting some of the inaccuracies in sample-wise estimation (see Table 7). It is worth noting that these sample-wise and domain-wise expertise estimations **do not require any hyperparameters** for computing interpolation coefficients. This eliminates the need for intensive hyperparameter tuning that is often required in prior works [70, 27] to achieve the best performance.

Efficient Dynamic Interpolation by Mixture Modeling Although our primary objective is to achieve better downstream task performance by way of dynamic interpolation, performing interpolation on every sample can substantially increase the inference-time computation, with the cost scaling linearly with the number of model parameters [44]. To ensure practicality while pursuing state-of-the-art downstream performance, we devise an efficient dynamic interpolation by mixture modeling. Let $\{x_i\}_{i=1}^N$ denote the N test samples and $\{\lambda(x_i)\}_{i=1}^N$ the corresponding interpolation coefficient computed by `DaWin` framework. Note that $\lambda_i \in [0,1]$ for all $i = 1, \ldots, N$, and they can be regarded as observed random variables from some Beta distributions. Then, we accurately model the probability density function of interpolation coefficients via **Beta mixture model**[2] as follows:

$$\Lambda_1, ..., \Lambda_N \sim \Sigma_{k=1}^K \pi_k \text{Beta}(x_i; a_k, b_k),$$

where Λ_i indicates random variables of $\lambda(x_i)$ and $\{\pi_k\}_{k=1}^K$ are the prior probabilities for each Beta distribution $\text{Beta}(\cdot; a_1, b_1), ..., \text{Beta}(\cdot; a_K, b_K)$. We initialize the parameters $(a_k, b_k)_{k=1}^K$ randomly for each Beta distribution and estimate the initial membership of each sample via K-means clustering. Then, the expectation-maximization (EM) algorithm is adopted to iteratively refine the parameters of the Beta mixture and the membership inference quality. This process only takes around 30 seconds for 50K samples. Finally, given a test sample x_i, we infer the membership of that sample via maximum likelihood $k^* = \arg\max_k \text{Beta}(x_i; a_k, b_k)$, and use the mean of the corresponding Beta distribution $\frac{a_{k^*}}{a_{k^*}+b_{k^*}}$. This approach significantly reduces the computational burden of sample-wise dynamic interpolation from N (number of test samples) to K (number of pre-defined clusters), where $K \ll N$. We outline the detailed procedures of `DaWin` in Algorithm.

Details on Beta Mixture Model (BMM) with Expectation Maximization algorithm. To enhance the inference-time efficiency of dynamic interpolation, we propose a mixture modeling approach over the estimated per-sample interpolation coefficients to reduce the number of interpolation operations from N (entire test sample) to K (pre-defined number of mixture components). Here, we elaborate on the detailed procedure of Beta Mixture Modeling[3].

We have coefficient estimates over N test samples via Eq. 1 or Eq. 2 as $\{\lambda(x_i)\}_{i=1}^N$. Our goal is to model those coefficients with a mixture of K Beta distributions as below:

$$\text{Beta}(\lambda(x_i); a_k, b_k) = \frac{\Gamma(a_k + b_k)}{\Gamma(a_k)\Gamma(b_k)} \lambda(x_i)^{a_k - 1}(1 - \lambda(x_i))^{b_k - 1} \tag{3}$$

$$p(\lambda(x_i)) = \sum_{k=1}^{K} \pi_k \text{Beta}(\lambda(x_i); a_k, b_k) \tag{4}$$

[2]For cases of interpolating more than two models, this can be easily extended to Dirichlet mixture model

[3]The same procedure is adopted for the Dirichlet Mixture Model in multi-task learning scenario by modifying the probability density function from Beta distribution to Dirichlet distribution.

where $a_k > 0$ and $b_k > 0$ are the shape parameters of component k, $\Gamma(\cdot)$ and π_k denote the Gamma function and mixing prior probabilities for each Beta component satisfying $\sum_{k=1}^{K} \pi_k = 1$ and $\pi_k \geq 0$. We first initialize the responsibilities $\{\gamma_{ik}\}$ by applying K-Means clustering to $\{\lambda(x_i)\}$ to assign the initial membership per each observation to one of K components. We also initialize the parameter estimates of BMM as below:

$$\text{Mixing Priors}(\pi_k) : \pi_k^{(0)} = \frac{N_k^{(0)}}{N}, \quad \text{where} \;\; N_k^{(0)} = \sum_{i=1}^{N} \gamma_{ik}^{(0)}. \tag{5}$$

$$\text{Shape Parameters}(a_k, b_k) : a_k^{(0)} = C_k^{(0)} \times \bar{\lambda}_k^0 + \epsilon, \tag{6}$$

$$b_k^{(0)} = C_k^{(0)} \times (1 - \bar{\lambda}_k^{(0)}) + \epsilon \tag{7}$$

$$\text{where} \;\; \bar{\lambda}_k^{(0)} = \frac{1}{N_k^0} \sum_{i=1}^{N} \gamma_{ik}^{(0)} \lambda(x_i), \tag{8}$$

$$s_k^{2,(0)} = \frac{1}{N_k^0} \sum_{i=1}^{N} \gamma_{ik}^{(0)} (\lambda(x_i) - \bar{\lambda}_k^{(0)})^2, \tag{9}$$

$$C_k^{(0)} = \frac{\bar{\lambda}_k^0 (1 - \bar{\lambda}_k^0)}{s_k^{2,(0)}} - 1, \tag{10}$$

where ϵ is a small positive constant to ensure numerical stability. Here, the initial shape parameters are estimated by *method-of-moments* [53]. Then, we conduct the expectation step (E-step) and the maximization step (M-step) alternatively until convergence to refine the parameter estimate as follows:

- **E-step**:
  - Compute log responsibilities $\ln\gamma_{ik}^{(t)}$.
  - Update responsibilities $\gamma_{ik}^{(t)} = \exp(\ln\gamma_{ik}^{(t)})$.
- **M-step**:
  - Update mixing priors $\pi_k^{(t+1)}$.
  - Update shape parameters $a_k^{(t+1)}, b_k^{(t+1)}$ using *method-of-moments* estimation.
- **Convergence Check**:
  - Compute log-likelihood $\mathcal{L}^{(t+1)}$, where $\mathcal{L}^{(t)} = \sum_{i=1}^{N} \ln p(\lambda(x_i))$
  - If $|\mathcal{L}^{(t+1)} - \mathcal{L}^{(t)}| <$ tolerence, stop the iterations.

Then, we get the estimated parameter $\Theta = \{\pi_k, a_k, b_k\}_{k=1}^{K}$ of BMM to infer per-sample weight interpolation coefficients.

A.2 Additional details on the experiment setup

In this section, we provide extended details for task definition, baseline methods, and implementation details. Some contents might be duplicated from the main paper.

A.2.1 Tasks and Datasets

We validate `DaWin` on two scenarios: (1) **robust fine-tuning** [70] and (2) **multi-task learning** [26] with focusing on the top-1 classification accuracy (Acc). Following robust fine-tuning literature [70, 35], we use ImageNet-1K [58] and its five variants, ImageNet-V2 [57], a post-decade reproduced version of the original ImageNet test set by following the dataset generating process of ImageNet, ImageNet-R [23], a rendition-specific collection of 200 ImageNet classes, ImageNet-A [24], an actual examples from ImageNet test set misclassified by a ResNet-50 model over 200 ImageNet classes, ImageNet-Sketch [67], a sketch-specific collection of 1000 ImageNet classes, and ObjectNet [2] for evaluating robustness under distribution shifts. For multi-task learning, we follow the standard evaluation protocol [26, 74] using eight benchmark datasets from the optical character images, traffic signs, scenery or satellite imagery, and fine-grain categorization over cars and texture: SUN397 [71], Cars [34], RESISC45 [8], EuroSAT [22], SVHN [77], GTSRB [61], MNIST [36], and DTD [11].

A.2.2 Models and Baselines

For robust fine-tuning, we adopt CLIP ViT-{B/32, B/16, L/14} [55] as zero-shot (ZS) models to ensure a fair comparison with [70, 69, 26, 74], and fine-tuned (FT) checkpoints for each CLIP ViT backbone from Jang et al. [29]. For the weight interpolation baseline methods, we include WiSE-FT [70], which conducts a weight interpolation between a pre-trained model and a fine-tuned model given a single pre-defined interpolation coefficient, Model Soup [69], which conducts averaging of all models' weights (Uniform Soup) or greedily selected models' weights (Greedy Soup) trained with different training hyperparameter configurations, and Model Stock [29], iteratively interpolate a pre-trained model with few fine-tuning model based on the cosine distance between pre-trained model weight and the fine-tuning model weights, along with the traditional output ensemble method. In addition, we consider several state-of-the-art robust fine-tuning methods such as LP-FT [35], a two-stage method to avoid pre-trained feature distortion, CAR-FT [46], a regularized fine-tuning method leveraging context-aware prompt, FLYP [17], a contrastive learning based fine-tuning method, Lipsum-FT [50], a regularized fine-tuning method motivated by energy score gap between zero-shot and fine-tuned models, and CaRot [51], a theory-inspired singular value regularization method.

For multi-task learning, we use CLIP ViT-B/32 as our backbone and consider the model merging baselines as follows: a simple weight averaging [26], Fisher Merging [48], a Fisher information metric-based weighted averaging method, RegMean [31], an averaging method that minimizes L_2 distance between the averaged weight and individual weights, Task Arithmetic [27], a method perform arithmetic across task vectors rather the original weight vector itself which are produced by the subtractions between a pre-trained model weight and individual fine-tuned model weights, Ties-Merging [72], a post-hoc weight refinement method mitigating conflicts between task vectors, AdaMerging [74] and Pareto Merging [6] methods those driving additional optimization procedure to a global interpolation coefficient and the conditional coefficient generation models that trained to minimize entropy over entire test sample.

A.2.3 Implementation Details

For fine-tuning CLIPs on ImageNet, Wortsman et al. [69] and its successor [29] conducted multiple training with different training configurations such as data augmentation, learning rate, weight decay, and random initialization seeds given fixed epochs (16) and batch size (512). Here, we adopted the best model weight provided by the authors of Jang et al. [29] per each backbone. For fine-tuning weights of CLIP on multi-task learning setup, we adopt the official checkpoints from Ilharco et al. [27] on the eight datasets.

On `DaWin`'s evaluation, we first get the entropy of batch test samples from the interpolation candidate models[4] (wherein temperature scaling [20] is applied in the robust fine-tuning setup with ID validation set), then we compute interpolation coefficients by building a softmax-like model expertise ratio term with exponentiated negative entropy of each model. We further perform the mixture modeling over the batch test samples and finally conduct dynamic model merging with interpolation coefficients corresponding to estimated membership from the fitted mixture model to obtain the prediction per sample. About the Beta (on robust fine-tuning setup) and Dirichlet (on multi-tasks learning setup) mixture modeling on interpolation coefficients for `DaWin`, we set K to 3, 5, 2 for ViT-{B/32, B/16, L/14} in the robust fine-tuning and $K = 1$ in the multi-task setups. Unless otherwise mentioned, we adopt the offset adjustment term (Eq. equation 2) by default, assuming that the entire test samples per task are available and fit the Beta (and Dirichlet) mixture model on the entire coefficients per task, likewise assumption of Yang et al. [74, 73], Chen and Kwok [6]. Our code will be publicized upon paper acceptance.

Table 4: Accuracy on ImageNet (ID) and distribution shifts (OOD) for CLIP ViT-B/16.

Method	ID	OOD Avg.
ZS	68.3	57.7
FT	82.8	56.3
LP-FT [35]	82.5	61.3
CAR-FT [46]	83.2	59.4
FLYP [17]	82.7	60.6
Lipsum-FT [50]	83.3	62.7
CaRot [51]	83.1	64.0
Output ensemble	83.4	63.6
WiSE-FT [70]	83.5	64.2
Model Stock [29]	**84.1**	62.4
`DaWin`	83.4	**66.9**

Table 5: Accuracy on ImageNet (ID) and distribution shifts (OOD) for CLIP ViT-L/14.

Method	ID	OOD Avg.
ZS	75.5	70.9
FT	87.0	70.6
FLYP	86.2	71.4
CaRot	87.0	74.1
Output ensemble	87.3	73.3
WiSE-FT	87.3	73.2
Model Stock	**87.7**	73.5
`DaWin`	86.9	**76.0**

A.3 Additional empirical results

Ablation study. We explore alternative metrics beyond entropy for estimating sample-wise model expertise, as shown in Figure 6. Specifically, we consider four different pseudo label (PL) approaches [37] to replace the entropy, $\{\hat{y}, \tilde{y}\} \times \{\text{soft}, \text{hard}\}$, where $\hat{y} = \frac{1}{2}f(x;\theta_0) + \frac{1}{2}f(x;\theta_1)$ and $\tilde{y} = f(x; \frac{1}{2}\theta_0 + \frac{1}{2}\theta_1)$. The settings $\{\text{soft}, \text{hard}\}$ indicate whether the `argmax` operation is applied to PL or not. We observe that some PLs slightly outperform entropy in terms of ID accuracy but bring no gains in OOD accuracy. To maintain simplicity in line with Occam's razor [16], we adopt entropy as the expertise metric on unlabeled test samples.

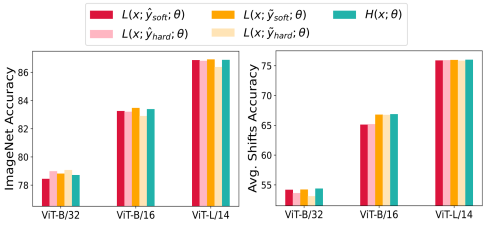

Figure 6: **Effectiveness of our expertise metric**. We evaluate five different expertise metrics on ID (left) and OOD (right). H behaves like L, with performance nearly matching or even surpassing the best-performing L.

Applications: dynamic selection and dynamic output ensemble. Given that `DaWin` is motivated by the concept of "*entropy as a measure of model expertise*," we can extend this beyond weight interpolation to two other approaches, dynamic classifier selection (DCS; Giacinto and Roli [14], Britto et al. [4]) and dynamic output ensemble (DOE; Alam et al. [1], Li et al. [40]).

To be specific, DCS aims to select the most suitable classifier for each test sample based on the *competence* (referred to as *expertise* in our terminology) among multiple classifiers. We provide the results of DCS by adopting entropy as a competence measurement for classifier selection, *i.e.*, selecting a lower entropy model per sample. Besides, we also present a dynamic output ensemble (DOE) by applying Eq. 2 directly on the output probability space rather than weight space. In Table 6, we see that entropy-based DCS and DOE bring large performance gains compared with a single fine-tuned model (FT), while the `DaWin` achieves the largest gain. This implies that users can flexibly build their dynamic system using test sample entropy according to their budget and performance criteria.

Table 6: Results of per-sample dynamic classifier selection (DCS), dynamic output ensemble (DOE), and dynamic weight interpolation (`DaWin`) on ImageNet (ID) and under distribution shifts (OOD) for different CLIP ViT backbone models.

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

We ablate the offset adjustment and expertise metric to investigate the effectiveness of the design choices of each component. Firstly, as we can see in Table 7, offset adjustment consistently boosts the ID and OOD accuracy across all cases, which supports the use of domain-wise relative expertise (average entropy over all test samples) to enhance sample-wise expertise estimation. To secure inference time efficiency, we adopt the Beta mixture modeling

[4]Candidate models are constituted with the pre-trained and fine-tuned models for robust fine-tuning setting, the task-specific eight fine-tuned models for multi-task learning setting.

Table 7: **Ablation study on offset adjustment**. We validate the effect of using the offset adjustment term on ImageNet ID and OOD accuracy.

Model	Offset Adjustment	ID	Avg. Acc on OOD
ViT-B/32	-	78.3	54.1
	✓	**78.7**	**54.4**
ViT-B/16	-	83.1	66.6
	✓	**83.4**	**66.9**
ViT-L/14	-	86.7	75.9
	✓	**86.9**	**76.0**

Table 8: **Sensitivity analysis on sample size.** We report `DaWin`'s ImageNet and its OOD variants' accuracy of CLIP ViT-B/32 under varying sample wise for fitting Beta Mixture Model. `DaWin` shows robustness against varying sample size.

Sample size	ImageNet Accuracy	Avg. Acc on OOD
32	78.55	54.30
64	78.69	54.27
128	78.71	54.27
256	78.70	54.28
512	78.70	54.25
1024	78.71	54.24
2048	78.67	54.26
N	78.70	54.36

approach on the batch-wise `DaWin`'s coefficients. The fitness of Beta mixture model may be improved as batchsize is increased, whereas smaller batchsize enables applying finer-granular interpolations. Therefore, we evaluate the performance of `DaWin` under varying batchsize. Table 8 presents the ID and OOD performance of `DaWin` on the ImageNet distribution shift benchmark for CLIP ViT-B/32 backbone model. `DaWin` shows strong robustness against varying batchsize.

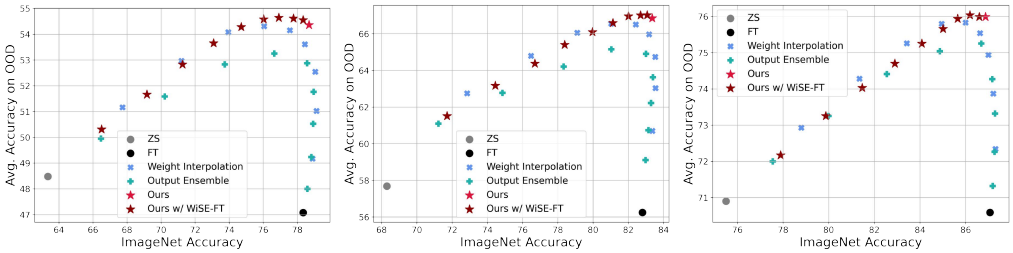

Figure 7: **ID and OOD trade-off analysis**. We visualize performance trade-offs regarding ID and OOD classification accuracy on the ImageNet distribution shift benchmarks with CLIP ViT-{B/32, B/16, L/14} from left to right. Interpolation coefficients are swept over $\{0.1, 0.2, ..., 0.9\}$.

In Figure 7, we present the results of `DaWin` with WiSE-FT as a plug-in augmentation on the robust fine-tuning setup. We multiply our estimated coefficients from Beta mixture by a scalar coefficient α, e.g., $\lambda_{wise}(x) = \lambda(x) \times \alpha$. Although it brings slight benefits in the case of CLIP ViT-B/32, the WiSE-FT interpolation trace becomes almost a line on the ViT-B/16 and ViT-L/14 cases. This implies that `DaWin` already achieves performance beyond the Pareto-optimal trade-offs and cannot be further improved by WiSE-FT, given models $f(\cdot;\theta_0)$ and $f(\cdot;\theta_1)$. Meanwhile, Figure 8 reveals that `DaWin` produces larger $\lambda(x)$ for samples that are hard to recognize the target object due to overwhelming background semantics, which the pre-trained model may wrongly pay attention to.

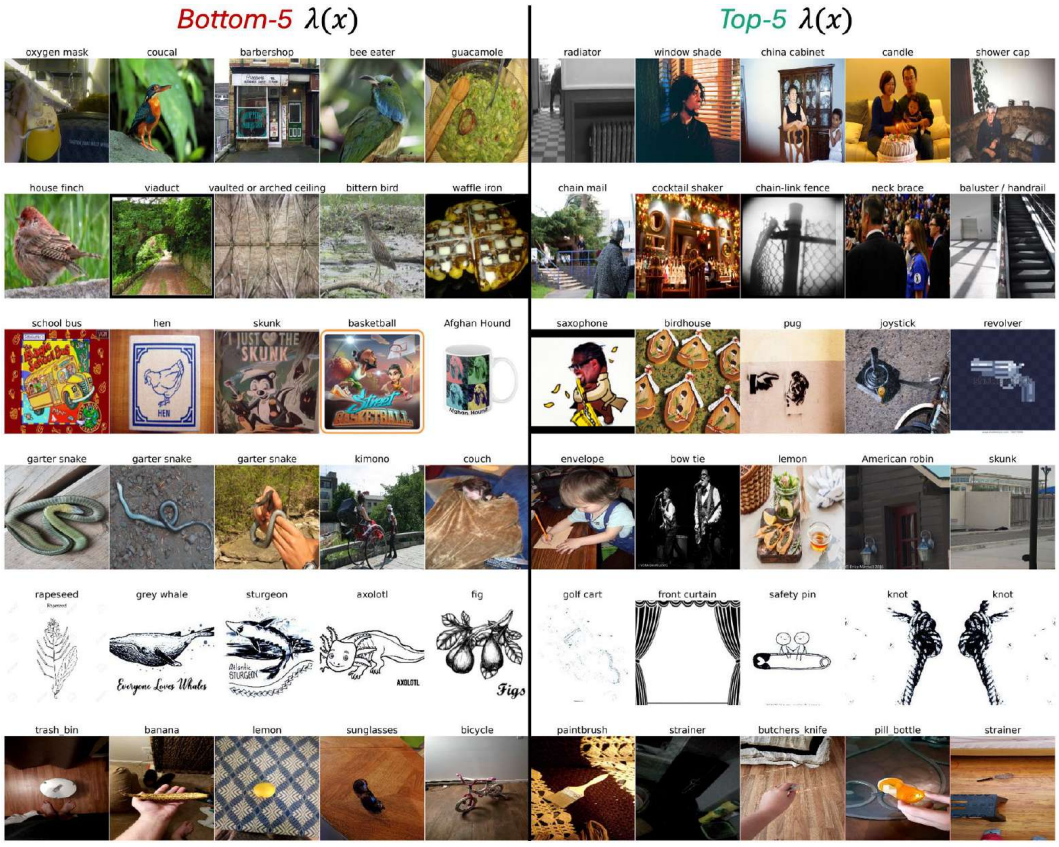

Figure 8: **DaWIN's bottom-5 and top-5 estimated coefficient analysis**. We visualize the images with labels corresponding to bottom-5 and top-5 sample-wise interpolation coefficients estimated by DaWIN of pre-trained and ImageNet fine-tuned CLIP ViT-B/32. Each row denotes the actual test samples from ImageNet, ImageNet-V2, ImageNet-R, ImageNet-A, ImageNet-Sketch, and ObjectNet. We see that the images corresponding to coefficients lean towards the fine-tuned model, which typically contains multiple semantics, and the object corresponding to the ground truth label is overwhelmed by other semantics.

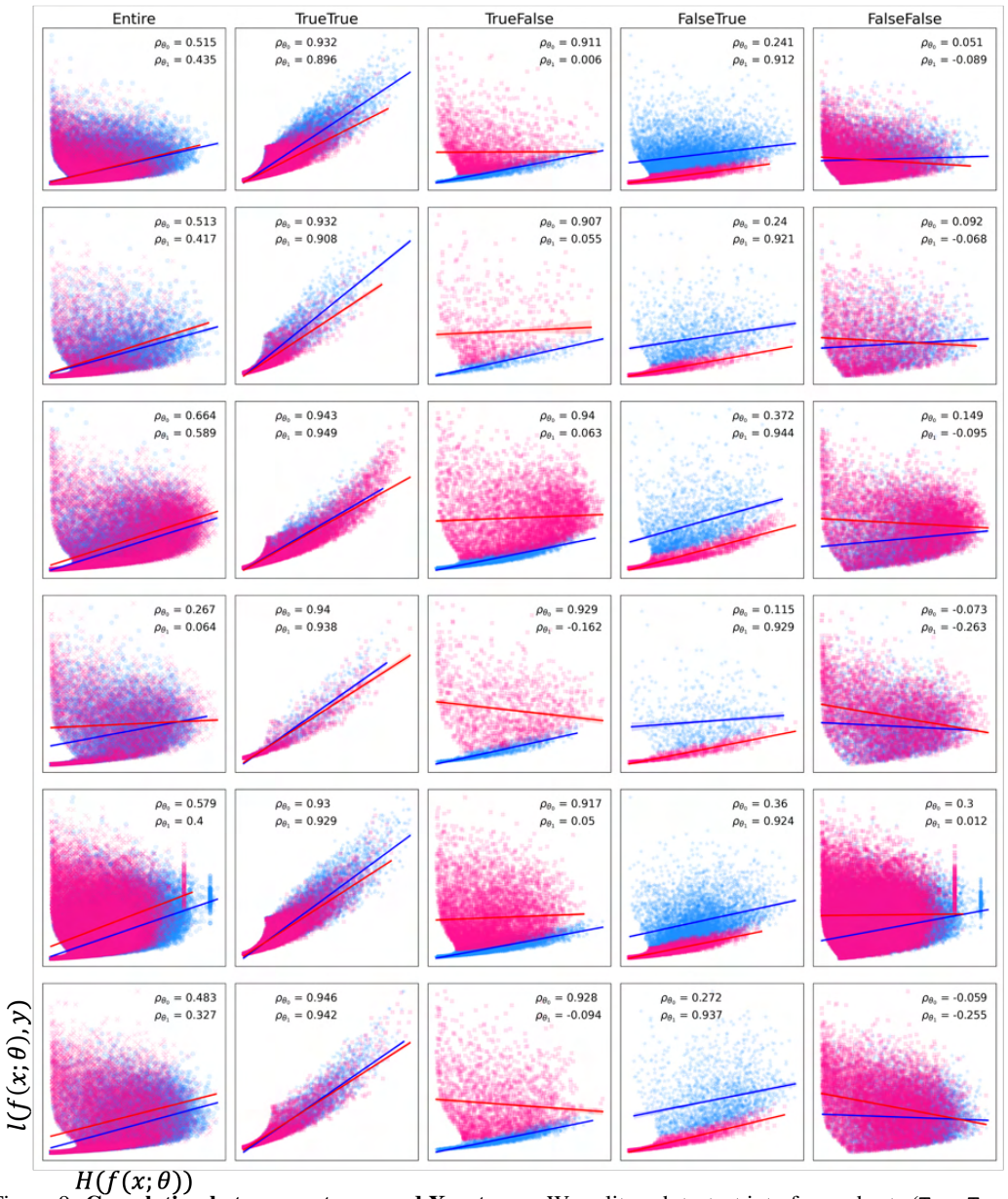

Figure 9: **Correlation between entropy and X-entropy**. We split each testset into four subsets (`TrueTrue`, `TrueFalse`, `FalseTrue`, `FalseFalse`) based on the correctness of the two models' predictions. On each data split and the entire dataset, we visualize scatter plots of entropy (x-axis) and cross-entropy (y-axis) computed by two models (zero-shot and fine-tuned) with corresponding Pearson correlation coefficients per model. Each row from top to bottom shows results from ImageNet, ImageNetV2, ImageNetR, ImageNetA, ImageNetSketch, and ObjectNet.

`DaWin` adopts entropy as a proxy of X-entropy to estimate the model expertise without access to the true target label. Figure 9 presents the correlation between entropy and X-entropy of the pre-trained CLIP ViT-B/32 $f(\cdot;\theta_0)$ and ImageNet fine-tuned counterpart $f(\cdot;\theta_1)$. Results indicate that the model producing correct predictions holds a strong correlation between entropy and X-entropy while the model failing to correctly predict test samples shows bad or no correlation. However, if at least one model success in making a correct prediction in a given sample x, and the entropy of the correct predictor may be smaller than that of another model, thereby `DaWin` would be likely to produce $\lambda(x)$ biased towards the correct predictor's weight (See Lemma A.5).

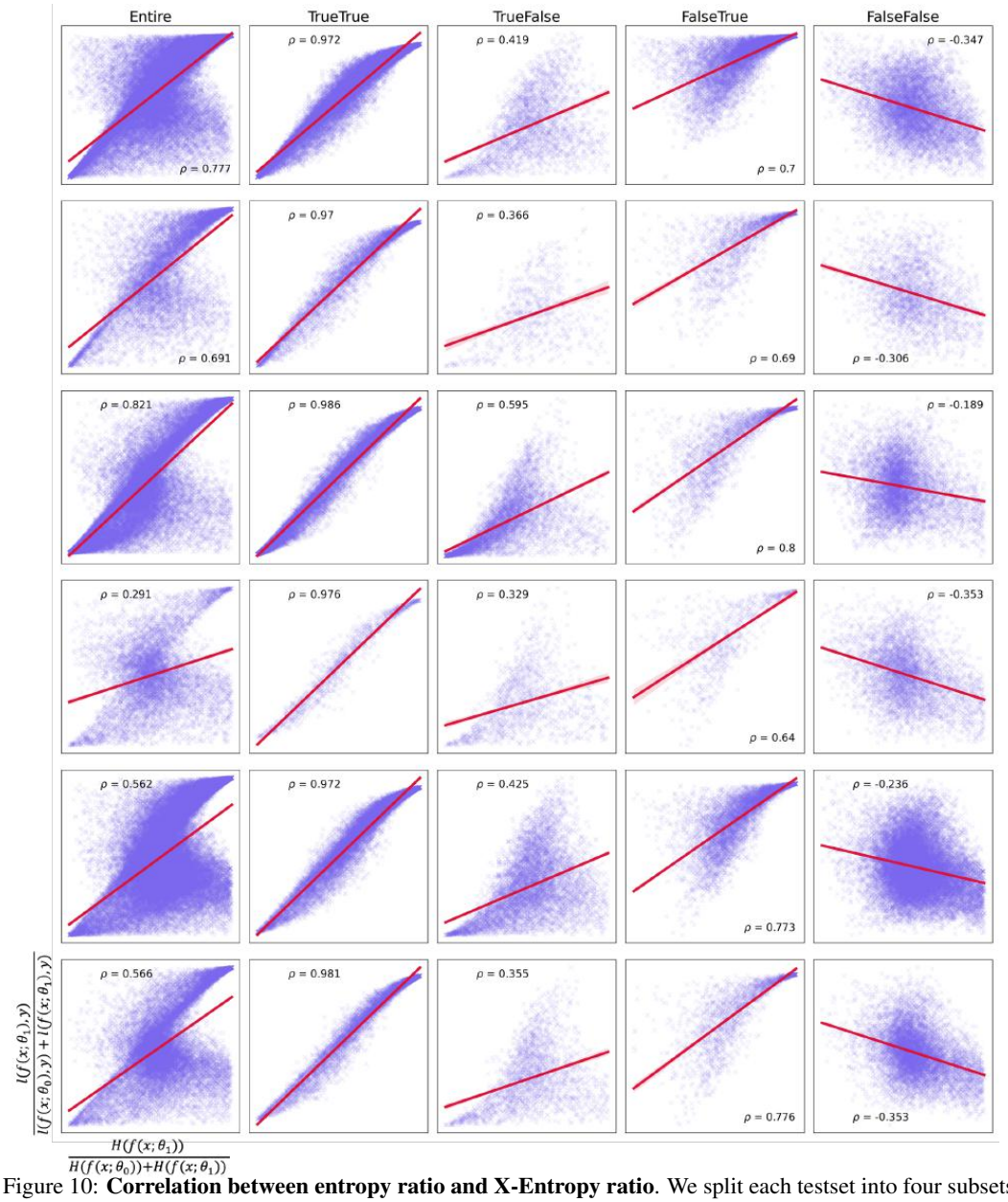

Figure 10: **Correlation between entropy ratio and X-Entropy ratio**. We split each testset into four subsets (`TrueTrue`, `TrueFalse`, `FalseTrue`, `FalseFalse`) based on the correctness of the two models' predictions. On each split and the entire dataset, we visualize scatter plots of entropy ratio (x-axis) and cross-entropy ratio (y-axis) computed by two models (zero-shot and fine-tuned) with corresponding Pearson correlation coefficients. Each row from top to bottom shows results from ImageNet, ImageNetV2, ImageNetR, ImageNetA, ImageNetSketch, and ObjectNet.

In Figure 10, we provide the extended results of Figure 2, showing the correlations between entropy ratio and X-entropy ratio, on the whole evaluation datasets of robust fine-tuning setup. The entropy ratio approximates the X-entropy ratio overall across datasets and sub-populations of each dataset, even though for the most challenging OOD, i.e., ImageNet-A, which is constructed with natural adversarial examples, there is a weak-yet-non-trivial correlation [59] between entropy and X-entropy. Moreover, we note that weight interpolation (or output ensemble) has an advantage that elicits the correct prediction by modifying the relative feature importance [75] even though two individual models fail to produce correct predictions (i.e., in the `FalseFalse` case), thereby expected to outperform the model selection method (See Table 6).

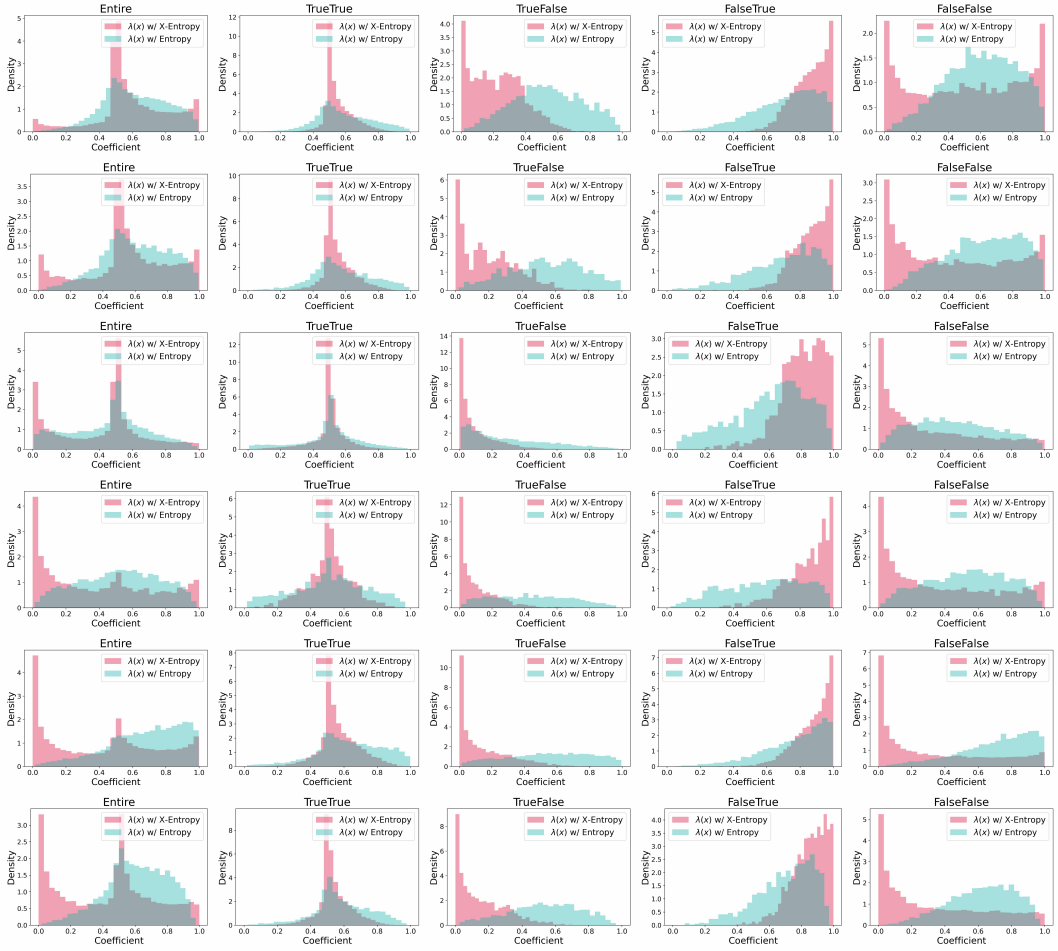

Figure 11: **Estimated coefficient density analysis on ImageNet distribution shift benchmarks with CLIP ViT-B/32**. We visualize histograms of estimated sample-wise interpolation coefficients by the expertise ratio, which is computed using X-entropy (Oracle; we can not access it in reality) and entropy as expertise metrics. Each column denotes the splits {`Entire`, `TrueTrue`, `TrueFalse`, `FalseTrue`, `FalseFlase`} of the test set, those are categorized by the correctness of zero-shot CLIP, and ImageNet fine-tuned CLIP's predictions. Each low denotes the result of the test set: from ImageNet, ImageNet-V2, ImageNet-R, ImageNet-A, ImageNet-Sketch, and ObjectNet. Entropy-based coefficient estimation shows remarkably good fitness in the `TrueTrue`, `FalseTrue` splits, and produces left-skewed distribution in the case of `TrueFalse`, which is desired to construct the interpolated model biased towards zero-shot model weight. Overall, except the `FalseFalse` case, the entropy-based coefficient estimation provides reasonable alternatives to X-entropy-based oracle coefficients.

We visualize the histogram of interpolation coefficients computed by entropy and X-entropy ratio in Figure 11. While the entropy-based coefficients quite diverge from the oracle X-entropy-based coefficients in some cases (e.g., `TrueFalse` and `FalseFalse` of ImageNet-Sketch and ObjectNet), the overall distributions of entropy-based coefficients show good fitness to X-entropy-based coefficients across datasets. This result supports using entropy as a proxy of X-entropy to estimate model expertise given unlabeled test-time input to determine the per-sample interpolation coefficient.

Figure 12: **Entropy comparison on ImageNet distribution shift with CLIP ViT-B/32**. Across IN, IN-V2, IN-R, IN-A, IN-S, and ObjNet from top to bottom, we visualize the final output entropy from each model on the entire dataset and four different splits based on the correctness of zero-shot (ZS) and fine-tuned (FT) models. Compared with individual models, weight averaging (WA) induces lower entropy overall, and our `DaWin` achieves the lowest entropy across all splits.

Our `DaWin` method is built on the correlation between entropy and X-entropy. While the analyses from Section 2 support using entropy as a proxy of X-entropy, it does not address how the entropies of interpolated final models behave. To explore this, we analyze the average entropy of interpolated models generated by `DaWin` in Figure 12. In almost all cases, we see that the simple weight averaging produces lower entropy compared to individual models, and our `DaWin` achieves the lower entropy in all cases. This indicates that `DaWin`'s expertise-based interpolation successfully weighs individual experts for accurate prediction and decreases per-sample entropy accordingly and supports our *sample-wise entropy valley* hypothesis to understand `DaWin`'s great performance gains.

A.4 Related Work

Robust fine-tuning aims to adapt a model on a target task while preserving the generalization capability learned during pre-training. A straightforward approach injects regularization into the learning objective. For example, Ju et al. [32] proposed a regularization motivated by Hessian analysis, Tian et al. [64, 65] devised a trainable projection method to constrain the parameter space, CAR-FT [46] devised the context-awareness regularization, and CaRot [51] introduced a regularization based on singular values. Another line of works modifies the training procedure to keep the pre-trained knowledge by decoupling the tuning of a linear head from the entire model [35], employing a data-dependent tunable module [38], utilizing bi-level optimization [9], or mimicking the pre-training procedure [17]. In contrast, weight interpolation approaches emerged as an effective yet efficient solution that conducts simple interpolation of individual model weights [28, 69, 70, 29]. Unlike existing works inducing a single interpolated model, we propose a dynamic interpolation method that produces per-sample models for better adaptation.

Model merging studies mainly focus on integrating multiple models trained on different tasks into a single model to create a versatile, general-purpose multi-task model. After some seminal works in the era of foundation models [26, 27, 31], numerous advances have been made those aim at reducing conflict between merged parameters [72, 76, 47], merging with weight disentanglement [68, 52, 30], optimizing interpolation coefficients on unlabeled test samples [74, 6] or learning additional modules generating per-sample/domain coefficients dynamically [7, 44, 63, 73]. While those methods provide huge performance gains compared with static methods such as [69], they all bring extra learnable modules and training. We focus on methods that do not induce extra complex training and devise a training-free dynamic merging method, `DaWin`, which seeks a good trade-off between efficiency and downstream performance.

A.5 Theoretical analysis

To understand the empirical success of `DaWin`, we present an analytic behavior of entropy-based dynamic weight interpolation in contrast to input-independent uniform weight interpolation, which yields a uniform interpolation coefficient [69].

Lemma A.1 (**Expert-biased weighting behavior of `DaWin`**). *Suppose we have M different models $\{f(\cdot;\theta_j)\}_{j=1}^M$ parameterized by $\{\theta_j\}_{j=1}^M$ with a homogeneous architecture defined by $f(\cdot)$. Let $\lambda(x) = (\lambda_1(x), \dots, \lambda_M(x))$ be the sample-wise interpolation coefficient vector given x. If the models that render a correct prediction for x always have smaller output entropy $H(\cdot)$ compared with the models rendering an incorrect prediction, `DaWin`'s interpolation coefficient of true experts for the sample x has always greater than $\frac{1}{M}$. That is*

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

where $[f(x;\theta)]_c$ indicates the probability mass for class c. Lemma A.5 implies that, under the entropy-dominancy assumption, `DaWin` always produces per-sample expert-biased coefficient vectors, which result in the interpolated models being biased towards true experts, i.e., models that produce correct prediction given x. This desirable behavior is aligned with the motivation of dynamic classifier selection discussed in Sec. 4, whereas `DaWin` conducts interpolation rather than selection. Meanwhile, as the number of models participating in interpolation increased, samples that at least one model correctly classifies also increased. Therefore, the coverage of `DaWin` for weighting the correct experts expands accordingly. This analysis endows a potential clue for remarkable gains observed in the multi-task setting, which conducts merging beyond two models (See Tab. 3).

Proof. The proof is very straightforward, given the assumption and definitions of problem setup.

$$\begin{aligned}
H(f(x;\theta_{j\in\mathcal{J}})) &\leq H(f(x;\theta_{k\notin\mathcal{J}})) && \text{(By assumption)} \\
\exp(-H(f(x;\theta_{j\in\mathcal{J}}))) &\geq \exp(-H(f(x;\theta_{k\notin\mathcal{J}}))) \\
\lambda_{j\in\mathcal{J}}(x) &\geq \lambda_{k\notin\mathcal{J}}(x) && \text{(By definition of } \lambda(x)) \\
\lambda_{j\in\mathcal{J}}(x) &\geq \frac{1}{M} && \text{(Given that } \textstyle\sum_{j=1}^M \lambda_j(x) = 1)
\end{aligned}$$

□

A.6 Limitation and future work

By following previous works [70, 27, 74], we limited our validation scope to an image classification of fully fine-tuned visual foundation models with restricted scale and architecture, e.g., CLIP ViT-{B/32, B/16, L/14}. Exploring `DaWin` on diverse model architecture [42, 19, 43], large-scale modeling setup [12], large language models [76, 15, 56], multimodal generative models [41, 3, 49], continual adaptation scenario [47], and parameter-efficient tuning regime [39, 25, 10] can be exciting future work directions.