Robust and Efficient Finetuning of Vision Foundation Models via Implicit Ensembling

Masih Aminbeidokhti* **Heitor Rapela Medeiros** **Eric Granger** **Marco Pedersoli**

LIVIA, ÉTS Montréal, Canada
International Laboratory on Learning Systems (ILLS)

Abstract

Robust finetuning of foundation models seeks to improve out-of-distribution (OOD) generalization during in-distribution (ID) adaptation. We revisit Mixout, a stochastic regularizer that intermittently replaces finetuned parameters with a reference anchor, and study its effectiveness for robust finetuning of vision foundation models. We reinterpret Mixout as a single-run, weight-sharing implicit ensemble and analyze its expected OOD error through a bias–variance–covariance–locality (BVCL) lens. This perspective reveals three key factors that govern the ID–OOD trade-off: the choice of masking anchor, the mask resampling frequency, and mask sparsity. Guided by this analysis, we propose GMixout, which generalizes Mixout in three ways. First, it explicitly controls the masking period through a resampling-frequency hyperparameter. Second, it replaces the fixed pretrained anchor with an exponential moving average snapshot that adapts throughout training. Third, it uses sparse kernels to update only a small subset of parameters at each iteration, introducing no inference-time overhead and enabling large-model finetuning on consumer-grade GPUs. Experiments across five vision benchmarks show that GMixout consistently outperforms Mixout, Model Soups, and strong parameter-efficient finetuning baselines under both covariate shift and class imbalance.

1 Introduction

A core assumption in standard machine learning is that training and test examples are independently and identically distributed (i.i.d.) [Vapnik, 1991]. In practice, this assumption is often violated: input distributions may shift across environments [Koh et al., 2021, Gulrajani and Lopez-Paz, 2021, Yang et al., 2023, Mayilvahanan et al., 2025], models may rely on spurious cues rather than the intended signal [Geirhos et al., 2020], and class frequencies can be highly imbalanced [Yang et al., 2023, Liu et al., 2019]. Foundation models have reshaped learning under distribution shift [Bommasani, 2021, Radford et al., 2021, Oquab et al., 2023]. Through large-scale and heterogeneous pretraining, they learn transferable representations that can generalize well, even in zero-shot settings [Radford et al., 2021]. Nevertheless, real-world deployments usually provide some domain-specific data, and finetuning remains the standard way to maximize downstream performance [Tian et al., 2023, Wortsman et al., 2022b]. This makes robust finetuning under distribution shift a central problem.

Existing robust finetuning methods, however, still have important limitations. Domain-invariant representation learning can be overly restrictive and may even underperform strong zero-shot baselines [Arjovsky et al., 2019, Sun and Saenko, 2016, Ganin et al., 2016, Mayilvahanan et al., 2025]. Augmentation-based methods require invariance-preserving transformations, which often depend on domain expertise and can be difficult or costly to design [Gontijo-Lopes et al., 2020, Liu et al., 2022, Aminbeidokhti et al., 2024, Gao et al., 2023]. Ensembling and Model Soups are among the most

*Correspondence to: masih.aminbeidokhti.1@ens.etsmtl.ca. Code at https://github.com/Masseeh/GMixout.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

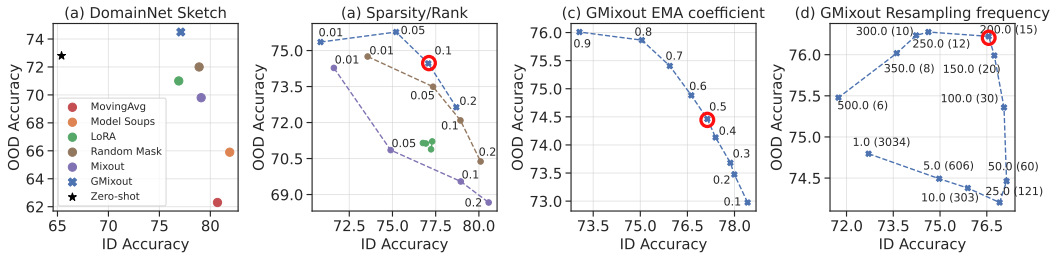


Figure 1: **OOD-ID accuracy curve on the DomainNet dataset. Models are trained on Sketch (ID) and then evaluated on Real, Painting, and Clipart (OOD) data.** (a) Each point represents a method, with ID accuracy on the x-axis and OOD accuracy on the y-axis. The ideal method should improve over the zero-shot baseline along both axes. (b) Mixout regulates the ID–OOD trade-off through mask sparsity. (c, d) GMixout extends Mixout with two mechanisms: (i) an EMA coefficient that updates the masking anchor, and (ii) a resampling frequency that determines the number of uncovered subnetworks during optimization (shown in parentheses in (d)). By controlling the variance and covariance of the expected test error, these hyperparameters significantly enhance OOD performance while maintaining competitiveness on ID.

reliable approaches for improving robustness, but their training cost scales poorly for large foundation models [Wortsman et al., 2022a, Rame et al., 2022]. Parameter-efficient finetuning (PEFT) methods, such as random masking [Xu and Zhang, 2024] and low-rank adaptation (LoRA) [Hu et al., 2022], reduce this cost by updating only a subset of parameters, but their OOD performance can still degrade under even modest shifts from the finetuning distribution [Shuttleworth et al., 2024, Han et al., 2024, Hu et al., 2022, Xu and Zhang, 2024].

Implicit-ensemble methods, which interpolate model weights along a single training trajectory, have shown strong performance on OOD benchmarks [Cha et al., 2021, Arpit et al., 2022, Izmailov et al., 2018]. Yet they are typically based on full-parameter finetuning, making them less practical for large foundation models, especially under limited computational resources. Our preliminary results in Figure 1(a) further show that, when models are finetuned on DomainNet-Sketch and evaluated on Real, Painting, and Clipart, MovingAvg improves in-distribution accuracy but remains substantially weaker than PEFT counterparts on OOD domains. Moreover, none of the considered robust finetuning methods surpasses the zero-shot model under distribution shift. These observations motivate us to revisit implicit ensembling in a parameter-efficient setting, with the goal of combining the practicality of PEFT with the robustness benefits of weight-space averaging.

To this end, we reinterpret Mixout as a parameter-efficient implicit ensemble method. Mixout applies a binary mask during training to randomly replace finetuned weights with their frozen pretrained counterparts, thereby regularizing optimization toward the initialization [Lee et al., 2019]. Its sparsity provides a simple mechanism for controlling the trade-off between in-distribution adaptation and robustness. However, although Mixout was effective in the original NLP setting, it is less competitive for vision foundation models. As shown in Figure 1(b), vanilla Mixout does not achieve the same ID–OOD curve as strong PEFT baselines such as LoRA and Random Mask.

We argue that this limitation is better understood from a weight-space ensemble perspective. Under this view, Mixout forms a single-run, weight-sharing implicit ensemble: each binary mask defines a different subnetwork, and training averages over many such subnetworks. Robustness, therefore, depends not only on sparsity, but also on the diversity and correlation among the sampled subnetworks, which determine the variance–covariance behavior of the ensemble [Dietterich, 2000, Allen-Zhu and Li, 2023]. We formalize this intuition using a bias–variance–covariance–locality (BVCL) decomposition [Rame et al., 2023]. This analysis highlights two important design choices that are overlooked in standard Mixout: the anchor weights and the mask resampling frequency. Standard Mixout always reverts to the fixed pretrained initialization and resamples the mask at every step, which can increase subnetwork correlation and bias learning toward an anchor that may no longer be optimal after downstream adaptation.

Guided by this analysis, we propose GMixout, a generalized Mixout framework that decouples sparsity, anchor selection, and mask resampling. First, GMixout introduces an explicit resampling-frequency hyperparameter that controls how often weight switching occurs, thereby regulating the

diversity and correlation of the implicit subnetworks; see Figure 1(d). Second, GMixout replaces the fixed pretrained anchor with an exponential moving average (EMA) reference that evolves during training, allowing the anchor to track a task-adapted solution while preserving the regularization effect of the pretrained initialization; see Figure 1(c). Finally, we implement GMixout with efficient sparse CUDA kernels [Gale et al., 2020], so that only a small fraction of parameters is updated at each step. The resulting method has no inference-time overhead and enables scalable finetuning of large foundation models on consumer-grade GPUs.

Our contributions are threefold. **(1)** We reinterpret Mixout as a single-run, weight-sharing implicit ensemble and analyze its ID–OOD behavior through a BVCL framework, identifying anchor weights, mask resampling frequency, and sparsity as key factors controlling the adaptation–robustness trade-off. **(2)** We introduce GMixout, which generalizes Mixout with an adaptive EMA anchor and an explicit resampling-frequency hyperparameter, together with a memory- and compute-efficient sparse-kernel implementation. **(3)** We empirically validate GMixout across a broad range of distribution shifts on vision benchmarks, showing consistent improvements over strong PEFT and ensemble baselines while substantially reducing training cost.

2 Related Work

Robust finetuning. A large body of work studies how to improve the robustness of finetuned foundation models under OOD shifts. In vision–language models, prior studies show that preserving aspects of the pretraining objective can substantially outperform standard cross-entropy finetuning [Goyal et al., 2023, Oh et al., 2024]. More broadly, several methods improve robustness by constraining finetuned weights to remain close to the pretrained initialization [Kumar et al., 2022, Xuhong et al., 2018, Tian et al., 2023, 2024]. Another effective direction is ensembling, where methods such as Model Soups aggregate multiple finetuned models to reduce variance and improve robustness [Dietterich, 2000, Rame et al., 2022, Allen-Zhu and Li, 2023]. However, explicit ensembles require multiple full finetuning runs, making them expensive at foundation-model scale. To reduce the cost of explicit ensembles, prior work has explored single-run alternatives, including weight averaging, mean-teacher updates, robust weight interpolation, and batch ensembles [Izmailov et al., 2018, Tarvainen and Valpola, 2017, Wortsman et al., 2022b, Oh et al., 2025, Zhu et al., 2024, Wen et al., 2020]. Although more efficient, these methods often trade off ensemble diversity or still rely on costly full-parameter updates [Bommasani, 2021]. GMixout is related to this line of work, particularly adaptive-reference methods such as Jang et al. [2025], but realizes a moving-anchor mechanism within a single sparse training run rather than by maintaining or finetuning multiple full-parameter models.

Parameter-efficient finetuning methods. Parameter-efficient finetuning (PEFT) reduces adaptation cost by updating only a small subset of parameters or by adding a small number of trainable components [Han et al., 2024]. Common PEFT methods include Adapters [Houlsby et al., 2019, Chen et al., 2022], which insert lightweight bottleneck modules into transformer blocks, and Low-Rank Adaptation (LoRA) [Hu et al., 2022, Liu et al., 2024], which represents weight updates using low-rank matrices and introduces no inference-time overhead after merging. More recently, Xu and Zhang [2024] showed that Random Masking, which updates only a fixed, randomly selected subset of parameters, can match or exceed LoRA while retaining the full expressive capacity of the pretrained model. Despite these advances, standard PEFT methods often provide limited OOD robustness gains over zero-shot baselines [Biderman et al., 2024, Shuttleworth et al., 2024].

Several recent methods specifically target robust LoRA adaptation. PACE [Ni et al., 2024] perturbs adapter features with multiplicative noise and enforces consistency across perturbed representations, while SoMA [Yun et al., 2025] and related approaches [Hu et al., 2024, Meng et al., 2024] use SVD-based initialization and selectively tune minor singular components. These methods are competitive on benchmarks such as DomainNet, but we find that they are less effective when the downstream distribution differs more substantially from the pretraining distribution, such as on iWildCam. One possible reason is that the low-rank parameterization can limit adaptation capacity [Hu et al., 2022], especially when stronger task-specific adaptation is required.

Prompt tuning is another related PEFT direction, where trainable continuous prompts are added to the model input or intermediate representations [Li et al., 2025, Zhou et al., 2022c]. It is particularly effective for few-shot adaptation in vision–language models [Zhou et al., 2022b]. In contrast, GMixout

is a general finetuning strategy for medium- and large-scale adaptation and is not restricted to the vision–language setting; see Appendix A.10 for further comparisons.

Although GMixout updates only a sparse subset of parameters at each iteration, it exposes the same overall parameter space as full finetuning over the course of training. We therefore view it as a PEFT method under the standard definition, which emphasizes reducing additional parameters, memory, and computational cost rather than strictly limiting the set of accessible parameters [Han et al., 2024].

Dropout. Dropout [Srivastava et al., 2014] and its variants [Ghiasi et al., 2018, Wan et al., 2013] are widely used regularization techniques that improve in-domain generalization by randomly deactivating subsets of activations during training. While these techniques are commonly used when training models from scratch, applying dropout directly to pretrained models can disrupt critical learned representations and undermine the benefits of pretraining [Zhang and Bottou, 2024]. To address this issue, Mixout [Lee et al., 2019] randomly replaces finetuned weights with their pretrained values during training, thereby regularizing the model toward the pretrained initialization.

GMixout goes beyond Mixout in four fundamental ways. First, GMixout introduces an explicit resampling-frequency hyperparameter that controls how often weight switching occurs, thereby regulating the diversity and correlation of the implicit subnetworks. Second, whereas Mixout uses a fixed pretrained anchor, GMixout employs a moving anchor that evolves during training, allowing the regularization target to adapt as the model improves. Third, Mixout and most dropout-based methods are designed for full-parameter finetuning, whereas GMixout combines mixout-style regularization with hardware-efficient sparse masked implementation [Gale et al., 2020]. This substantially reduces training cost and enables efficient finetuning on consumer-grade GPUs. Finally, Mixout was originally studied in NLP, where it achieved strong results; our preliminary language-model ablations are consistent with these findings. In contrast, this work focuses on vision benchmarks, where we find that vanilla Mixout is less effective than GMixout.

3 Background

Task and evaluation. Given training examples sampled from a distribution P_{id}, our goal is to finetune a foundation model $f : \mathbb{R}^d \to \mathcal{Y}$ that maps inputs $x \in \mathbb{R}^d$ to outputs $y \in \mathcal{Y}$. We evaluate the model using its standard *in-distribution* (ID) performance on test samples drawn from the same distribution P_{id}. In addition, we assess *out-of-distribution* (OOD) performance on test samples, indicating a different distribution $P_{\mathrm{ood}} \neq P_{\mathrm{id}}$. In practice, P_{ood} may differ from P_{id} due to (i) covariate shift in $P(x)$, (ii) class imbalance in $P(y)$, or (iii) changes in $P(y \mid x)$ arising from spurious correlations that are absent at test time. During training, we assume no access to samples from P_{ood} and therefore minimize empirical risk using only data drawn from P_{id}. For a loss function ℓ and a foundation model with parameter vector Φ this objective is given by:

$$\mathcal{L}(\Phi) \;=\; \mathbb{E}_{(\boldsymbol{x},y)\sim P_{\mathrm{id}}}\big[\ell\big(f(x), y\big)\big]. \tag{1}$$

Mixout. Let Φ_0 denote the frozen pretrained model weights, and let Δ be learnable parameters of the same shape and initial value as Φ_0. During training with Mixout [Lee et al., 2019], at each iteration, a binary mask is generated, where each element is independently sampled from Bernoulli$(1-p)$, where p is the masking probability, with the sparsity of $s = 1-p$. According to the mask, the corresponding coordinates of the learnable parameter Δ are then swapped with their pretrained counterparts Φ_0, effectively freezing these coordinates from gradient updates. To ensure that the expected output of layer activations during training matches their output at test time, Mixout rescales Δ by $(1-p)^{-1}$ during training. This process yields the following risk minimization:

$$\min_{\Delta} \mathbb{E}\big[\mathcal{L}\big(\tfrac{1}{1-p}((\mathbf{I}-\mathbf{M})\odot\Phi_0 + \mathbf{M}\odot\Delta)\big)\big], \tag{2}$$

where $\mathbf{I}$ is the identity matrix, $\odot$ is the Hadamard product, and the expectation is taken over the random masks $\mathbf{M}$.

4 Proposed Method

4.1 Mixout Error Analysis

As shown in Lee et al. [2019], Mixout can be interpreted as an adaptive form of weight decay, where the effective regularization strength increases with the mask probability p, biasing the solution

toward the reference parameters Φ_0 (see Appendix A.1 for details). While informative, this viewpoint does not capture how the *unmasked subnetworks* interact during training. We therefore develop a complementary perspective based on implicit ensembling in weight space. Consider M finetuned models, each trained under an identically distributed (i.d.) learning procedure $l_{P_{\text{id}}}$. The expected error of a model with weights $\Phi_{\text{WA}} = \frac{1}{M}\sum_{m=1}^{M}\Phi_m$ with respect to the joint distribution $L_{P_{\text{id}}}^{M} = \{l_{P_{\text{id}}}^{(m)}\}_{m=1}^{M}$ can be decomposed into [Rame et al., 2022][2]:

$$\mathbb{E}_{L_{P_{\text{id}}}^{M}}\left[\mathcal{L}_{\text{ood}}(\Phi_{\text{WA}})\right] = \mathbb{E}_{(x,y)\sim P_{\text{ood}}}\left[\text{bias}^2(x,y) + \frac{1}{M}\,\text{var}(x) + \frac{M-1}{M}\,\text{cov}(x)\right] + O(\bar{\Pi}^2). \tag{3}$$

Here, the bias is the same as that of each of its i.d. members. The total variance is split into the variance of each of its i.d. members divided by M and a covariance term. The locality term $\bar{\Pi}^2$ quantifies the expected squared distance between individual weights and their mean. Eq. (3) highlights that averaging M models reduces variance by a factor of M, but the resulting OOD error also depends on controlling covariance and locality (in addition to bias).

Implications for Mixout. Now consider the Mixout training procedure. Let $\{\Phi_i\}_{i=1}^{I}$ denote the sequence of parameters associated with the masked updates over I training iterations, and define the corresponding weight average $\Phi_{\text{Mixout}} = \frac{1}{I}\sum_{i=1}^{I}\Phi_i$. Since standard Mixout resamples the binary mask at every iteration, it effectively corresponds to an ensemble size $M = I$ in the BVCL view. Combining the ensemble decomposition in Eq. 3 with the adaptive ℓ_2-penalty view of Mixout reveals the following two key limitations of standard Mixout.

(1) Covariance–adaptation trade-off via p. In standard Mixout, the primary mechanism to reduce covariance across subnetworks is increasing the mask probability p, which decreases the overlap between subnetworks and can reduce correlation (Figure 1(b)). However, according to Eq. 4, a large p also strengthens the implicit pull toward Φ_0, limiting downstream adaptation and making optimization brittle when a small subset of parameters is effectively updated.

(2) Fixed anchor can hinder adaptation. Mixout always reverts masked parameters to a fixed anchor Φ_0. This is beneficial when Φ_0 is already well-aligned with the downstream task, but can be restrictive when the task requires substantial adaptation (e.g., when zero-shot performance is poor as in iWildCam benchmark). In such cases, a fixed anchor increases bias and prevents the implicit ensemble from tracking a better reference throughout training.

These findings are further supported by the empirical results in Section 5.2 and motivate three fundamental modifications to Mixout, which we formalize next.

4.2 From Mixout to GMixout

GMixout generalizes Mixout along three dimensions that directly address the limitations identified in the previous section. First, it introduces a resampling-frequency hyperparameter k, which controls how often the binary mask is refreshed; standard Mixout is recovered as the special case $k = 1$. By decoupling mask refresh from the total training iterations, it provides a direct handle on the diversity–covariance trade-off of sampled subnetworks, without relying solely on increasing p. Second, GMixout replaces the fixed pretrained anchor Φ_0 with an adaptive reference updated via an EMA. Concretely, we partition training into *episodes* indexed by i, where each episode consists of k consecutive optimization steps performed under a fixed random mask. Let Δ_i denote the parameter accumulated within episode i. At the end of the episode, the anchor is updated as $\Phi_i = \lambda\Phi_{i-1} + (1-\lambda)\Delta_i$ with $\lambda \in [0, 1)$, and the Δ is reset (i.e., the next episode starts from $\Delta_{i+1} \leftarrow \Phi_i$). This moving anchor allows the implicit ensemble to track a task-adapted reference, mitigating the bias induced by reverting to a static Φ_0 when substantial adaptation is required. The resulting risk minimization objective follows Eq. 2, with Φ_0 replaced by the episode-dependent anchor Φ_i. In practice, k is chosen as a function of the total training iterations; for clarity, we parameterize it via E, the total number of episodes. Finally, unlike the standard Mixout implementation which uses a dense matrix for masking, GMixout keeps $(\mathbf{S}, \Delta_{\mathbf{S}})$ for the unmasked indices $\mathbf{S}$ and reconstructs $\mathbf{M} \odot \Delta$ with sparse CUDA kernels [Gale et al., 2020]. At inference, we merge Δ into Φ_0, therefore adding no extra cost on the forward pass. See Algorithm 1 for the GMixout procedure and its differences from standard Mixout.

[2]Refer to Appendix A.2 for the detailed formulation.

Algorithm 1 GMixout and Mixout Training Procedure

Initialize model parameters Φ_0, mask probability p, total number of episodes E, EMA coefficient λ, total training iterations T.
Define $\Delta \leftarrow \Phi_0$, $k \leftarrow 1$ or $k \leftarrow \lfloor \frac{T}{E} \rfloor$
for each iteration $iter$ **do**
  if $iter \bmod k = 0$ **then**
    $\Phi_i \leftarrow \lambda \Phi_{i-1} + (1-\lambda)\Delta$
    $\Delta \leftarrow \Phi_i$
    $\mathbf{M} \sim \text{Bernoulli}(1-p)$
  end if
  Update parameters according to Eq. 2.
end for

5 Results and Discussion

Models and Datasets. We use a ViT-B/16 [Dosovitskiy et al., 2020] pretrained on OpenAI's CLIP [Radford et al., 2021] dataset as our main vision backbone, due to its strong robustness. However, as shown in our ablation studies (Section 5.2), GMixout is not tied to this initialization and generalizes across alternative pretrained weights, including both vision-only and language-only models. We evaluate GMixout under two major distribution-shift settings: covariate shift and long-tailed class imbalance. For covariate shift, we consider `ImageNet-1K` [Russakovsky et al., 2015] with OOD evaluation on `ImageNet-V2` [Recht et al., 2019], `ImageNet-R` [Hendrycks et al., 2021a], `ImageNet-Sketch` [Wang et al., 2019], and `ImageNet-A` [Hendrycks et al., 2021b]; `DomainNet` [Peng et al., 2019], following prior work [Kumar et al., 2022, Tian et al., 2024] by training on one domain and evaluating on the remaining domains, repeated for each source domain; and `iWildCam` [Beery et al., 2021] from WILDS [Koh et al., 2021], where we use the standard split with an ID validation set drawn from diverse training environments and an OOD test set from unseen environments. For long-tailed class imbalance, we evaluate on `CIFAR100-LT` [Cao et al., 2019] and `ImageNet-LT` [Liu et al., 2019]. Full dataset descriptions are provided in Appendix A.14.

Baselines. Across all benchmarks, we compare GMixout against LoRA, Random Masking, Mixout, full-parameter finetuning (Full-FT), its moving average variant (MovingAvg [Arpit et al., 2022]), and Model Soups constructed from five independently finetuned models. These methods represent widely adopted and competitive approaches on robustness-focused vision benchmarks [Rame et al., 2022, Shi et al., 2024, Yang et al., 2023, Koh et al., 2021]. Additionally, for DomainNet and iWildCam, we include task-specific state-of-the-art methods, including PACE [Ni et al., 2024], SoMA [Yun et al., 2025], DoRA [Liu et al., 2024], LP-FT [Kumar et al., 2022], AdamSPD [Tian et al., 2024], CLIPOOD [Shu et al., 2023], CaRot [Oh et al., 2024], and FLYP [Goyal et al., 2023]. See Appendix A.12 for detailed descriptions of all baselines.

Preliminary experiments on DomainNet suggest that, for PEFT methods, updating approximately 10% of parameters provides a favorable adaptation–robustness trade-off. For GMixout, we fix the EMA coefficient to $\lambda = 0.5$ and tune the resampling frequency k based on ID validation performance. A comprehensive analysis of the hyperparameter sensitivity of GMixout is provided in Section 5.2. Across all experiments, we replace every linear layer with its corresponding PEFT variant. Additional implementation details are deferred to Appendix A.15.

5.1 Main Results

Medium-scale Benchmarks. We begin by evaluating performance on medium-scale benchmarks (40k–500k samples), including DomainNet, iWildCam, CIFAR100-LT and ImageNet-LT (Table 1, 2). Across these datasets, the zero-shot CLIP baseline serves as a proxy for the proximity between the downstream task and the pretraining distribution: DomainNet exhibits the closest alignment, while iWildCam is the most distant.

On DomainNet, GMixout achieves the best overall performance with an average accuracy of 72.6%. This corresponds to improvements of +5.3 points over Mixout, +2.6 over zero-shot CLIP, +7.6 over

Model Soups, and +1.1 over the strongest baseline, while consistently leading across all domains (Real, Sketch, Painting, and Clipart). On iWildCam, where zero-shot performance is substantially lower (ID: 11.5, OOD: 12.7), GMixout yields a large performance gain, achieving the best results both on ID and OOD macro-F1 score and substantially improving over Mixout in both settings. Among baseline methods, CaRot, PACE, and SoMA are the closest competitor, but with important limitations: (i) CaRot requires full-parameter finetuning of both the vision and language encoders, incurring significantly higher computational and memory costs; and (ii) the performance of all three methods strongly depends on the alignment between the pretrained model and the downstream task, which limits their effectiveness on iWildCam.

On long-tailed benchmarks (ImageNet-LT and CIFAR100-LT; Table 2), GMixout achieves the best performance in both overall and few-shot settings. The gains are particularly pronounced on ImageNet-LT, with improvements of +6.1 and +11.6 over Mixout in the overall and few-shot regimes, respectively, highlighting its robustness on underrepresented classes. Full results across all imbalance categories are reported in Table 12 in the Appendix.

Table 1: OOD accuracy of GMixout and SOTA methods on four DomainNet domains (train on one, evaluate on the others), and macro-F1 on ID and OOD test sets of iWildCam (WILDS).

Method	**DomainNet (Training Domains)**					**iWildCam**	
	Real	Sketch	Painting	Clipart	Avg.	ID	OOD
Zero-shot	65.8	72.8	71.4	70.4	70.1	11.5	12.7
Full-FT	60.2	62.4	61.6	61.9	61.5	46.5	36.4
MovingAvg	60.2	62.3	62.0	61.8	61.6	46.6	34.2
Model Soups	64.3	65.9	64.6	65.2	65.0	46.8	<u>37.6</u>
Linear Probing	62.2	64.4	60.2	62.7	62.4	31.6	24.7
LoRA	66.5	71.0	69.0	69.5	69.0	49.7	35.9
Random Mask	66.8	72.0	68.9	69.8	69.4	47.9	35.2
Mixout	66.5	69.8	65.0	67.7	67.3	43.5	32.9
CaRot	67.8	73.7	<u>72.0</u>	71.7	71.3	43.8	31.2
FLYP	57.9	63.9	65.7	67.5	63.7	42.5	30.4
LP-FT	59.3	61.2	60.3	59.3	60.0	47.4	31.8
AdamSPD	66.2	68.5	66.3	67.6	67.1	47.4	36.1
CLIPOOD	60.2	62.6	61.9	61.7	61.6	41.6	24.7
DoRA	66.0	70.6	68.6	69.2	68.6	50.4	35.7
PACE	<u>67.5</u>	<u>74.0</u>	71.6	<u>72.9</u>	<u>71.5</u>	43.1	32.3
SoMA	67.3	73.8	70.8	<u>72.9</u>	71.2	<u>50.8</u>	36.2
GMixout	**68.3**	**75.3**	**73.5**	**73.4**	**72.6**	**51.8**	**38.4**

Table 2: Balanced accuracy of GMixout and SOTA methods on the long-tailed benchmarks ImageNet-LT and CIFAR100-LT.

Method	**ImageNet-LT**		**CIFAR100-LT**	
	All	Few	All	Few
Zero-shot	68.2	66.4	62.0	57.3
Full-FT	73.1	60.4	79.6	69.3
MovingAvg	73.0	63.1	81.0	71.9
Model Soups	76.0	65.5	<u>82.1</u>	73.0
Linear Probing	74.2	67.4	70.0	60.4
LoRA	76.7	67.4	81.1	<u>75.7</u>
Random Mask	<u>76.8</u>	<u>70.4</u>	81.5	74.6
Mixout	70.9	59.3	81.5	72.9
GMixout	**77.0**	**70.9**	**82.2**	**76.0**

Table 3: Per-step training overhead of GMixout and baselines with a batch size of 16.

Method	#Params ↓ (M)	FLOPs ↓ (T)	Latency ↓ (ms)	GPU VRAM ↓ (G)
ViT-B/16				
Full-FT	85.5	1.7	70.4	3.2
MovingAvg	85.5	1.7	70.4	3.2
Model Soups	427.5	8.5	70.4	16.0
Linear Probing	0.5	0.6	22.5	0.5
LoRA	8.7	1.3	77.6	2.3
Random Mask	9.0	1.1	112.0	2.5
GMixout	9.0	1.1	123.1	2.6
ViT-L/14@336				
LoRA	28.3	12.7	690	22.5
GMixout	31.0	14.2	1038	23.2

Large-scale Benchmarks. Table 4 reports results on ImageNet-1K (ID) and four natural distribution shifts derived from ImageNet (OOD). As expected, methods that fine-tune on ImageNet-1K substantially improve ID accuracy over zero-shot CLIP: all the baselines reach around 83–84% ID accuracy, compared to 68.2% for zero-shot. However, the picture changes under a distribution shift. Zero-shot CLIP remains strong on IN-R (76.4) and IN-A (51.6), while full fine-tuning methods (Full-FT, MovingAvg), including Mixout, degrade to the 40s on IN-A. Model Soups partially restore

robustness but require costly multi-model ensembles. PEFT methods offer the best trade-off: slightly lower ID accuracy than Model Soups but consistently higher OOD averages than zero-shot. Among them, GMixout achieves the highest OOD average (60.7), corresponding to relative gains of +0.7 points over Model Soups, +1.3 over LoRA, +0.3 over Random Mask, and +11.6 over Mixout.

Experiments on medium- and large-scale datasets show GMixout is most effective at medium scale, with its advantage persisting, though narrowing, at larger scales. Figure 2 further reinforces this trend on ImageNet: across all ImageNet subsamples (25–200 shots per class), GMixout maintains a strong ID–OOD balance. Unlike Model Soups, which favors ID accuracy, or MovingAvg, which fluctuates, PEFT methods, especially GMixout, scale smoothly with data. Even at 25 shots, GMixout maintains strong OOD accuracy, and with more data, it matches or surpasses the strongest alternatives.

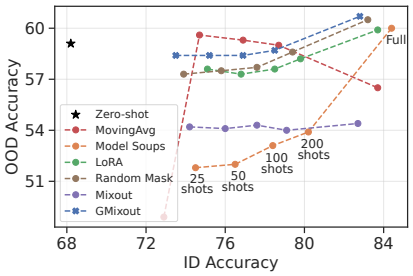


Figure 2: OOD-ID accuracy on ImageNet with varying training set sizes.

Method	ID	OOD				
	IN-1k	IN-V2	IN-R	IN-Sketch	IN-A	Avg.
Zero-shot	68.2	62.0	**76.4**	46.6	**51.6**	59.1
Full-FT	83.7	73.9	65.0	47.1	40.0	56.5
MovingAvg	83.7	73.9	64.9	47.0	40.0	56.5
Model Soups	**84.4**	**74.9**	68.9	**50.9**	45.3	60.0
Linear Probing	77.0	66.4	70.0	43.7	45.7	56.5
LoRA	83.7	74.0	69.3	49.6	46.6	59.9
Random Mask	83.2	74.0	70.4	49.9	47.7	60.5
Mixout	82.7	72.6	62.7	45.0	37.4	54.4
GMixout	82.8	73.6	71.6	50.4	47.2	**60.7**

Table 4: Accuracy on ImageNet-1k (ID) and natural ImageNet distribution shifts (OOD).

Memory-Compute Benchmarks. Table 3 summarizes the per-step training overhead across two architectures (ViT-B/16 and ViT-L/14). Model Soups requires training multiple full models before merging; accordingly, we report its training complexity assuming five parallel runs, corresponding to roughly five dense models. Although GMixout has the same total number of trainable parameters as Full-FT, in our implementation only the unmasked parameters participate in backpropagation, substantially reducing computation and memory per step and making GMixout feasible on limited hardware. LoRA achieves lower latency through better hardware utilization; however, on consumer GPUs, memory remains the primary bottleneck, with usage comparable to GMixout. Finally, all methods share the same inference cost as a single dense baseline, since the learnable parameters in GMixout and LoRA can be merged into Φ_0 after training (see Appendix A.11 for full details).

5.2 Ablation Studies

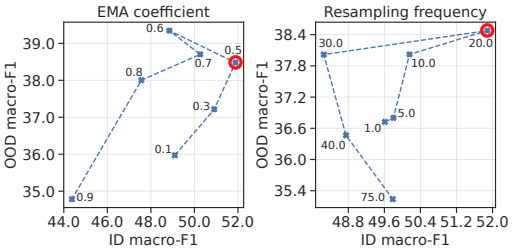


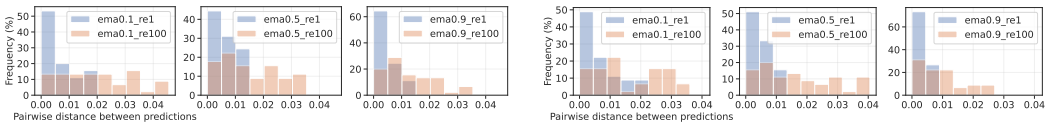
Figure 3: OOD-ID curve given variable λ (left) and k (right) on the iWildCam dataset.

Method	Worst acc.
Full-FT	63.6
Model Soups	62.5
LoRA	58.5
Random Mask	56.8
Mixout	67.6
DoRA	64.5
GMixout (Ours)	**69.0**

Table 5: Worst-group accuracy on the CivilComments benchmark for language understanding.

Hyperparameter sensitivity. To further examine how GMixout's hyperparameters affect the ID–OOD curve, we replicate the analysis from Figure 1(c,d) on iWildCam, a downstream task farther from the pretraining distribution than DomainNet, as reflected by its lower zero-shot OOD performance (12.7 vs. 70.1). In Figure 3, we vary the EMA coefficient λ and resampling frequency k separately, keeping mask sparsity fixed at 0.1. Unlike DomainNet, increasing λ does not monotonically improve iWildCam performance, and overly large values reduce both ID and OOD accuracy. This supports our analysis in Section 4.1: stale anchors can impede adaptation when downstream data are poorly aligned with pretraining. The effect of k is also dataset-dependent, however the values that maximize ID performance also tend to optimize maximize OOD performance, making k easy to tune using ID

validation alone. Accordingly, we fix the EMA coefficient to $\lambda = 0.5$ and tune k once for DomainNet and once for iWildCam, and reuse the DomainNet setting for ImageNet, CIFAR, and CivilComments.



(a) Sketch (ID)

(b) Clipart (OOD)

Figure 4: Frequencies of prediction diversities [Aksela, 2003] between two subnetworks obtained from a single training run on DomainNet-Sketch and evaluated on Sketch as ID and Clipart as OOD, under $\lambda \in \{0.1, 0.5, 0.9\}$ and $k \in \{1, 100\}$.

Prediction diversity. Diversity among ensemble members is a key driver of ensemble effectiveness [Dietterich, 2000, Rame et al., 2022, Fort et al., 2019, Izmailov et al., 2018]. Figure 4 measures pairwise prediction diversity between GMixout checkpoints on DomainNet-Sketch using the ratio-error metric [Aksela, 2003]. For $k = 100$, we extract the weights from the last ten re-masking steps of GMixout. To ensure fair comparison, we extract an equivalent number of checkpoints for the $k = 1$ setting over the same training interval. Across ID (Figure 4a) and OOD (Figure 4b) evaluations, $k = 1$ yields near-zero diversity for many checkpoint pairs, indicating that subnetworks collapse to similar behavior; increasing k substantially increases diversity. Diversity is highest at $\lambda = 0.1$ and decreases as λ grows, consistent with stronger EMA anchoring suppressing variation across subnetworks (refer to Figure 9 in the Appendix for results on all domains).

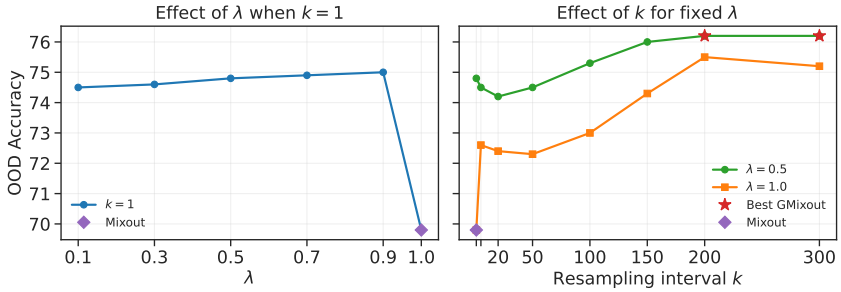


Figure 5: OOD performance of GMixout on DomainNet-Sketch under different choices of k and λ. The standard Mixout setting corresponds to $k = 1$ and $\lambda = 1$.

Synergy of GMixout components. To disentangle the contributions of GMixout's components, we conduct ablation experiments on DomainNet-Sketch by varying one hyperparameter at a time while fixing the other to its standard Mixout value, with $k = 1$ and $\lambda = 1$ corresponding to standard Mixout. Figure 5 presents the results. The left panel isolates the effect of EMA anchoring by varying λ while fixing $k = 1$, showing that EMA anchoring alone improves performance over standard Mixout by mitigating anchor staleness. Similarly, the right panel isolates the effect of the mask resampling interval by varying k while fixing $\lambda = 1$, demonstrating consistent improvements with longer intervals, which reduce correlations among the implicit subnetworks. Combining both modifications yields further gains: the default GMixout configuration, $(k = 100, \lambda = 0.5)$, achieves 75.3, outperforming either component in isolation. Increasing the resampling interval to $k = 200$ or $k = 300$, with $\lambda = 0.5$, further improves performance to 76.2.

To further distinguish the contribution of masked subnetwork training from EMA anchoring, we introduce a no-mask episodic EMA baseline that follows the same episodic anchor-update schedule as GMixout but keeps all parameters active throughout training. This baseline achieves 68.3, compared with 75.3 for the default GMixout configuration, a difference of 7.0 percentage points. This substantial gap demonstrates that EMA anchoring alone cannot explain GMixout's performance gains and underscores the importance of masked subnetwork training.

Taken together, these results highlight the distinct contributions of masking, EMA anchoring, and extended mask resampling intervals. They further suggest that the improvements from EMA anchoring

and longer resampling intervals are complementary, with their combination yielding greater gains than either modification alone.

Table 6: OOD accuracy on DomainNet across pretrained weight sources and model scales. Results are reported for models trained on Real/Sketch and evaluated on the remaining domains.

Method	ViT-B/16 IN-21k		ViT-B/16 DINO		ViT-L/14@336 CLIP	
	Real	Sketch	Real	Sketch	Real	Sketch
Zero-shot	-	-	-	-	74.2	79.2
Full-FT	55.1	58.0	-	-	-	-
MovingAvg	55.0	59.0	-	-	-	-
Model Soups	56.4	59.4	-	-	-	-
Linear Probing	49.0	57.8	38.4	45.4	71.0	73.3
LP-FT	-	-	50.3	51.6	-	-
LoRA	54.9	60.0	49.8	52.0	74.7	75.7
Random Mask	54.2	59.2	49.2	51.3	73.6	76.2
Mixout	55.8	59.4	48.6	50.9	-	-
GMixout	**65.0**	**60.3**	**52.3**	**53.6**	**76.2**	**81.4**

Beyond standard CLIP backbones. We further evaluate GMixout by varying the pretrained weight source and model scale. For vision-only ViT-B/16 backbones pretrained on ImageNet-21k [Ridnik et al., 2021] and DINO [Caron et al., 2021], with classification heads initialized from linear probing [Kumar et al., 2022], GMixout remains competitive despite these models being less robust than CLIP-based ones. As shown in Table 6, it outperforms Model Soups on ImageNet-21k by 8.6 points on Real and 0.9 points on Sketch, and improves over the best competing DINO baseline by 2.0 and 1.6 points, respectively. To test scalability, we also finetune ViT-L/14@336 CLIP on a single 24GB RTX 3090, where full finetuning is infeasible. GMixout achieves the best performance, surpassing the strongest PEFT baseline by 1.5 points on Real and 5.2 on Sketch, while also improving over zero-shot by 2.2 points on Sketch.

Language understanding. Although our evaluation focuses primarily on vision benchmarks, GMixout makes no vision-specific assumptions and can be applied to any pretrained model. To illustrate this broader applicability, we include a preliminary language experiment on CivilComments [Duchene et al., 2023], a standard benchmark for subpopulation shift [Koh et al., 2021, Yang et al., 2023]. We follow the WILDS data splits [Koh et al., 2021] and hide demographic labels during training. Following Yang et al. [2023], we finetune $\text{BERT}_{\text{BASE}}$ Devlin et al. [2018] and report worst-group accuracy, a standard measure of robustness to subpopulation imbalance. We apply PEFT layers to all linear layers in the transformer. As shown in Table 5, unlike the vision benchmarks, the standard Mixout is already the strongest finetuning baseline on CivilComments; nevertheless, GMixout further improves worst-group accuracy to 69.0%, outperforming Mixout by 1.4 points and achieving the best overall robustness.

6 Conclusions

We reinterpret Mixout as a single-run implicit ensemble, identifying anchor choice, resampling frequency, and mask sparsity as key drivers of robustness. Building on this, we propose GMixout, which combines an adaptive EMA anchor, controlled resampling, and an efficient sparse implementation. Across five OOD benchmarks, GMixout consistently improves robustness while preserving ID performance, outperforming Mixout, prior PEFT methods, and Model Soups, while also reducing training overhead. Overall, it offers a practical and parameter-efficient approach to robust fine-tuning of vision foundation models.

Acknowledgements. This work was supported by Distech Controls Inc., the Natural Sciences and Engineering Research Council of Canada, the Digital Research Alliance of Canada, and MITACS.

References

Matti Aksela. Comparison of classifier selection methods for improving committee performance. In *International Workshop on Multiple Classifier Systems*, pages 84–93. Springer, 2003.

Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Uuf2q9TfXGA.

Masih Aminbeidokhti, Fidel A Guerrero Pena, Heitor Rapela Medeiros, Thomas Dubail, Eric Granger, and Marco Pedersoli. Domain generalization by rejecting extreme augmentations. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2215–2225, 2024.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Devansh Arpit, Huan Wang, Yingbo Zhou, and Caiming Xiong. Ensemble of averages: Improving model selection and boosting performance in domain generalization. *Advances in Neural Information Processing Systems*, 35:8265–8277, 2022.

Sara Beery, Arushi Agarwal, Elijah Cole, and Vighnesh Birodkar. The iwildcam 2021 competition dataset. *arXiv preprint arXiv:2105.03494*, 2021.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John Patrick Cunningham. LoRA learns less and forgets less. *TMLR*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=aloEru2qCG. Featured Certification.

Rishi Bommasani. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision (ECCV)*, volume 8694 of *Lecture Notes in Computer Science*, pages 446–461. Springer, 2014.

Manh-Ha Bui, Toan Tran, Anh Tran, and Dinh Phung. Exploiting domain-specific features to enhance domain generalization. *Advances in Neural Information Processing Systems*, 34:21189–21201, 2021.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. *Neurips*, 34:22405–22418, 2021.

Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *NeurIPS*, 35:16664–16678, 2022.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3606–3613, June 2014.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Corentin Duchene, Henri Jamet, Pierre Guillaume, and Reda Dehak. A benchmark for toxic comment classification on civil comments dataset. *arXiv preprint arXiv:2301.11125*, 2023.

Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

Trevor Gale, Matei Zaharia, Cliff Young, and Erich Elsen. Sparse gpu kernels for deep learning. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–14. IEEE, 2020.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 17(1):2096–2030, 2016.

Irena Gao, Shiori Sagawa, Pang Wei Koh, Tatsunori Hashimoto, and Percy Liang. Out-of-domain robustness via targeted augmentations. In *International conference on machine learning*, pages 10800–10834. PMLR, 2023.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. *Neurips*, 31, 2018.

Raphael Gontijo-Lopes, Sylvia J Smullin, Ekin D Cubuk, and Ethan Dyer. Affinity and diversity: Quantifying mechanisms of data augmentation. *arXiv preprint arXiv:2002.08973*, 2020.

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.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019.

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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021a.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15262–15271, 2021b.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Jiajun Hu, Jian Zhang, Lei Qi, Yinghuan Shi, and Yang Gao. Learn to preserve and diversify: Parameter-efficient group with orthogonal regularization for domain generalization. In *European Conference on Computer Vision*, pages 198–216. Springer, 2024.

Pavel Izmailov, Dmitrii Podoprikhin, T. Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *Conference on Uncertainty in Artificial Intelligence*, 2018. URL https://api.semanticscholar.org/CorpusID:3833416.

Dong-Hwan Jang, Sangdoo Yun, and Dongyoon Han. Model stock: All we need is just a few fine-tuned models. In *ECCV*, pages 207–223. Springer, 2025.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. Wilds: A benchmark of in-the-wild distribution shifts, 2021.

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. URL https://openreview.net/forum?id=UYneFzXSJWh.

Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. Mixout: Effective regularization to finetune large-scale pretrained language models. *arXiv preprint arXiv:1909.11299*, 2019.

Fei-Fei Li, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5400–5409, 2018.

Zongqian Li, Yixuan Su, and Nigel Collier. A survey on prompt tuning, 2025. URL https://arxiv.org/abs/2507.06085.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. In *ICML*, 2024.

Ziquan Liu, Yi Xu, Yuanhong Xu, Qi Qian, Hao Li, Rong Jin, Xiangyang Ji, and Antoni B Chan. An empirical study on distribution shift robustness from the perspective of pre-training and data augmentation. *arXiv preprint arXiv:2205.12753*, 2022.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, pages 2537–2546, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Prasanna Mayilvahanan, Roland S. Zimmermann, Thaddäus Wiedemer, Evgenia Rusak, Attila Juhos, Matthias Bethge, and Wieland Brendel. In search of forgotten domain generalization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Fk3eod9aaD.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. *Advances in Neural Information Processing Systems*, 37: 121038–121072, 2024.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *ICLR*, 2021. URL https://openreview.net/forum?id=37nvvqkCo5.

Yao Ni, Shan Zhang, and Piotr Koniusz. Pace: Marrying generalization in parameter-efficient fine-tuning with consistency regularization. *Advances in Neural Information Processing Systems*, 37: 61238–61266, 2024.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)*, pages 722–729, Dec 2008. doi: 10.1109/icvgip.2008.47.

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*, 37:12677–12707, 2024.

Changdae Oh, Yixuan Li, Kyungwoo Song, Sangdoo Yun, and Dongyoon Han. Dawin: Training-free dynamic weight interpolation for robust adaptation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=L8e7tBf4pP.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C.Ṽ. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Neurips*, 32, 2019.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019.

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.

Alexandre Rame, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. Diverse weight averaging for out-of-distribution generalization. *NeurIPS*, 35: 10821–10836, 2022.

Alexandre Rame, Kartik Ahuja, Jianyu Zhang, Matthieu Cord, Léon Bottou, and David Lopez-Paz. Model ratatouille: Recycling diverse models for out-of-distribution generalization. 2023.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019.

Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Jiang-Xin Shi, Tong Wei, Zhi Zhou, Jie-Jing Shao, Xin-Yan Han, and Yu-Feng Li. Long-tail learning with foundation model: Heavy fine-tuning hurts. In *ICML*, 2024.

Yuge Shi, Jeffrey Seely, Philip HS Torr, Narayanaswamy Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. *arXiv preprint arXiv:2104.09937*, 2021.

Yang Shu, Xingzhuo Guo, Jialong Wu, Ximei Wang, Jianmin Wang, and Mingsheng Long. Clipood: Generalizing clip to out-of-distributions. In *International conference on machine learning*, pages 31716–31731. PMLR, 2023.

Reece Shuttleworth, Jacob Andreas, Antonio Torralba, and Pratyusha Sharma. Lora vs full fine-tuning: An illusion of equivalence. *arXiv preprint arXiv:2410.21228*, 2024.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958, 2014.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCVW*, pages 443–450. Springer, 2016.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *NeurIPS*, 30, 2017.

Junjiao Tian, Yen-Cheng Liu, James S Smith, and Zsolt Kira. Fast trainable projection for robust fine-tuning. *Neurips*, 36:11374–11393, 2023.

Junjiao Tian, Chengyue Huang, and Zsolt Kira. Rethinking weight decay for robust fine-tuning of foundation models. *Advances in Neural Information Processing Systems*, 37:22418–22440, 2024.

Vladimir Vapnik. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4, 1991.

Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *ICML*, pages 1058–1066. PMLR, 2013.

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in neural information processing systems*, 32, 2019.

Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. *arXiv preprint arXiv:2002.06715*, 2020.

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 *ICML*, pages 23965–23998. PMLR, 2022a.

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, 2022b.

Jing Xu and Jingzhao Zhang. Random masking finds winning tickets for parameter efficient fine-tuning. In *ICML*, 2024. URL `https://openreview.net/forum?id=VrwIrAa1Lc`.

LI Xuhong, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *International conference on machine learning*, pages 2825–2834. PMLR, 2018.

Yuzhe Yang, Haoran Zhang, Dina Katabi, and Marzyeh Ghassemi. Change is hard: A closer look at subpopulation shift. In *ICML*, 2023.

Seokju Yun, Seunghye Chae, Dongheon Lee, and Youngmin Ro. Soma: Singular value decomposed minor components adaptation for domain generalizable representation learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 25602–25612, 2025.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Jianyu Zhang and Léon Bottou. Fine-tuning with very large dropout. *arXiv preprint arXiv:2403.00946*, 2024.

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4396–4415, 2022a.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16816–16825, 2022b.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022c.

Beier Zhu, Jiequan Cui, and Hanwang Zhang. Robust fine-tuning of zero-shot models via variance reduction. *Advances in Neural Information Processing Systems*, 37:76967–76990, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The implications of GMixout are illustrated in Figure 1, which appears in the introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of GMixout are discussed throughout the paper, including in the related work, proposed method, results sections.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: Our work introduces a BVCL decomposition as a diagnostic tool, supported by empirical evidence rather than intended as a formal proof.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section A.15.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All of the datasets are publicly available (see Section A.14). The source code is provided in the supplementary materials.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section A.15 and Section A.12 and Section 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Section A.9 in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section A.11.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work is based on public data.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Section A.16

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We only use public datasets.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The provided source code is reproducible and easy to follow.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: We used LLM only for writing, editing, or formatting purposes.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

A Appendix

Appendix Contents

Section	Page
A.1 Regularization View for Mixout	24
A.2 Bias–Variance–Covariance–Locality Decomposition	24
A.3 Measure forgetting	25
A.4 Parameters budget	25
A.5 Data Augmentation	25
A.6 Extended training on ImageNet-1k	25
A.7 WISE-FT	26
A.8 Robustness under common corruptions and perturbations	26
A.9 Statistical significance analysis	26
A.10 Comparison with Prompt-tuning	27
A.11 Benchmark details	28
A.12 Additional baselines	28
A.13 Extended Related Work	29
A.14 Datasets	29
A.15 Implementation details	29
A.16 Broader impact	30
A.17 Detailed results	30

A.1 Regularization View for Mixout

Assume that the loss function $\mathcal{L}$ is strongly convex in a neighborhood of Φ_0 and consider the Mixout parameterization as in Eq. 2. Lee et al. [2019] by a first-order expansion around Φ_0 shows the following inequality:

$$\mathbb{E}\big[\mathcal{L}\big(\Phi_0 + \tfrac{1}{1-p}(M \odot \Delta)\big)\big] \;\geq\; \mathcal{L}(\Phi_0) \;+\; \frac{\mu\,(p)}{2(1-p)}\,\|\Delta \;-\Phi_0\|_2^2, \tag{4}$$

where $\mu > 0$. Eq. 4 shows that, as the mask probability p increases, the effective ℓ_2-penalty grows, biasing the solution toward Φ_0.

A.2 Bias–variance–covariance–locality Decomposition

Consider M finetuned models, each trained under an identically distributed (i.d.) learning procedure $l_{P_{\text{id}}}$. The expected error of a model with weights $\Phi_{\text{WA}} = \frac{1}{M}\sum_{m=1}^{M}\Phi_m$ with respect to the joint distribution $L_{P_{\text{id}}}^{M} = \{l_{P_{\text{id}}}^{(m)}\}_{m=1}^{M}$ was decomposed in Rame et al. [2023] into:

$$\begin{aligned}
\mathbb{E}_{L_{P_{\text{id}}}^{M}}\big[\mathcal{L}_{\text{ood}}(\Phi_{WA})\big] &= \mathbb{E}_{(x,y)\sim P_{\text{ood}}}\Big[\text{bias}^2(x,y) + \frac{1}{M}\text{var}(x) + \frac{M-1}{M}\text{cov}(x)\Big] + O(\bar{\Pi}^2),\\
\textit{where } \text{bias}(x,y) &= y - \bar{f}(x),\\
\textit{and } \text{var}(x) &= \mathbb{E}_{l_{P_{\text{id}}}^{i}}\Big[\big(f(x,\Phi_i) - \bar{f}(x)\big)^2\Big],\\
\textit{and } \text{cov}(x) &= \mathbb{E}_{l_{P_{\text{id}}}^{i},l_{P_{\text{id}}}^{j}}\Big[\big(f(x,\Phi_i) - \bar{f}(x)\big)\big(f(x,\Phi_j) - \bar{f}(x)\big)\Big],\\
\textit{and } \bar{\Pi}^2 &= \mathbb{E}_{L_{P_{\text{id}}}^{M}}\big[\max_i \|\Phi_i - \Phi_{WA}\|_2^2\big],
\end{aligned} \tag{5}$$

where $\bar{f}(x) = \mathbb{E}[f(x,\Phi_i)]$ denotes the expected prediction of f, $\text{bias}(x,y)$ and $\text{var}(x)$ are the bias and variance of the models with respect to a sample (x,y), and $\text{cov}(x)$ is the prediction covariance between two ensemble members whose weights are averaged. The locality term $\bar{\Pi}^2$ represents the expected squared maximum distance between individual weights and their weight average.

A.3 Measure forgetting

To study how finetuning alters the backbone's representations, we evaluate transfer performance on five classification tasks: Flowers102 [Nilsback and Zisserman, 2008], Food101 [Bossard et al., 2014], Oxford Pets [Parkhi et al., 2012], DTD [Cimpoi et al., 2014], and Caltech101 [Li et al., 2007]. In this setup, the finetuned backbone is frozen and paired with a new text-encoder-based classification head. Figure 6 shows that LoRA best preserves prior knowledge, followed by GMixout—an important property, as the goal is to adapt to new tasks without forgetting what has already been learned.

A.4 Parameters budget

Figure 7 compares the effect of the learnable parameter ratio—controlled by rank in LoRA and by mask sparsity in Random Mask, Mixout, and GMixout—on the ID–OOD curve. We evaluate ratios from 1% to 10% of learnable parameters. As the ratio increases, all methods trade off OOD for ID, with LoRA being the least sensitive. Across both domains, GMixout consistently yields higher OOD accuracy under any parameter budget while maintaining competitive ID accuracy on Real. On Sketch, it requires lower sparsity to match Random Mask and Mixout. Overall, optimizing about 10% of parameters is a good heuristic for balancing ID and OOD performance.

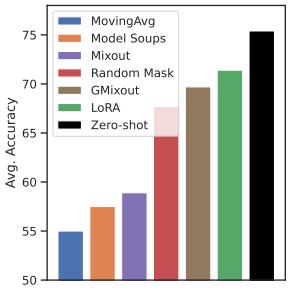


Figure 6: Average transfer accuracy on five out-of-task datasets after finetuning the methods on ImageNet-1k.

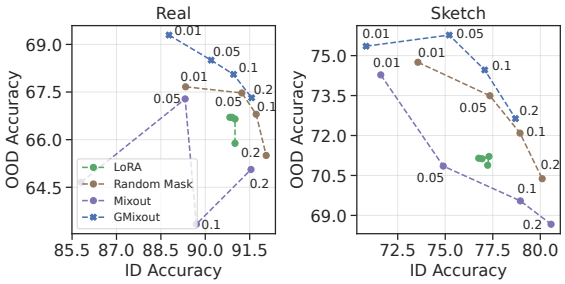


Figure 7: OOD-ID curve given the variable trainable parameters budget for PEFT methods.

A.5 Data Augmentation

GMixout differs fundamentally from data augmentation as it operates in weight space rather than input space. While techniques like RandAugment [Cubuk et al., 2020] improve robustness by applying random input perturbations—such as cropping, color jittering, and geometric distortions—often relying on domain-specific assumptions about label-preserving transformations, GMixout is instead a domain-agnostic parameter-space regularizer that stochastically reverts subsets of weights to an anchor while controlling subnetwork diversity. As shown in Table 7, data augmentation improves robustness across all baselines, with particularly pronounced gains for GMixout when training on the Sketch domain, yielding a 0.6-point improvement over the standard pipeline—likely due to Sketch's similarity to other rendition domains such as Painting and Clipart. Overall, augmentation broadens the training distribution, whereas GMixout constrains and diversifies optimization, making the two approaches complementary, consistent with our RandAugment results in Table 8.

A.6 Extended training on ImageNet-1k

An easy way to improve downstream performance is to train for longer. However, this often reduces robustness, particularly when starting from a strong foundation model. In Table 9, we extend training on ImageNet-1k for 30 epochs. As shown, LoRA and Random Mask show no ID improvement, suggesting overfitting, while GMixout continues to improve ID accuracy. OOD accuracy drops across all methods, but GMixout is the most resilient. A similar trend holds for out-of-task datasets, where all methods experience decreased performance.

Table 7: OOD accuracy on DomainNet Real and Sketch with applying **RandAugment** during finetuning.

Method	**ID**	**OOD**				**ID**	**OOD**			
	Real	Clipart	Sketch	Painting	Avg.	Sketch	Real	Clipart	Painting	Avg.
Zero-shot	86.1	69.3	62.0	66.1	65.8	65.4	82.9	69.3	66.1	72.8
Full-FT	91.7	68.0	55.0	61.2	61.4	80.7	65.9	68.3	54.5	63.0
MovingAvg	91.7	67.9	55.0	61.2	61.4	80.7	66.3	68.4	54.7	63.1
Model Soups	92.2	70.8	59.6	64.7	65.0	**81.7**	70.6	71.1	59.5	67.0
Linear Probing	90.4	66.8	59.0	63.6	63.1	74.2	72.5	64.8	57.7	65.0
LoRA	90.6	72.0	63.1	66.3	67.1	76.0	79.0	71.4	65.7	72.0
Random Mask	91.5	72.3	63.1	66.3	67.3	78.0	79.6	72.5	66.0	72.7
Mixout	92.0	69.7	57.9	63.3	63.6	78.9	76.5	71.9	63.3	70.6
GMixout	90.9	72.6	63.5	66.8	**67.7**	76.2	82.4	73.5	69.3	**75.0**

Table 8: OOD accuracy on DomainNet (Real and Sketch) and micro-F1 on iWildCam with **WISE-FT** with merging coefficient 0.5.

Method	**DomainNet**				**iWildCam**	
	Real		Sketch			
	ID	OOD	ID	OD	ID	OOD
Zero-shot	86.1	65.8	69.3	72.8	11.5	12.7
Full-FT	92.0	68.9	80.4	75.2	38.6	31.1
MovingAvg	92.0	68.9	80.5	75.2	38.7	31.2
Model Soups	**92.2**	69.9	**80.7**	75.7	40.1	32.3
LoRA	89.0	69.0	73.1	75.6	22.4	23.2
Random Mask	90.6	69.7	76.3	76.1	42.2	32.7
Mixout	89.5	69.3	78.2	75.7	38.8	31.0
GMixout	90.1	**69.9**	76.0	**76.4**	**43.1**	**34.2**

A.7 WISE-FT

An effective way to combine the downstream performance of finetuned models with the robustness of zero-shot models is to ensemble their weights [Wortsman et al., 2022b]. This technique, known as WISE-FT, can be applied to any finetuning method. By adjusting a merging coefficient, WISE-FT trades downstream adaptability for robustness. Table 8 reports results on DomainNet (trained on Real/Sketch) and iWildCam with a coefficient of 0.5. On DomainNet, where the zero-shot model is already strong, WISE-FT improves all baselines. However, on iWildCam, where CLIP performs poorly in zero-shot, WISE-FT provides no benefit—in fact, the baselines perform better without it. In contrast, GMixout (without WISE-FT) consistently improves OOD performance regardless of the zero-shot model's strength.

A.8 Robustness under common corruptions and perturbations

We further compare GMixout with baselines under synthetic corruptions using `CIFAR100-C` [Hendrycks and Dietterich, 2019], which introduces 15 corruption types at the highest severity level (5). Table 10 summarizes the results.

On CIFAR100, where CLIP's transfer to low-resolution images is more limited, GMixout achieves the second-best performance under both clean and corrupted settings, trailing Model Soups by only 0.2 and 0.9 points, respectively, while outperforming all other methods.

A.9 Statistical significance analysis

We evaluate GMixout across three random seeds on three benchmarks: DomainNet-Real, DomainNet-Sketch, and iWildCam. As shown in Figure 8, GMixout consistently outperforms the compared

Table 9: Accuracy on ImageNet-1k (ID) and four natural distribution shifts (OOD) under extended finetuning, along with transfer performance on five out-of-task datasets.

Method	ID	OOD					Out-of-Task
	IN-1k	IN-V2	IN-R	IN-Sketch	IN-A	Avg.	Avg. of 5 Datasets
Zero-shot	68.2	62.0	76.4	46.6	51.6	59.1	75.4
10 Epochs							
LoRA	83.7	74.0	69.3	49.6	46.6	59.9	71.4
Random Mask	83.2	74.0	70.4	49.9	47.7	60.5	67.7
Mixout	82.7	72.6	62.7	45.0	37.4	54.4	58.9
GMixout	82.8	73.6	71.6	50.4	47.2	60.7	69.7
30 Epochs							
LoRA	83.5	74.0	68.9	49.6	47.2	59.9	70.2
Random Mask	83.2	73.8	69.1	49.2	47.3	59.9	66.4
Mixout	81.8	71.2	58.4	41.3	32.4	50.8	55.5
GMixout	83.1	73.8	70.4	50.1	46.1	60.1	68.0

Table 10: Accuracy on CIFAR100 (ID) and CIFAR100-C (Corruption).

Method	ID	Corruption
Zero-shot	64.3	33.3
Full-FT	89.7	57.6
MovingAvg	89.9	58.7
Model Soups	**91.1**	**63.0**
Linear Probing	74.1	42.9
LoRA	89.9	55.0
Random Mask	90.2	56.1
Mixout	88.7	55.7
LP-FT	74.6	43.0
AdamSPD	88.2	56.5
CLIPOOD	86.7	53.2
DoRA	88.2	54.4
PACE	87.4	54.7
SoMA	87.8	55.2
GMixout	<u>90.9</u>	<u>62.1</u>

baselines while maintaining small variance across seeds. This suggests that the gains are not due to a particular randomness but reflect stable improvements across datasets.

A.10 Comparison with Prompt-tuning

In this section, we compare GMixout with prompt-tuning baselines, including CoOp [Zhou et al., 2022c] and CoCoOp [Zhou et al., 2022b], on the few-shot learning benchmark. CoOp replaces hand-crafted prompts in vision–language models, such as CLIP, with learnable context vectors trained end-to-end, thereby improving performance but showing limited generalization to unseen classes. CoCoOp extends this by employing a meta-network to generate image-conditioned context tokens, enhancing generalization at the cost of slower inference. Following the evaluation protocol of Zhou et al. [2022b], we sample 16 shots per class from ImageNet-1k for training and evaluate on both the ID validation set and four natural distribution shifts derived from ImageNet-1k. We use ViT-B/16 with CLIP weights as the image encoder. We use the mask sparsity of 0.01 for GMixout. As shown in Table 11, GMixout consistently outperforms prompt-tuning baselines on both ID and OOD, except on IN-A.

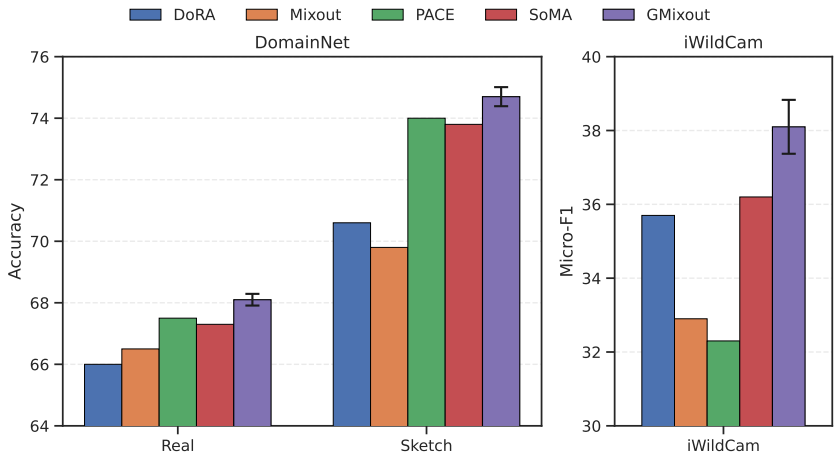


Figure 8: OOD accuracy on DomainNet (trained on Real/Sketch) and OOD micro-F1 on iWildCam, across 3 seeds.

Table 11: Accuracy on few-shot ImageNet-1k (ID, 16 shots per class) and four natural distribution shifts (OOD).

Method	**ID**	**OOD**				
	IN-1k	IN-V2	IN-R	IN-Sketch	IN-A	Avg.
Zero-shot	68.2	62.0	76.4	46.6	51.6	59.1
CoOp [Zhou et al., 2022c]	71.5	64.2	75.2	48.0	49.7	59.3
CoCoOp [Zhou et al., 2022b]	71.0	64.0	76.2	48.7	50.6	59.9
GMixout	**73.4**	**65.8**	**77.3**	**49.0**	49.3	**60.3**

A.11 Benchmark details

All benchmarks are conducted using the PyTorch v2.7.1 profiling tools [Paszke et al., 2019]. Experiments are run on an NVIDIA RTX 3090 (24 GB VRAM). We set the LoRA rank to $r = 64$, and use a mask sparsity of $s = 0.1$ for Random Masking, and GMixout. For ViT-L/14, full finetuning is no longer feasible on a single consumer GPU, so we compare only against LoRA.

The standard Mixout method does not reduce memory usage, since it zeroes gradients only during the backward pass while still requiring all gradients to be stored. As a result, its per-step cost is comparable to full fine-tuning. However, our sparse-kernel implementation is compatible with Mixout and can achieve similar computational benefits to GMixout.

A.12 Additional baselines

In this work, we include a unified set of baselines that are widely recognized for achieving state-of-the-art performance under different types of distribution shift. In particular, for corruption and covariate-shift benchmarks—especially when the target distribution departs substantially from the pretraining data or when datasets are large—Model Soups [Wortsman et al., 2022a] constitute a strong and competitive reference [Rame et al., 2022, Fort et al., 2019]. Several recent works [Jang et al., 2025, Rame et al., 2023] explicitly acknowledge Model Soups as a robustness upper bound and aim to approximate their performance while reducing the associated computational cost. Below, we provide additional details on the state-of-the-art baselines included for the DomainNet, iWildCam, and CIFAR100 benchmarks.

- **DoRA** [Liu et al., 2024] is a variant of LoRA that decomposes pretrained weights into magnitude and direction, applying low-rank updates primarily to the directional component to better approximate full finetuning capacity without inference-time overhead.

- **LP-FT** [Kumar et al., 2022] is a two-stage finetuning strategy that first performs linear probing and then finetunes the model, motivated by the observation that direct finetuning can distort pretrained representations and harm OOD generalization.
- **AdamSPD** [Tian et al., 2024] improves robustness by introducing selective projection decay into Adam-style optimizers, strongly regularizing sensitive parameter subspaces while allowing others to adapt during finetuning.
- **CLIPOOD** [Shu et al., 2023] is a CLIP-specific robust finetuning method that preserves OOD performance by combining margin-based objectives with temporal ensembling to stabilize adaptation under distribution shift.
- **CaRot** [Oh et al., 2024] is a vision–language finetuning approach that jointly targets OOD robustness and confidence calibration, improving both predictive accuracy and calibration under domain shift.
- **FLYP** [Goyal et al., 2023] (*Finetune Like You Pretrain*) adapts vision–language models by continuing contrastive learning with class prompts, aligning finetuning with the original pretraining objective to improve robustness.

A.13 Extended Related Work

Domain generalization (DG) aims to train models on limited source domains that generalize to unseen targets [Zhou et al., 2022a]. Existing approaches can be broadly categorized into: (i) feature alignment methods that learn domain-invariant representations [Ganin et al., 2016, Li et al., 2018, Sun and Saenko, 2016], (ii) invariant predictor learning that enforces consistency of classifiers across domains [Arjovsky et al., 2019, Shi et al., 2021], (iii) distributionally robust optimization that focuses on worst-case performance [Sagawa et al., 2019], (iv) meta-learning strategies for rapid adaptation to domain shifts [Bui et al., 2021], and (v) data augmentation methods that increase training diversity [Aminbeidokhti et al., 2024, Cubuk et al., 2020, Hendrycks et al., 2019, Zhang et al., 2017]. Despite strong theoretical motivation, many DG methods do not consistently outperform empirical risk minimization (ERM) under fair model selection [Gulrajani and Lopez-Paz, 2021, Koh et al., 2021], whereas ensemble-based approaches often yield substantial improvements [Wortsman et al., 2022a, Rame et al., 2023].

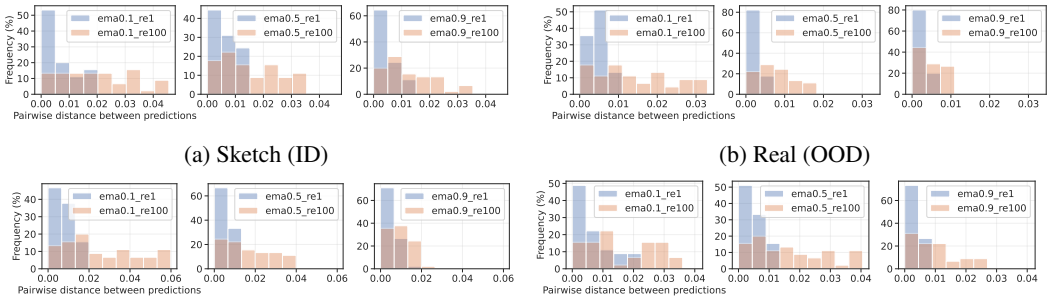
Although related in spirit, DG differs from robust fine-tuning due to its underlying assumptions. DG trains on multiple labeled source domains and evaluates on a held-out target domain. In contrast, robust fine-tuning assumes access to a single or mixed (unlabeled) source domain and evaluates robustness to distribution shifts within the same task. While DG was highly effective in the ImageNet pretraining era, robust fine-tuning is more suitable for evaluating modern foundation models, which already exhibit strong zero-shot generalization on DomainNet-like benchmarks. Notably, recent follow-up work to DomainBed [Gulrajani and Lopez-Paz, 2021] adopts a similar protocol for evaluating such models [Mayilvahanan et al., 2025].

A.14 Datasets

We evaluate on `ImageNet-1K` [Russakovsky et al., 2015], with OOD tests on `ImageNet-V2` [Recht et al., 2019], `ImageNet-R` [Hendrycks et al., 2021a], `ImageNet-Sketch` [Wang et al., 2019], and `ImageNet-A` [Hendrycks et al., 2021b]. From the WILDS benchmark [Koh et al., 2021], we include `iWildCam` [Beery et al., 2021], which contains animal images with variations in camera types, backgrounds, and illumination. `DomainNet` [Peng et al., 2019] is also considered, where one domain is treated as ID and the remaining three as OOD. For long-tailed class imbalance, we evaluate on `CIFAR100-LT` [Cao et al., 2019] and `ImageNet-LT` [Liu et al., 2019]. In both datasets, the number of training samples per class follows a long-tailed distribution that decreases exponentially, while the test set is balanced with a uniform distribution across classes.

A.15 Implementation details

As the feature extractor, we use CLIP's vision encoder [Radford et al., 2021] implemented with a ViT [Dosovitskiy et al., 2020]. For the classification head g, a prototypical head is adopted with ℓ_2-normalized features and class prototypes, and a temperature scaling on the logits. The parameters w are initialized via the CLIP text encoder. The full set of parameters is denoted as $\Phi = \{\theta, w\}$.



(a) Sketch (ID)

(b) Real (OOD)

(c) Paint (OOD)

(d) Clipart (OOD)

Figure 9: Frequencies of prediction diversities [Aksela, 2003] between two subnetworks obtained from a single training run on DomainNet-Sketch and evaluated on the rest of domains, under $\lambda \in \{0.1, 0.5, 0.9\}$ and $k \in \{1, 100\}$.

We train all methods with the AdamW optimizer [Loshchilov and Hutter, 2019], using weight decay 0.1. Batch size is 128 for all datasets except ImageNet-1k, where it is 512. To initialize the linear classifier in CLIP-based ImageNet-1k experiments, we use the prompt templates provided in Radford et al. [2021]. For all other datasets, we adopt the template "a photo of a `<class>`" to obtain the corresponding embeddings from the CLIP text encoder. For each baseline, we tune learning rates from $\{1e-2, 1e-3, 1e-4, 1-5\}$ with a cosine decay scheduler that anneals to 0 after a 1-epoch warmup. Training runs for 10 epochs on ImageNet-1k and DomainNet, and 20 epochs on all other datasets. The best checkpoint is selected using each dataset's in-domain validation set [Gulrajani and Lopez-Paz, 2021].

For long-tail benchmarks, we adopt the *Logit Adjustment* (LA) loss [Menon et al., 2021], which mitigates head-class bias by adding a label-dependent offset to the logits:

$$\ell_{LA}(y, f(\boldsymbol{x})) = -\log \frac{\exp(g_y(\boldsymbol{x}) + \log \pi_y)}{\sum_{y' \in \mathcal{C}} \exp(f_{y'}(\boldsymbol{x}) + \log \pi_{y'})}, \tag{6}$$

where $\pi \in \Delta_y$ are estimates of the class priors $P(y)$ based on the empirical class frequencies on the training data D. In a recent study, Shi et al. [2024] observed that starting from CLIP pretrained weights, the LA loss alone is insufficient to achieve strong performance.

A.16 Broader impact

This work advances research on robust finetuning under distribution shift, a challenge that commonly arises in real-world applications such as medical imaging, wildlife monitoring, and autonomous driving. By improving robustness without increasing inference cost or training complexity, GMixout supports the deployment of more reliable models in settings where data distributions evolve over time. Improved robustness can also help reduce performance disparities across underrepresented groups or conditions, contributing to fairer and more dependable AI systems.

A.17 Detail results

In this section, we show detailed results of our main experiments. Table 12 reports the full results on long-tail benchmarks, Tables 13 and 14 report the full results on the Real, Sketch, Clipart, and Painting domains of DomainNet, treating each as ID with their corresponding OOD domains. Table 15 provides the detailed ID–OOD results corresponding to Figure 2, using the ImageNet-1k subsampled training sets with 25, 50, 100, and 200 images per class, respectively.

Table 12: Balanced accuracy of GMixout and SOTA methods on the long-tailed benchmarks ImageNet-LT and CIFAR100-LT.

Method	**ImageNet-LT**				**CIFAR100-LT**			
	All	Many	Med.	Few	All	Many	Med.	Few
Zero-shot	68.2	69.4	67.8	66.4	62.0	66.2	61.8	57.3
Full-FT	73.1	80.3	70.8	60.4	79.6	88.1	79.9	69.3
MovingAvg	73.0	79.0	71.0	63.1	81.0	89.4	80.6	71.9
Model Soups	76.0	81.5	74.5	65.5	82.1	**89.9**	82.2	73.0
Linear Probing	74.2	77.8	73.3	67.4	70.0	77.2	71.1	60.4
LoRA	76.7	**81.6**	75.3	67.4	81.1	85.7	81.0	75.7
Random Mask	76.8	80.7	75.5	70.4	81.5	87.4	81.7	74.6
Mixout	70.9	76.5	69.7	59.3	81.5	88.1	82.3	72.9
GMixout	**77.0**	80.3	**76.2**	**70.9**	**82.2**	87.3	**82.5**	**76.0**

Table 13: OOD accuracy on DomainNet Real and Sketch.

Method	**ID**	**OOD**				**ID**	**OOD**			
	Real	Clipart	Sketch	Painting	Avg.	Sketch	Real	Clipart	Painting	Avg.
Zero-shot	86.1	69.3	62.0	66.1	65.8	65.4	82.9	69.3	66.1	72.8
Full-FT	91.6	67.0	53.3	60.2	60.2	80.7	65.7	67.7	53.7	62.4
MovingAvg	91.6	67.0	53.4	60.2	60.2	80.7	65.7	67.8	53.5	62.3
Model Soups	92.4	70.5	58.2	64.1	64.3	81.9	69.0	70.7	58.0	65.9
Linear Probing	90.7	66.0	57.9	62.7	62.2	75.7	71.4	65.4	56.4	64.4
LoRA	90.9	71.7	62.3	65.6	66.5	76.9	77.8	71.5	63.6	71.0
Random Mask	91.7	72.1	62.2	66.1	66.8	78.9	78.8	72.9	64.6	72.0
Mixout	89.7	71.5	62.4	65.7	66.5	79.1	75.2	71.3	62.8	69.8
GMixout	91.0	73.2	64.1	67.6	**68.3**	77.0	82.6	74.1	69.3	**75.3**

Table 14: OOD accuracy on DomainNet Painting and Clipart.

Method	**ID**	**OOD**				**ID**	**OOD**			
	Painting	Clipart	Sketch	Real	Avg.	Clipart	Sketch	Painting	Real	Avg.
Zero-shot	69.3	69.3	62.0	82.9	71.4	69.5	62.0	66.1	82.9	70.4
Full-FT	83.6	62.1	51.2	71.6	61.6	85.0	57.9	55.0	72.7	61.9
MovingAvg	83.6	63.0	51.5	71.7	62.0	84.9	58.0	54.9	72.5	61.8
Model Soups	84.7	65.2	54.9	73.6	64.6	86.3	61.6	58.7	75.1	65.2
Linear Probing	80.6	58.4	53.2	68.9	60.2	81.2	57.8	56.8	73.6	62.7
LoRA	81.5	68.1	61.4	77.6	69.0	81.8	64.0	64.1	80.4	69.5
Random Mask	82.8	68.4	59.9	78.3	68.9	83.2	64.5	64.5	80.4	69.8
Mixout	82.5	65.1	55.8	74.3	65.0	84.9	62.3	60.7	77.6	66.9
GMixout	81.3	71.9	66.0	82.8	**73.5**	81.8	67.6	69.0	83.7	**73.4**

Table 15: ID and OOD results on ImageNet-1k subsampled training datasets (25, 50, 100, and 200)-shots.

Method	**ID**	**OOD**				
	IN-1k	IN-V2	IN-R	IN-Sketch	IN-A	Avg.
Zero-shot	68.2	62.0	76.4	46.6	51.6	59.1
25 Shots						
Full-FT	72.7	63.4	59.3	40.2	29.4	48.1
MovingAvg	72.9	63.5	61.6	41.1	29.6	48.9
Model Soups	74.5	65.8	62.7	43.5	35.0	51.8
Linear Probing	72.0	63.2	72.4	43.8	47.6	56.7
LoRA	75.1	66.8	72.8	46.5	44.1	57.6
Random Mask	73.9	66.3	71.4	47.2	44.4	57.3
Mixout	74.2	66.3	66.2	44.7	39.7	54.2
GMixout	73.5	65.9	74.6	48.6	44.6	58.4
50 Shots						
Full-FT	74.7	65.3	57.0	39.7	28.6	47.7
MovingAvg	74.7	63.4	76.1	47.1	51.6	59.6
Model Soups	76.5	67.4	62.1	44.0	34.4	52.0
Linear Probing	73.3	63.8	71.2	43.3	46.5	56.2
LoRA	76.8	68.3	71.6	46.0	43.2	57.3
Random Mask	75.8	67.9	70.4	47.6	44.2	57.5
Mixout	76.0	67.2	65.1	44.8	39.3	54.1
GMixout	75.2	67.3	72.8	48.3	44.9	58.4
100 Shots						
Full-FT	76.9	66.7	58.2	41.3	28.9	48.8
MovingAvg	76.9	63.3	76.0	46.8	51.2	59.3
Model Soups	78.4	69.0	62.8	45.5	35.3	53.1
Linear Probing	74.7	64.6	70.3	43.2	45.6	55.9
LoRA	78.5	69.0	70.9	46.2	44.0	57.6
Random Mask	77.6	69.0	70.2	47.9	43.9	57.7
Mixout	77.6	68.5	65.3	45.4	38.0	54.3
GMixout	76.9	68.9	71.9	48.1	44.5	58.4
200 Shots						
Full-FT	78.7	68.6	57.9	41.2	29.5	49.3
MovingAvg	78.7	63.4	75.5	46.2	50.8	59.0
Model Soups	80.2	70.5	63.3	46.0	35.7	53.9
Linear Probing	75.8	65.6	70.0	43.4	45.9	56.2
LoRA	79.8	70.2	70.7	46.8	45.1	58.2
Random Mask	79.4	70.6	70.5	47.9	45.3	58.6
Mixout	79.1	69.9	64.4	45.0	36.9	54.0
GMixout	78.5	70.2	71.3	48.0	45.0	58.7