Discrepancy-Guided Parameter Suppression for Robust Fine-tuning

Chang Liu
Department of ECE
Northeastern University
Boston, MA 02115
liu.chang6@northesatern.edu

Jingyu Ma
Department of Statistics
University of Michigan
Ann Arbor, MI 48104
jingyum@umich.edu

Abstract

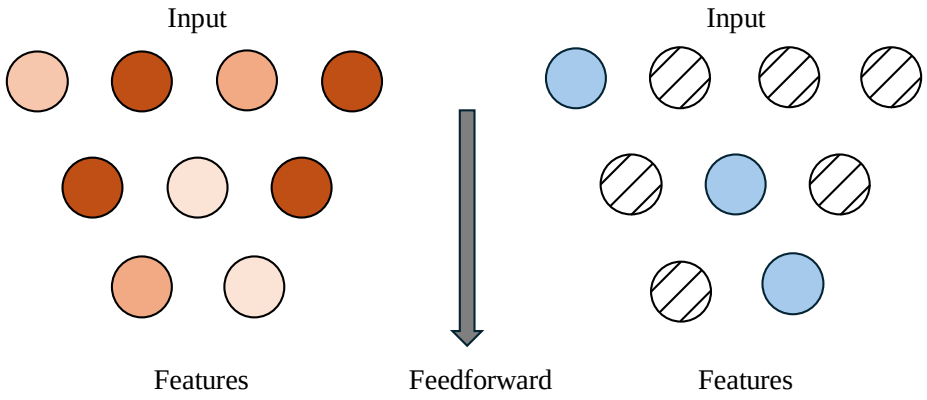
Foundation models (FMs) have demonstrated remarkable success in zero-shot learning and transferability across a broad range of unseen tasks. However, despite their robustness, fine-tuning these models on specific downstream tasks often leads to a trade-off: improvements in in-distribution (ID) performance typically come at the expense of out-of-distribution (OOD) generalization. To address this, recent research has focused on strategies that balance performance on the target dataset while retaining robustness on unseen data. In this paper, we propose a novel fine-tuning method that leverages parameter discrepancy between pre-trained and fine-tuned models to identify ID-specific parameters prone to overfitting. Our hypothesis is that parameters undergoing the most significant changes during fine-tuning are more likely to capture task-specific information. We introduce a Discrepancy-guided Parameter Suppression (DPS) mechanism to rank parameters with discrepancy score and selectively suppress those with the highest discrepancies to prevent overfitting. This approach encourages the model to learn task-invariant representations, improving OOD generalization. We evaluate our method on the DomainNet image classification benchmark, achieving a 1% improvement in OOD performance over the state-of-the-art method, without sacrificing ID performance. Additionally, we analyze the effects of parameter suppression percentages, selection granularity, and normalization strategies on discrepancy scores, providing comprehensive insights into robust fine-tuning.

1 Introduction

Recent advancements in foundation models (FMs), particularly vision-language models (VLMs)(15; 6; 9), have introduced significant improvements in zero-shot learning and transferability across a wide array of unseen tasks. However, despite their inherent robustness, the process of fine-tuning these models on a specific task often results in a trade-off: improvements in in-distribution (ID) performance on downstream tasks typically come at the expense of out-of-distribution (OOD) generalization on unseen data. Existing research emphasizes the need for robust fine-tuning methods(20; 5) that can balance the model's performance on the target dataset while retaining its robustness on OOD data.

To address this issue, researchers have explored various strategies along the directions of regularization in both representation space and parameter space. For the first direction, LP-FT(9) proposes fine-tuning the linear layers only, while FLYP(5) introduces pre-training-like fine-tuning to maintain generalization. Additionally, WISE-FT(20) suggests ensembling the zero-shot model and the fine-tuned model for better generalization. In the second direction, the focus is on regularizing the distance between the fine-tuned model and the zero-shot model. For example, L2-SP(22) constrains parameter

38th Workshop on Fine-Tuning in Machine Learning (NeurIPS 2024).



(a) Vanilla Fine-tuning (b) Discrepancy-Guided Parameter Suppression

Figure 1: (a) Vanilla fine-tune all the model parameters, (b) Discrepancy-Guided Parameter Suppression, which freezes ID-specific parameters while fine-tuning on others. The dark color on (a) indicates a high discrepancy score which is prone to be frozen.

deviation from the pre-trained weights using the L2 norm, while FTP (18) projects the parameter gradients to meet a constraint.

A key insight drawn from these works is the importance of identifying which parameters to fine-tune or preserve relative to the pre-trained weights to maintain OOD generalization. This raises important questions: What is the most effective metric to identify such parameters, and what is the optimal strategy to fine-tune them?

In this paper, we propose a simple yet effective metric to measure the ID-specific nature of parameters based on the discrepancy between the fine-tuned model and the pre-trained weights. Our central hypothesis is that parameters undergoing the most significant changes during fine-tuning are more likely to capture ID-specific information, which could lead to overfitting on the downstream domain. Based on this discrepancy metric, we rank the model's parameters from high to low at both the layer-wise and channel-wise granularity. To mitigate the model's tendency to learn spurious features (7) from the ID dataset, we propose to mute the parameters exhibiting the largest parameter discrepancies. By suppressing these parameters during fine-tuning, we encourage the model to learn task-invariant representations, improving generalization to OOD data.

Finally, we evaluate our method on DomainNet image classification benchmark, demonstrating its effectiveness in improving OOD robustness without sacrificing ID performance. Our experiments show a notable improvement in OOD performance, surpassing the state-of-the-art method (18) by 1%. Further, we explore the role of parameter suppression percentage, parameter selection granularity, and normalization techniques on discrepancy score, offering a detailed analysis of their contributions to robust fine-tuning. In summary, our contributions are:

- We propose a simple yet effective metric to measure the ID-specific nature of parameters based on the discrepancy between the fine-tuned model and the pre-trained weights. By muting parameters that exhibit the largest discrepancy during fine-tuning, we mitigate the learning of spurious features and encourage task-invariant representation learning, enhancing OOD performance.

- We conduct an in-depth analysis of the role of parameter suppression percentages, selection granularity (layer-wise and channel-wise), and normalization strategies, offering insights into the optimal setup for robust fine-tuning.

- Our method is evaluated on the DomainNet and iWildCam benchmarks, where we achieve state-of-the-art OOD performance, surpassing existing methods by 1.7%.

2 Related Works

2.1 Robust Fine-tuning

Fine-tuning large-scale models effectively for both in-distribution (ID) and out-of-distribution (OOD) performance remains a central challenge, as models often overfit to task-specific data and lose generalization capability. To address this, WiSE-FT (20) proposes combining the strengths of a zero-shot model and a fine-tuned model by ensembling their predictions. This approach seeks to leverage the robust, general-purpose knowledge of the zero-shot model, integrating it with task-specific adaptability from the fine-tuned model. The ensemble mechanism allows controlled adaptation, balancing OOD robustness with improvements on ID data.

Another key area of focus is constraining fine-tuning to reduce model drift from pre-trained representations. FLYP (5) encourages the model to maintain a similar optimization process to pre-training, aiming to preserve zero-shot functionality during adaptation to new tasks. In a similar vein, LP-FT (9) restricts updates to the classifier layer only, freezing all other layers to retain the original model's broad generalization abilities, thereby reducing overfitting risks.

Beyond structural constraints, regularization techniques are essential in controlling parameter updates. L2-SP (22) introduces L2 regularization to limit deviations from the pre-trained weights, effectively preventing parameter drift. Complementing this, FTP (18) and TPGM (17) employ projected gradients to constrain parameter updates according to predefined criteria. These techniques ensure fine-tuning remains focused on ID performance improvements without compromising OOD robustness. Our work builds on these regularization methods by introducing a novel discrepancy-based metric that identifies parameters vulnerable to overfitting. We then selectively control updates to these parameters, which helps further enhance OOD generalization and complements the current regularization efforts in both representation and parameter spaces.

2.2 Domain Generalization

Domain generalization (DG) seeks to train models capable of generalizing across multiple unseen domains. A foundational method, Empirical Risk Minimization (ERM)(19), minimizes average loss across source domains without explicit adjustments for domain shifts. More recent methods, including IRM(1), MixUp (21), DANN (3), CORAL (16), FVD (11), aim to achieve domain-invariant representations by adapting loss functions or feature spaces to enhance DG performance. Additionally, SWAD (2) demonstrated that models trained on flatter loss surfaces improve DG, achieving more consistent performance across domains. In applied settings, DG techniques extend beyond classification. For example, they have been applied to face recognition (12), where generalization across variations in pose and ethnicity is critical. In hyperparameter optimization, frameworks like HyperStar (10; 13) enable tuning across diverse tasks, addressing domain shifts in model configuration spaces.

Our work connects to DG by enhancing OOD robustness during fine-tuning, aligning with domain-invariant learning goals. By applying selective parameter regularization through a discrepancy metric, our approach bridges robust fine-tuning and DG, strengthening generalization in varied settings.

3 Method

3.1 Preliminaries

In this work, we aim to fine-tune a pre-trained model $f_{\mathbf{W}_0}$, which is parameterized by its weights $\mathbf{W}_0$ (e.g., CLIP (15)), to adapt it to a downstream task. After fine-tuning, the model is parameterized by the weights $\mathbf{W}$, representing the fine-tuned model.

We consider training data sampled from a distribution P_{ID}, where the objective is to learn the model $f_{\mathbf{W}} : \mathbb{R}^d \to Y$ that maps input data $x \in \mathbb{R}^d$ to output labels $y \in Y$. Formally, for a given loss function ℓ, we evaluate the performance of the fine-tuned model f by calculating both L_{sup}:

$$L_{\text{sup}}(f, \mathbf{W}, P_{\text{ID}}) = \mathbb{E}_{(x,y)\sim P_{\text{ID}}} \left[\ell(f(x; \mathbf{W}), y)\right] \tag{1}$$

The main challenge in this setting is to fine-tune the model in a way that not only achieves high accuracy on P_{ID}, but also ensures that the model generalizes well to unseen data from P_{OOD}, thereby enhancing its robustness to distributional shifts.

3.2 Identifying ID-specific parameters via Discrepancy-based Metric

As discussed in Section 1 , determining which parameters to fine-tune or freeze is critical for maintaining OOD performance. Motivated by prior work in OOD generalization (7), we observe that models tend to overfit to spurious features from the ID dataset, compromising their robustness to OOD data.

To address this, we propose a discrepancy-guided metric to identify parameters specific to the ID distribution. Specifically, we compute the absolute difference between the pre-trained weights $\mathbf{W}_0$ and the fine-tuned weights $\mathbf{W}_T$, where T denotes the training time. The key intuition is that parameters undergoing significant changes during fine-tuning are likely to capture ID-specific information. The parameter discrepancy is formally defined as:

$$\Delta \mathbf{W}^i = |\mathbf{W}_0^i - \mathbf{W}_T^i|, \tag{2}$$

where $\Delta \mathbf{W}^i$ represents the weight change in the i-th layer. Given that layers can have varying weight scales, normalization is necessary to ensure comparability. We normalize $\Delta \mathbf{W}^i$ by its L2 norm as follows:

$$\Delta \mathbf{W}_{norm}^i = \frac{\Delta \mathbf{W}^i}{\|\Delta \mathbf{W}^i\|_2}. \tag{3}$$

The ID-specific score for each channel is then computed as:

$$\mathbf{s}_{i,j} = \Delta \mathbf{w}_{norm}^{i,j}. \tag{4}$$

This score encapsulates the extent to which individual channels contribute to the downstream task. In the following section, we describe how this score is used to guide parameter updates during fine-tuning.

3.3 Discrepancy-guided Suppression on ID-specific parameters

After identifying ID-specific parameters through the discrepancy-based score in Eqn. 4, the next step is to determine the optimal strategy for fine-tuning. In standard fine-tuning, a model often relies heavily on these ID-specific parameters, leading to the risk of learning shortcuts based on the ID distribution, which degrades OOD performance. Our approach is designed to reduce the model's dependence on these ID-specific parameters and encourage learning task-relevant information from other parameters, thereby improving generalization.

To achieve this, we rank the model parameters by their discrepancy score from Eqn. 4 at the channel level, from highest to lowest. We then freeze the top-$K\%$ of parameters that exhibit high ID-specific scores while allowing fine-tuning on the remaining parameters. This freezing mechanism is incorporated into the gradient update for the i-th layer at iteration t as follows:

$$\mathbf{W}_{t+1}^i = \mathbf{W}_t^i - \eta \left(\mathbf{m}^i \odot \mathbf{g}_t^i \right), \tag{5}$$

where η is the learning rate, $\mathbf{m}^i$ is a binary mask for the i-th layer, and $\mathbf{g}_t^i$ represents the gradient. The binary mask $\mathbf{m}_{i,j}$ is defined as:

$$\mathbf{m}_{i,j} = \begin{cases} 0, & \text{if } \mathbf{s}_{i,j} \text{ belongs to the top-K\% of ranked scores,} \\ 1, & \text{otherwise.} \end{cases} \tag{6}$$

This ensures that the model does not receive gradient updates for the top-$K\%$ of ID-specific parameters, thereby reducing their influence during fine-tuning. In the next section, we discuss our experimental setup and results.

4 Experiments

In this section, we present experiments to evaluate the effectiveness of freezing channels in a pre-trained model during fine-tuning. We investigate three key aspects: the use of different discrepancy

Table 1: DomainNet Results using CLIP pre-trained ResNet50. Note that the results of baselines are adopted from FTP (18).

Methods	ID	OOD				Statistics
	Real	Sketch	Painting	Infograph	Clipart	OOD Avg.
Vanilla FT	80.93	31.81	41.02	20.29	43.59	34.18
Linear Prob.	52.56	20.05	24.92	19.18	21.15	21.33
L2-SP (22)	82.07	36.67	45.62	22.97	47.78	38.26
MARS-SP (4)	77.19	25.33	33.43	14.81	39.20	28.19
LP-FT (9)	80.82	34.85	44.03	22.23	46.13	36.81
TPGM (17)	83.64	38.78	43.11	28.70	48.01	39.65
FTP (18)	84.22	37.66	46.11	28.33	47.67	39.94
DPS	**84.53**	**39.87**	**47.41**	**30.09**	**49.18**	**41.63**

score strategies for ranking and freezing channels, the impact of layer-wise versus channel-wise freezing, and the effect of freezing varying percentages of channels.

4.1 Dataset

We follow prior works (5; 18) and consider following benchmarks for evaluating both in-distribution (ID) and out-of-distribution (OOD) performance.

- **DomainNet** (14) is one of the largest datasets specifically for domain adaptation tasks, consisting of six distinct domains: Clipart, Infograph, Painting, Quickdraw, Real, and Sketch. This dataset offers a comprehensive test bed for models tasked with adapting to varying visual styles across a broad range of categories.
- **iWildCam** (8) is a classification dataset consisting of 182 classes of animal images. The ID and OOD datasets differ based on the camera used and factors such as background, illumination, and other environmental conditions.

4.2 Implementation Details

Baselines. We consider five most recent and well performing methods namely, LP-FT (9), MARS-SP (4), WiSE-FT (20), FLPY (5), FTP (18), TPGM (17) and L2-SP (22), for robust fine-tuning.

Models and Optimization. We consider two models: CLIP ResNet-50 for DomainNet and CLIP ViT-B/16 for iWildCam. For CLIP ResNet-50, we used Stochastic Gradient Descent (SGD) as the optimizer. Training was performed with the following hyperparameters: a learning rate of 0.01, momentum of 0.9, and a batch size of 32. For CLIP ViT-B/16, we used AdamW as the optimizer with a cosine learning rate scheduler. Training was performed with the following hyperparameters: a learning rate of 1e-5, weight decay of 0.2, and a batch size of 64.

Hyperparameters. We consider freezing or masking ratio α on model parameters as the only hyperparameter in our method. We set α to 90% for both DomainNet and iWildCam. We will have a detailed explanation in section 4.4.

4.3 Results

DomainNet. On the DomainNet dataset, the proposed DPS (Discrepancy-based Parameter Suppression) method demonstrates significant improvements in both in-distribution (ID) and out-of-distribution (OOD) performance compared to state-of-the-art (SOTA) methods in Table 1. DPS achieves the highest ID accuracy at 84.53%, surpassing methods like FTP (84.22%) and L2-SP (82.07%). For OOD performance, DPS consistently outperforms other approaches, achieving notable improvements in challenging domains such as Clipart (49.18%) and Painting (47.41%), leading to an overall OOD average of 41.63%. This represents a marked improvement over FTP (39.94%) and

Table 2: iWildCam Results using CLIP pre-trained ViT B/16. Following common practice, we report macro F1-score. Note that the results of baselines are adopted from FLPY (5). †indicates our reproduced result.

Methods	iWildCam	
	ID	OOD
Zeroshot	8.7	11.0
Linear Prob.	44.5	31.1
Vanilla FT	48.1	35.0
L2-SP (22)	48.6	35.3
LP-FT (9)	49.7	34.7
WiSE-FT (20)	48.1	35.0
FTP (18) †	47.3	35.8
FLPY (5) †	**51.4**	35.2
DPS	50.5	**36.9**

TPGM (39.65%), highlighting DPS's ability to maintain generalization across diverse domains with varying visual characteristics.

iWildCam. On the iWildCam dataset, DPS also shows competitive performance. The method achieves an ID macro F1-score of 50.5%, which is comparable to other leading methods such as FLPY (51.4%) and LP-FT (49.7%). More importantly, DPS demonstrates improved OOD robustness, achieving the highest OOD macro F1-score at 36.9%, surpassing baselines like FTP (35.8%) and WiSE-FT (35.0%). This result emphasizes DPS's effectiveness in handling real-world domain shifts, such as varying environmental conditions and camera perspectives, making it a strong candidate for robust fine-tuning on natural image classification tasks.

4.4 Analysis and Ablation Study

This section presents ablation studies on four key aspects of our method on DomainNet benchmark: the choice of discrepancy score (criterion for freezing), the granularity of parameters to freeze (where to freeze), the percentage of parameters to freeze (how much to freeze), and the comparison with random masking. We discuss these factors in the following paragraphs.

Impact of Discrepancy Score. In this experiment, we compare two strategies for calculating the discrepancy score used to rank layers for freezing during fine-tuning. The strategies are: (1) the absolute discrepancy score from Eqn.2, and (2) the normalized discrepancy score from Eqn.4, as shown in Figure 3. Both aim to identify important layers by examining changes in their parameters after fine-tuning, but the key difference lies in how the importance of these changes is scaled. To control for other variables, we perform parameter freezing at the channel level, where the smallest unit for freezing is the output channel per convolutional layer, and freeze the top 30% of layers with the highest discrepancy scores. As shown in Figure 2, the results indicate that DPS with the normalized discrepancy score outperforms the raw absolute score, improving ID accuracy by 0.07% and OOD accuracy by 0.45%. This demonstrates that normalizing the discrepancy score helps standardize the importance of parameters across different parts of the model, resulting in a more effective ID-specific metric.

Block-wise vs. Channel-wise Freezing. To explore the effect of freezing different parts of the model, we compared parameter freezing at different granularities: (1) Layer-wise Freezing, where entire layers (i.e., all channels in a convolutional layer) are frozen, and (2) Channel-wise Freezing, where individual channels within a layer are frozen. In both cases, we froze the top 30% of parameters using the normalized discrepancy score. We qualitatively compare the parameters which are masked or frozen by two methods in Figure 4. We observe that channel-wise freezing offers a broader

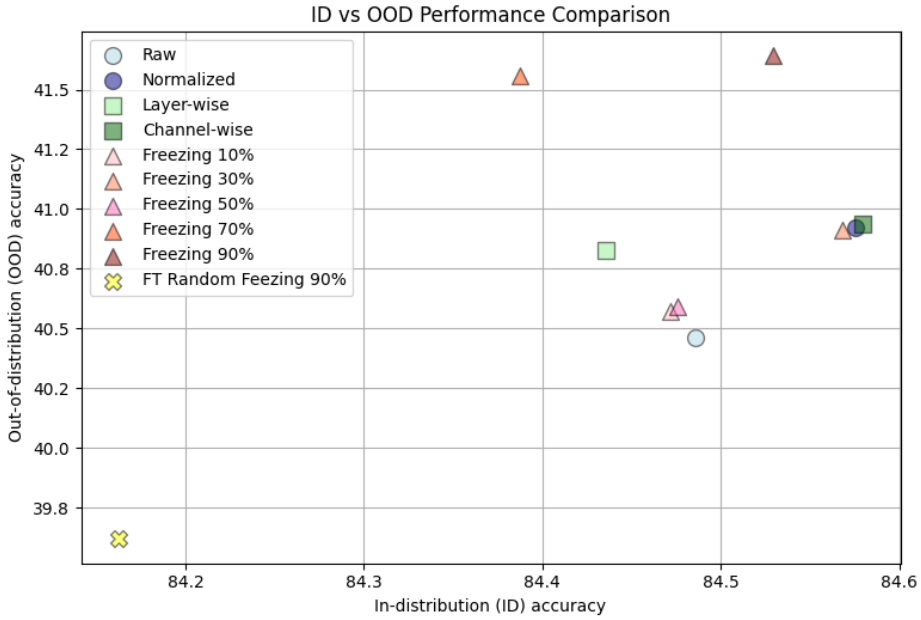
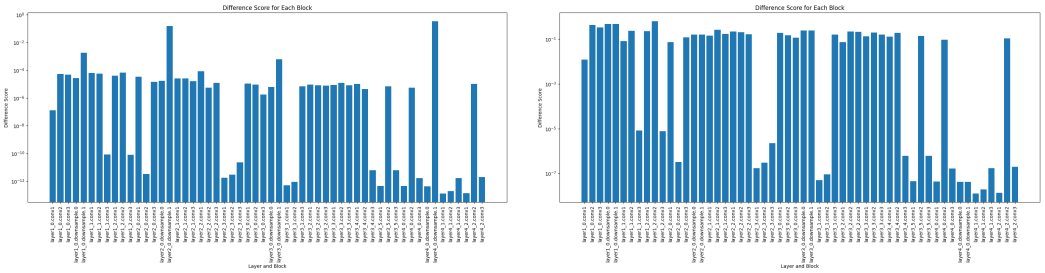


Figure 2: Comparing optimization on DomainNet Benchmark w.r.t Discrepancy Score, Freezing Granularity, Freezing percentage, and Random freezing.



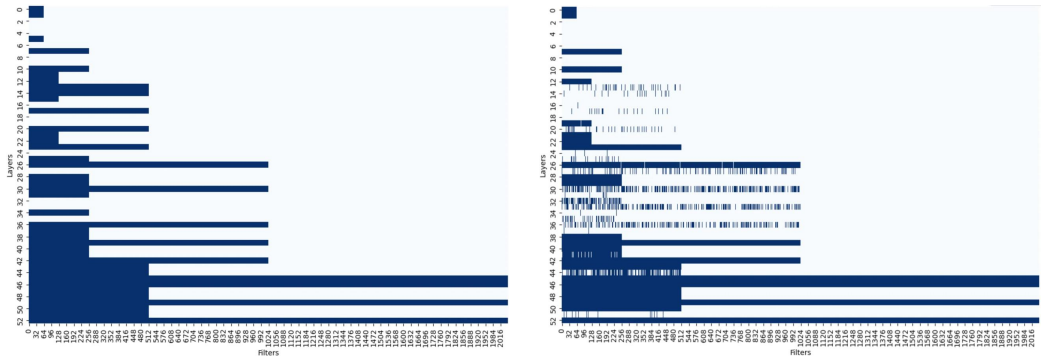
(a) Absolute Discrepancy Score (b) Normalized Discrepancy Score

Figure 3: Histogram of (a) Absolute and (b) Normalized, Discrepancy Score for each layer.

spectrum of control over parameters across layers than layer-wise freezing. The quantitative results in Figure 2 show that Channel-wise Freezing generally outperforms Layer-wise Freezing, offering greater flexibility and yielding performance improvements of 0.15% in ID accuracy and 0.09% in OOD accuracy. This suggests that freezing at a finer granularity provides better control over the learning process and enhances generalization.

Impact of Freezing Percentage. We also evaluated the impact of freezing different percentages of channels on model performance, with a focus on optimizing OOD performance. Specifically, we tested freezing 10%, 30%, 50%, 70%, and 90% of the parameters at the channel level using the normalized discrepancy score. The results in Figure 2 show that freezing 30% of channels provides a good balance between ID and OOD performance, achieving an ID accuracy of 84.57% and an OOD accuracy of 40.93%. As the freezing percentage increases to 70% and 90%, OOD performance continues to improve, with the best result at 90% freezing, reaching an OOD accuracy of 41.63%. Although this comes with a slight decrease in ID performance, it highlights that freezing larger percentages of parameters is more effective for enhancing OOD generalization, making 90% freezing the optimal choice when OOD robustness is prioritized.

Comparison with Random Masking. We further compare our discrepancy-based parameter suppression with a random masking baseline where 90% of the parameters are randomly frozen. As shown in Figure 2, random masking achieves an ID accuracy of 84.17% and an OOD accuracy of



(a) Layer-wise masking (30%)

(b) Channel-wise masking (30%)

Figure 4: Qualitative comparison between layer-wise and channel-wise freezing. Filters shown in dark blue represent the parameters to be fine-tuned, while filters with no color represent the parameters to be frozen.

39.61%, significantly lower than our approach, which achieves 84.52% ID accuracy and 41.63% OOD accuracy with 90% freezing. This comparison highlights the importance of selectively freezing parameters based on their discrepancy scores, as random masking fails to maintain strong performance, particularly in OOD generalization.

5 Conclusion and Limitation

In this paper, we proposed a novel fine-tuning approach that leverages parameter discrepancy between pre-trained and fine-tuned models to identify ID-specific parameters. By suppressing these parameters during fine-tuning, we successfully mitigate the model's tendency to overfit to the in-distribution (ID) data and encourage learning of task-invariant representations. Our approach, Discrepancy-based Parameter Suppression (DPS), outperforms existing state-of-the-art methods on the DomainNet and iWIldCam benchmarks, showing a 1.7% improvement in out-of-distribution (OOD) performance without compromising ID accuracy. Through detailed ablation studies, we demonstrated the effectiveness of varying suppression percentages, selection granularity, and normalization strategies, providing important insights into optimal fine-tuning configurations for robustness.

One limitation is that DPS requires training a vanilla fine-tuned (FT) model first, followed by the computation of parameter discrepancies between the pre-trained and fine-tuned models. This additional step increases the total training time. However, after computing the discrepancies, our DPS approach freezes 90% of the parameters, significantly reducing the number of parameters that need to be fine-tuned. This, in turn, minimizes the computational overhead in subsequent training stages, offsetting some of the initial time cost and making the overall fine-tuning process more efficient.

References

[1] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[2] J. Cha, S. Chun, K. Lee, H.-C. Cho, S. Park, Y. Lee, and S. Park. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34:22405–22418, 2021.

[3] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016.

[4] H. Gouk, T. M. Hospedales, and M. Pontil. Distance-based regularisation of deep networks for fine-tuning. *ICLR*, 2021.

[5] S. Goyal, A. Kumar, S. Garg, Z. Kolter, and A. Raghunathan. Finetune like you pretrain: Improved finetuning of zero-shot vision models. *arXiv preprint arXiv:2212.00638*, 2022.

[6] D. Hendrycks, K. Lee, and M. Mazeika. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning*, pages 2712–2721. PMLR, 2019.

[7] Z. Hu, Z. Zhao, X. Yi, T. Yao, L. Hong, Y. Sun, and E. Chi. Improving multi-task generalization via regularizing spurious correlation. *Advances in Neural Information Processing Systems*, 35:11450–11466, 2022.

[8] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, T. Lee, E. David, I. Stavness, W. Guo, B. A. Earnshaw, I. S. Haque, S. Beery, J. Leskovec, A. Kundaje, E. Pierson, S. Levine, C. Finn, and P. Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning (ICML)*, 2021.

[9] A. Kumar et al. Fine-tuning can distort pretrained features and underperform out-of-distribution. *ICLR*, 2022.

[10] C. Liu, G. Mittal, N. Karianakis, V. Fragoso, Y. Yu, Y. Fu, and M. Chen. Hyperstar: Task-aware hyperparameter recommendation for training and compression. *International Journal of Computer Vision*, 132(6):1913–1927, 2024.

[11] C. Liu, L. Wang, K. Li, and Y. Fu. Domain generalization via feature variation decorrelation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1683–1691, 2021.

[12] C. Liu, X. Yu, Y.-H. Tsai, M. Faraki, R. Moslemi, M. Chandraker, and Y. Fu. Learning to learn across diverse data biases in deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4072–4082, 2022.

[13] G. Mittal, C. Liu, N. Karianakis, V. Fragoso, M. Chen, and Y. Fu. Hyperstar: Task-aware hyperparameters for deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8736–8745, 2020.

[14] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019.

[15] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[16] B. Sun and K. Saenko. Deep CORAL: Correlation alignment for deep domain adaptation. *ECCV*, 2016.

[17] J. Tian, X. Dai, C.-Y. Ma, Z. He, Y.-C. Liu, and Z. Kira. Trainable projected gradient method for robust fine-tuning. *arXiv preprint arXiv:2303.10720*, 2023.

[18] J. Tian, Y.-C. Liu, J. S. Smith, and Z. Kira. Fast trainable projection for robust fine-tuning. *Advances in Neural Information Processing Systems*, 36, 2024.

[19] V. Vapnik. Statistical learning theory wiley. *New York*, 1998.

[20] M. Wortsman, G. Ilharco, J. W. Kim, M. Li, S. Kornblith, R. Roelofs, R. G. Lopes, H. Hajishirzi, A. Farhadi, H. Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7959–7971, 2022.

[21] M. Xu, J. Zhang, B. Ni, T. Li, C. Wang, Q. Tian, and W. Zhang. Adversarial domain adaptation with domain mixup. *arXiv*, 2019.

[22] L. Xuhong, Y. Grandvalet, and F. Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *International Conference on Machine Learning*, pages 2825–2834. PMLR, 2018.