UNDERSTANDING AND IMPROVING FEATURE LEARNING FOR OUT-OF-DISTRIBUTION GENERALIZATION

Yongqiang Chen[*1], **Wei Huang**[*2], **Kaiwen Zhou**[*1]
[1]The Chinese University of Hong Kong [2]RIKEN AIP
{yqchen,kwzhou,jcheng}@cse.cuhk.edu.hk wei.huang.vr@riken.jp
Yatao Bian[3], **Bo Han**[4], **James Cheng**[1]
[3]Tencent AI Lab [4]Hong Kong Baptist University
yatao.bian@gmail.com bhanml@comp.hkbu.edu.hk

ABSTRACT

A common explanation for the failure of out-of-distribution (OOD) generalization is that the model trained with empirical risk minimization (ERM) learns spurious features instead of the desired invariant features. However, several recent studies challenged this explanation and found that deep networks may have already learned sufficiently good features for OOD generalization. To understand these seemingly contradicting phenomena, we conduct a theoretical investigation and find that ERM learns *both* spurious features and invariant features. On the other hand, the quality of learned features during ERM pre-training significantly affects the final OOD performance, as OOD objectives rarely learn new features. Failing to capture all the useful features during pre-training will further limit the final OOD performance. To remedy the issue, we propose **F**eature **A**ugmented **T**raining (FAT), to enforce the model to learn all useful features by retaining the already learned features and augmenting new ones multiple rounds. In each round, the retention and augmentation are performed on different subsets of the training data that capture distinct features. Extensive experiments show that FAT learns richer features and consistently improves the OOD performance of various objectives.

1 INTRODUCTION

Understanding what features are learned by neural networks is crucial to understanding how they generalize to different data distributions (Rosenblatt, 1957; Shwartz-Ziv & Tishby, 2017; Shah et al., 2020; Allen-Zhu & Li, 2020). Deep networks trained with empirical risk minimization (ERM) learn highly predictive features that generalize surprisingly well to in-distribution data (Vapnik, 1991; Goodfellow et al., 2016). However, ERM also tends to learn *spurious* features such as image backgrounds (Beery et al., 2018; Geirhos et al., 2020) whose correlations with labels do not hold in the out-of-distribution (OOD) data, and suffers serious performance degeneration (Koh et al., 2021).

A common explanation for the OOD failures of deep networks is that ERM fails to learn the desired features that have *invariant* correlations with labels across different distributions (Beery et al., 2018). However, Rosenfeld et al. (2022); Kirichenko et al. (2022); Izmailov et al. (2022) found that ERM-trained models have *already learned sufficiently good features* that are able to generalize to OOD data. In addition, in the optimization of OOD objectives (Rojas-Carulla et al., 2018; Koyama & Yamaguchi, 2020; Parascandolo et al., 2021; Krueger et al., 2021; Pezeshki et al., 2021; Ahuja et al., 2021; Wald et al., 2021; Shi et al., 2022; Rame et al., 2021; Zhou et al., 2022; Chen et al., 2022a) that aim to capture the invariant features, there also exists an interesting phenomenon that the performance of OOD objectives largely relies on the pre-training with ERM (Zhang et al., 2022; Chen et al., 2022b). As shown in Fig. 1(b), the number of ERM pre-training epochs *has a large influence* on the OOD performance. These seemingly contradicting phenomena raise a challenging research question:

What features are learned by ERM and OOD objectives, respectively, and how do the learned features generalize to in-distribution and out-of-distribution data?

To answer the question, we conducted a theoretical investigation using a variation of data model of Allen-Zhu & Li (2020) that contains invariant and spurious features with different correlation

[*]Equal contributions.

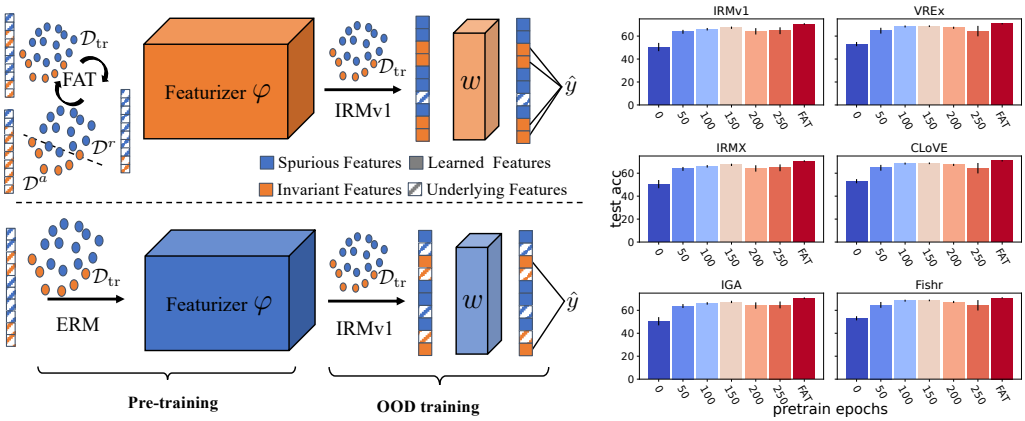


Figure 1: *(a)* An illustration of FAT (top row) compared to ERM (bottom row). FAT first identifies subsets containing distinct features by examining whether they are already learned by the model. Then FAT iteratively augments the new features while keep retaining the already learned features, and thus FAT learns richer features for OOD training. *(b)* OOD Performance vs. num. of pre-training epochs. The performance of various OOD objectives largely relies on the quality of ERM-learned features. When there exist underlying useful features poorly learned by ERM, the OOD performance will be limited, while FAT learns all useful features and thus improves OOD performance.

degrees with labels (Kamath et al., 2021). We studied the feature learning of a two-layer CNN, when trained with ERM and a representative OOD objective, IRMv1 (Arjovsky et al., 2019), respectively.

First, we found that ERM essentially learns *both* spurious features and invariant features (Theorem 3.1). The degrees of spurious and invariant feature learning are mostly controlled by their correlation strengths with labels. On the other hand, merely training with IRMv1 *cannot learn new* features (Theorem 3.2). Therefore, the *quality* of ERM-learned features affects the final OOD performance significantly. As the number of ERM pre-training epochs increases, the model also learns invariant features better and thus the final OOD performance will also improve (Fig. 1). However, when ERM training does not recover *all* useful features for OOD generalization, i.e., there exist some useful features that are poorly learned during ERM pre-training, then the model is less likely to learn these features during OOD training. In other words, OOD generalization requires the pre-training to learn all useful features that have non-trivial correlations with labels presented in the training data.

To remedy the issue, we propose **F**eature **A**ugmented **T**raining (FAT), an iterative data-centric training strategy to enforce the model to learn all useful features (Algorithm 1). As shown in Fig. 1(a), in each round k, FAT separates the training data into two subsets according to whether the underlying features are already learned (Retention set $\mathcal{D}_k^r$) or not (Augmentation set $\mathcal{D}_k^a$), which is identified by examining whether the model yields correct ($\mathcal{D}_k^r$) or incorrect ($\mathcal{D}_k^a$) predictions for samples from the subsets, respectively. Intuitively, different subsets contain distinct features that are discovered in different rounds. Then, FAT performs distributionally robust optimization (DRO) (Namkoong & Duchi, 2016; Zhang et al., 2022) on the grouped subsets $\{\mathcal{D}_k^a, \mathcal{D}_k^r\}$ to *augment* the feature learning by exploring new features. Meanwhile, FAT also needs to *retain* the already learned features by minimizing the empirical risk at the retention sets $\{\mathcal{D}_k^r\}$. FAT terminates when the model cannot learn any new predictive features from the augmentation subsets. Extensive experiments verified that FAT effectively learns richer features and improves OOD performance for various objectives (Sec. E).

2 Preliminaries and Problem Definition

We leave a detailed introduction of related work and backgrounds in Appendix A to Appendix B.
Notations. We use bold-faced letters for vectors and matrices otherwise scalars. We use $\|\cdot\|_2$ to denote the Euclidean norm of a vector or the spectral norm of a matrix.

We use the data model $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ that characterizes each data point $\mathbf{x}_i$ as invariant and spurious feature patches modified from (Allen-Zhu & Li, 2020; Kamath et al., 2021; Chen et al., 2022b).

Definition 2.1. $\mathcal{D} = \{\mathcal{D}_e\}_{e\in\mathcal{E}_{\text{all}}}$ contains multiple subsets $\mathcal{D}_e$ from various environments $e \in \mathcal{E}_{\text{all}}$, where each $\mathcal{D}_e = \{(\mathbf{x}_i^e, y_i^e)\}_{i=1}^{n_e}$ is composed of i.i.d. samples $(\mathbf{x}_i^e, y_i^e) \sim \mathbb{P}^e$. Each $(\mathbf{x}^e, y^e) \in \mathcal{D}_e$

with $\mathbf{x}^e \in \mathbb{R}^{2d}$ and $y^e \in \{-1, 1\}$ is generated as follows: (a) Sample $y^e \in \{-1, 1\}$ uniformly; (b) Each input $\mathbf{x}^e = [\mathbf{x}_1^e, \mathbf{x}_2^e]$ contains a feature patch $\mathbf{x}_1$ and a noise patch $\mathbf{x}_2$, sampled as:

$$\mathbf{x}_1 = y \cdot \text{Rad}(\alpha) \cdot \mathbf{v}_1 + y \cdot \text{Rad}(\beta) \cdot \mathbf{v}_2, \quad \mathbf{x}_2 = \boldsymbol{\xi},$$

where $\text{Rad}(\delta)$ takes value -1 $(+1)$ with probability δ $(1-\delta)$, $\mathbf{v}_1 = [1, 0, \ldots 0]^\top$ and $\mathbf{v}_2 = [0, 1, 0, \ldots 0]^\top$; (c) A noise vector $\boldsymbol{\xi}$ is sampled from $\mathcal{N}(\mathbf{0}, \sigma_p^2 \cdot (\mathbf{I}_d - \mathbf{v}_1\mathbf{v}_1^\top - \mathbf{v}_2\mathbf{v}_2^\top))$

Definition 2.1 is inspired by the structure of image data, where the inputs consist of different patches. Each environment is denoted as $\mathcal{E}_\alpha = \{(\alpha, \beta_e) : 0 < \beta_e < 1\}$, where $\mathbf{v}_1$ is the invariant feature as α is fixed for different e, and $\mathbf{v}_2$ is the spurious feature as β_e varies across different e.

CNN model. We consider training a two-layer convolutional neural network whose filters are applied to $\mathbf{x}_1$, $\mathbf{x}_2$, respectively, [1] and the second layer parameters of the network are fixed as $\frac{1}{m}$ and $-\frac{1}{m}$, respectively. m is the width of the hidden layer. Then the network becomes $f(\mathbf{W}, \mathbf{x}) = F_{+1}(\mathbf{W}_{+1}, \mathbf{x}) - F_{-1}(\mathbf{W}_{-1}, \mathbf{x})$, where $F_{+1}(\mathbf{W}_{+1}, \mathbf{x})$ and $F_{-1}(\mathbf{W}_{-1}, \mathbf{x})$ are defined as follows:

$$F_j(\mathbf{W}_j, \mathbf{x}) = \frac{1}{m} \sum_{r=1}^{m} \left[\sigma(\mathbf{w}_{j,r}^\top \mathbf{x}_1) + \sigma(\mathbf{w}_{j,r}^\top \mathbf{x}_2)\right], \tag{1}$$

where $\sigma(x)$ is the activation function. We focus on linear activation $\sigma(x) = x$ since it is sufficient to observe the desired feature learning behaviors of ERM and OOD objectives.[2]

ERM objective. We train the CNN model by minimizing the empirical cross-entropy loss function

$$L_S(\mathbf{W}) = \sum_{e \in \mathcal{E}_{\text{tr}}} \frac{1}{n_e} \sum_{i=1}^{n_e} \ell(y_i^e \cdot f(\mathbf{W}, \mathbf{x}_i^e)), \tag{2}$$

where $\ell(z) = \log(1 + \exp(-z))$ and $\mathcal{D}_{\text{tr}} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ is the training data set.

OOD objective. The goal of OOD generalization is, given the data from training environments $\mathcal{D}_{\text{tr}} = \{\mathcal{D}_e\}_{e \in \mathcal{E}_{\text{tr}}}$, with $n_e = |\mathcal{D}_e|$ to find a predictor $f : \mathcal{X} \to \mathcal{Y}$ that generalizes well to all (unseen) environments, or minimizes $\max_{e \in \mathcal{E}_{\text{all}}} L_e(f)$, where L_e is the empirical risk under environment e. The predictor $f = w \circ \varphi$ is usually composed of a featurizer $\varphi : \mathcal{X} \to \mathcal{Z}$ that learns to extract useful features, and a classifier $w : \mathcal{Z} \to \mathcal{Y}$ that makes predictions from the extracted features.

Since we are interested in the feature learning process when the OOD objective succeeds, without loss of generality, we analyze the IRMv1 objective and the data model where IRMv1 succeeds Arjovsky et al. (2019). Given the CNN (Eq. 1) and logistic loss (Eq. 2), IRMv1 can be written as

$$L_{\text{IRMv1}}(\mathbf{W}) = \sum_{e \in \mathcal{E}_{\text{tr}}} \frac{1}{n_e} \sum_{i=1}^{n_e} \ell\left(y_i^e \cdot f(\mathbf{W}, \mathbf{x}_i^e)\right) + \sum_{e \in \mathcal{E}_{\text{tr}}} \frac{\lambda}{n_e^2} \left(\sum_{i=1}^{n_e} \ell_i' \cdot y_i^e \cdot f(\mathbf{W}, \mathbf{x}_i^e)\right)^2, \tag{3}$$

where ${\ell'}_i^e = \ell'(y_i^e \cdot f(\mathbf{W}, \mathbf{x}_i^e)) = -\frac{\exp(-y_i^e \cdot f(\mathbf{W}, \mathbf{x}_i^e))}{1+\exp(-y_i^e \cdot f(\mathbf{W}, \mathbf{x}_i^e))}$. Due to the complexity of IRMv1, in the analysis below, we introduce C_{IRMv1}^e for the ease of expressions. Specifically,

$$C_{\text{IRMv1}}^e \triangleq \frac{1}{n_e} \sum_{i=1}^{n_e} \ell'\left(y_i^e \hat{y}_i^e\right) \cdot y_i^e \hat{y}_i^e,$$

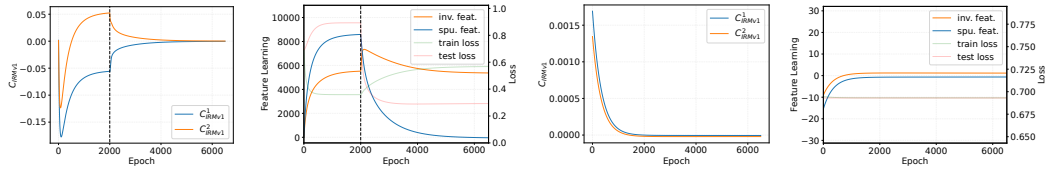
where $\hat{y}_i^e \triangleq f(\mathbf{W}, \mathbf{x}_i^e)$. The convergence of C_{IRMv1} indicates the convergence of IRMv1 penalty.

3 On Feature Learning in OOD Generalization

ERM Feature Learning. We first study the feature learning process of ERM objective. We consider a two training environments setup $\mathcal{E}_{tr} = \{(\alpha, \beta_1), (\alpha, \beta_2)\}$ where the signal of invariant feature is weaker than the average of spurious signals (i.e., $\alpha > \frac{\beta_1 + \beta_2}{2}$), which corresponds to Figure 2.

[1] When the environment e is not explicitly considered, we will omit it for clarity.

[2] It is also partially due to the complexity of IRMv1 dynamics, though to the best of our knowledge, we are the first to directly study the IRMv1 dynamics.



(a) C_{IRMv1}, w/ PT (b) FL, w/ PT (c) C_{IRMv1}, w/o PT (d) FL, w/o PT

Figure 2: The convergences of C_{IRMv1} and feature learning coefficients (FL) with or without ERM pre-training (PT). The training environments are $\mathcal{E}_{tr} = \{(0.25, 0.1), (0.25, 0.2)\}$. The black dashed line indicates the end of pre-training. Details are given in Appendix C.1.

Theorem 3.1. *(Informal) For $\rho > 0$, let $\underline{n} \triangleq \min_{e \in \mathcal{E}_{tr}} n_e$. Suppose that we run T iterations of GD for the ERM objective. With sufficiently large $\underline{n}$, assuming that (i) $\alpha, \beta_1, \beta_2 < \frac{1}{2}$, and (ii) $\alpha > \frac{\beta_1+\beta_2}{2}$, with properly chosen σ_0^2 and σ_p^2, there exists a constant η, such that with probability at least $1 - 2\rho$, both invariant and spurious features are converging and the increment of the spurious feature is larger than that of the invariant feature at any iteration $t \in \{0, \ldots, T-1\}$.*

The formal statement of this theorem and its proof are given in Appendix C.2. Theorem 3.1 states that ERM training learns both invariant feature and spurious feature at the same time, and if the average of spurious signals is stronger, the coefficient of spurious feature learning will dominate that of invariant feature learning in the whole training process, corresponding to Figure 2(b).

Theorem 3.1 explains the seemingly contradicting observations for the OOD failures. On the one hand, ERM fails since it learns the spurious features at a higher speed, when spurious correlations are stronger than invariant correlations. Although spurious feature learning effectively reduces the empirical risk, the learned features cannot generalize to OOD data where the correlations between spurious features and labels no longer hold (Beery et al., 2018). On the other hand, the invariant feature learning also happens, even when the spurious correlations are strong, so long as the invariant feature has a non-trivial correlation strength with the labels. Therefore, simply re-training a classifier based on a subset of unbiased data on top of the ERM-trained featurizer achieves impressive OOD performance (Rosenfeld et al., 2022; Kirichenko et al., 2022; Izmailov et al., 2022).

IRM Feature Learning. We then study IRMv1 training from scratch (w/o pre-training).

Theorem 3.2. *(Informal) Define $\mathbf{c}(t) \triangleq \left[C^1_{IRMv1}(\mathbf{W}, t), C^2_{IRMv1}(\mathbf{W}, t), \cdots, C^{|\mathcal{E}_{tr}|}_{IRMv1}(\mathbf{W}, t)\right]$, $\lambda_0 = \lambda_{\min}(\mathbf{H}^\infty)$, where $\mathbf{H}^\infty_{e,e'} \triangleq \frac{1}{n_e n_{e'}} \sum_{i=1}^{n_e} \mathbf{x}^{e\top}_{1,i} \sum_{i'=1}^{n_{e'}} \mathbf{x}^{e'}_{1,i}$. Suppose that dimension $d = \Omega(\log(m/\delta))$, network width $m = \Omega(1/\delta)$, regularization factor $\lambda \geq 1/\sigma_0$, noise variance $\sigma_p = O(d^{-2})$, weight initial scale $\sigma_0 = O(\min\{\frac{\lambda_0^2 m^2}{\log(1/\epsilon)}, \frac{\lambda_0 m}{\sqrt{d}\log(1/\epsilon)}\})$, then with probability at least $1 - \delta$, after training time $T = \Omega\left(\frac{\log(1/\epsilon)}{\eta\lambda\lambda_0}\right)$, we have $\|\mathbf{c}(T)\|_2 \leq \epsilon, \Lambda^t_{j,r} = o_d(1), \Gamma^t_{j,r} = o_d(1)$.*

The formal statement of this theorem and its proof are given in Appendix C.3. Intuitively, Theorem 3.2 implies that, when a heavy regularization of IRMv1 is applied, the model will not learn any features, corresponding to Figure 2(d). Then, what would happen when given a properly pre-trained network?

Proposition 3.3. *(Informal) Given $\Lambda^t_{j,r} = \Lambda^{t-1}_{j,r}$ and $\Gamma^t_{j,r} = \Gamma^{t-1}_{j,r}$ at the end of ERM pre-train t_1 and $\mathcal{E}_{tr} = \{(0.25, 0.1), (0.25, 0.2)\}$. Then, in the limit of $n \to \infty$, we have $\sum_e C^e_{IRMv1}(t_1) = 0, \Lambda^t_{j,r} > \Lambda^{t+1}_{j,r}, \Gamma^t_{j,r} < \Gamma^{t+1}_{j,r}$.*

The proof is given in Appendix **??**, which takes converged feature learning terms from Theorem 3.1 as the inputs. Proposition 3.3 demonstrates that with sufficient ERM pre-train, IRMv1 promotes invariant feature learning while suppressing spurious features learning, which is verified in Figure 2(b) and 2(a). Therefore, IRMv1 tends to exacerbate the learned invariant features from ERM pre-training and suppress the spurious features. Thus, when given the initialization with better learned invariant features from ERM pre-training, i.e., longer pre-training epochs, IRMv1 exacerbates the invariant features faster and better, Proposition 3.3 explains why the final OOD generalization performance highly depends on the number of ERM pre-training epochs (Zhang et al., 2022; Chen et al., 2022b).

Limitations of ERM Feature Learning. Previous results show that the underlying invariant features will be learned during ERM pre-training and discovered during OOD training, despite that IRMv1 will not learn any features at the beginning. The remaining curious question is, given a poorly learned invariant feature, will IRMv1 still identify and exacerbate it? The answer is negative.

Table 1: OOD performance on COLOREDMNIST datasets initialized with different representations.

	COLOREDMNIST-025				COLOREDMNIST-01			
	ERM-NF	ERM	BONSAI	FAT	ERM-NF	ERM	BONSAI	FAT
ERM	17.14 (±0.73)	12.40 (±0.32)	11.21 (±0.49)	**17.27** (±2.55)	73.06 (±0.71)	73.75 (±0.49)	70.95 (±0.93)	**76.05** (±1.45)
IRMV1	67.29 (±0.99)	59.81 (±4.46)	70.28 (±0.72)	**70.57** (±0.68)	76.89 (±3.25)	73.84 (±0.56)	76.71 (±4.10)	**82.33** (±1.77)
V-REX	68.62 (±0.73)	65.96 (±1.29)	70.31 (±0.66)	**70.82** (±0.59)	83.52 (±2.52)	81.20 (±3.27)	82.61 (±1.76)	**84.70** (±0.69)
IRMX	67.00 (±1.95)	64.05 (±0.88)	70.46 (±0.42)	**70.78** (±0.61)	81.61 (±1.98)	75.97 (±0.88)	80.28 (±1.62)	**84.34** (±0.97)
IB-IRM	56.09 (±2.04)	59.81 (±4.46)	70.28 (±0.72)	**70.57** (±0.68)	75.81 (±0.63)	73.84 (±0.56)	76.71 (±4.10)	**82.33** (±1.77)
CLOVE	58.67 (±7.69)	65.78 (±0.00)	65.57 (±3.02)	**65.78** (±2.68)	75.66 (±10.6)	74.73 (±0.36)	72.73 (±1.18)	**75.12** (±1.08)
IGA	51.22 (±3.67)	62.43 (±3.06)	**70.17** (±0.89)	67.11 (±3.40)	74.20 (±2.45)	73.74 (±0.48)	74.72 (±3.60)	**83.46** (±2.17)
FISHR	69.38 (±0.39)	67.74 (±0.90)	68.75 (±1.10)	**70.56** (±0.97)	77.29 (±1.61)	82.23 (±1.35)	84.19 (±0.66)	**84.26** (±0.93)
ORACLE	71.97 (±0.34)				86.55 (±0.27)			

Corollary 3.4. *Given $\Lambda^t_{j,r} = o(1)$ and $\Gamma^t_{j,r} = \Theta(1)$, at the end of ERM pre-train t_1 and $\mathcal{E}_{tr} = \{(0.25, 0.1), (0.25, 0.2)\}$. Then, in the limit of $n \to \infty$, we have $\Lambda^{t+1}_{j,r} < \Gamma^t_{j,r}$.*

As a direct extension of Proposition 3.3, Corollary 3.4 shows that IRMv1 OOD generalization requires sufficiently well-learned features. It is also consistent with the experimental results in Fig. 1, where all the OOD objectives achieve only performance comparable to random guesses.

Improving Feature Learning for OOD Generalization. The results in Sec. 3 imply that the model is expected to learn all potentially useful features during the pre-training in order to achieve the optimal OOD performance. To this end, we propose **F**eature **A**ugmented **T**raining (FAT), that adopts an iterative data-centric strategy to enforce the model to learn all useful features directly. We give a brief introduction below while leaving the detailed description of FAT in Appendix D.

Intuitively, the potentially useful features presented in the training data are features that have non-trivial correlations with labels. Moreover, the invariance principle assumes that the training data comes from different environments (Arjovsky et al., 2019), which implies that each set of features can only dominate the correlations with labels in a *subset* of data. Therefore, it is possible to differentiate the distinct sets of useful features entangled in the training data into different subsets.

The intuition naturally motivates an iterative rich feature learning algorithm, i.e., FAT, that differentiates the subsets and explores new features by multiple rounds. In round k, FAT first identifies the subset that contains the already learned features by collecting the data points where f yields the correct prediction, denoted as G^r_k, and the subset that contains the other samples as G^a_k. Given a grouped datasets $G = \{G^r, G^a\}$ with $2k-1$ groups, where $G^a = \{\mathcal{D}^a_i\}_{i=0}^{k-1}$ and $G^r = \{\mathcal{D}^r_i\}_{i=1}^{k-1}$ (notice that $\mathcal{D}^r_0$ is the empty set), FAT performs distributionally robust optimization (DRO) (Namkoong & Duchi, 2016; Zhang et al., 2022) on G^a to explore new features that have not been learned in previous rounds. Meanwhile, FAT also needs to *retain* the already learned features by ERM at G^r, as

$$\ell_{\text{FAT}} = \max_{\mathcal{D}^a_i \in G^a} \ell_{\mathcal{D}^a_i}(w_k \circ \varphi) + \lambda \cdot \sum\nolimits_{\mathcal{D}^r_i \in G^r} \ell_{\mathcal{D}^r_i}(w_i \circ \varphi), \tag{4}$$

where $\ell_{\mathcal{D}_i}(w \circ \varphi)$ is the empirical risk of $w \circ \varphi$ at $\mathcal{D}_i$, and $\{w_i | 1 \leq i \leq k-1\}$ are the historical classifiers.

Empirical verification. We verified the effectiveness of FAT by applying the learned features to various OOD objectives, based on COLOREDMNIST and WILDS datasets. The featurizer is frozen to prevent feature distortion (Kumar et al., 2022). More results and details are given in Appendix E and Appendix F, respectively. Table 1 presents the results in COLOREDMNIST datasets. We used both the original COLOREDMNIST with $\mathcal{E}_{\text{tr}} = \{(0.25, 0.1), (0.25, 0.2)\}$ (denoted as COLOREDMNIST-025), where spurious correlations are stronger, and the modified COLOREDMNIST (denoted as COLOREDMNIST-01) with $\mathcal{E}_{\text{tr}} = \{(0.1, 0.2), (0.1, 0.25)\}$, where invariant correlations are stronger. We also include the common practice with non-frozen ERM learned features (denoted as ERM-NF).

It can be found that FAT significantly improves the OOD performance of all OOD objectives for all COLOREDMNIST datasets over ERM. In contrast, although Bonsai boosts the OOD performance for COLOREDMNIST-025, it also leads to suboptimal performances when invariant correlations are stronger, which could be attributed to the uncontrolled feature learning with multiple initializations.

4 CONCLUSIONS

In this paper, we found that ERM learns both invariant and spurious features when OOD objectives rarely learn new features. Thus, the features learned in the ERM pre-training can greatly influence the final OOD performance. Having learned the limitations of ERM pre-training, we proposed FAT to learn all potentially useful features. Extensive experimental results verified the superiority of FAT.

ACKNOWLEDGEMENTS

We thank the reviewers for their valuable comments. This work was supported by CUHK direct grant 4055146. BH were supported by the NSFC Young Scientists Fund No. 62006202, Guangdong Basic and Applied Basic Research Foundation No. 2022A1515011652, RGC Early Career Scheme No. 22200720, and Tencent AI Lab Rhino-Bird Gift Fund.

REFERENCES

Kartik Ahuja, Ethan Caballero, Dinghuai Zhang, Jean-Christophe Gagnon-Audet, Yoshua Bengio, Ioannis Mitliagkas, and Irina Rish. Invariance principle meets information bottleneck for out-of-distribution generalization. In *Advances in Neural Information Processing Systems*, 2021.

Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020.

Martín 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. In *Advances in Neural Information Processing Systems*, 2022.

Péter Bándi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, Quanzheng Li, Farhad Ghazvinian Zanjani, Svitlana Zinger, Keisuke Fukuta, Daisuke Komura, Vlado Ovtcharov, Shenghua Cheng, Shaoqun Zeng, Jeppe Thagaard, Anders B. Dahl, Huangjing Lin, Hao Chen, Ludwig Jacobsson, Martin Hedlund, Melih Çetin, Eren Halici, Hunter Jackson, Richard Chen, Fabian Both, Jörg Franke, Heidi Küsters-Vandevelde, Willem Vreuls, Peter Bult, Bram van Ginneken, Jeroen van der Laak, and Geert Litjens. From detection of individual metastases to classification of lymph node status at the patient level: The CAMELYON17 challenge. *IEEE Trans. Medical Imaging*, 38(2):550–560, 2019.

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Computer Vision European Conference, Part XVI*, pp. 472–489, 2018.

Sara Beery, Elijah Cole, and Arvi Gjoka. The iwildcam 2020 competition dataset. *arXiv preprint arXiv:2004.10340*, 2020.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion of The 2019 World Wide Web Conference*, pp. 491–500, 2019.

Alon Brutzkus, Amir Globerson, Eran Malach, and Shai Shalev-Shwartz. SGD learns over-parameterized networks that provably generalize on linearly separable data. In *International Conference on Learning Representations*, 2018.

Yuan Cao, Zixiang Chen, Mikhail Belkin, and Quanquan Gu. Benign overfitting in two-layer convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2022a.

Yuan Cao, Zixiang Chen, Misha Belkin, and Quanquan Gu. Benign overfitting in two-layer convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2022b.

Yongqiang Chen, Yonggang Zhang, Yatao Bian, Han Yang, Kaili Ma, Binghui Xie, Tongliang Liu, Bo Han, and James Cheng. Learning causally invariant representations for out-of-distribution generalization on graphs. In *Advances in Neural Information Processing Systems*, 2022a.

Yongqiang Chen, Kaiwen Zhou, Yatao Bian, Binghui Xie, Kaili Ma, Yonggang Zhang, Han Yang, Bo Han, and James Cheng. Pareto invariant risk minimization. *arXiv preprint arXiv:2206.07766*, 2022b.

Gordon A. Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6172–6180, 2018.

Spencer Frei, Yuan Cao, and Quanquan Gu. Provable generalization of sgd-trained neural networks of any width in the presence of adversarial label noise. In *International Conference on Machine Learning*, pp. 3427–3438, 2021.

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

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

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

Arushi Gupta, Nikunj Saunshi, Dingli Yu, Kaifeng Lyu, and Sanjeev Arora. New definitions and evaluations for saliency methods: Staying intrinsic and sound, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2261–2269, 2017.

Pavel Izmailov, Polina Kirichenko, Nate Gruver, and Andrew Gordon Wilson. On feature learning in the presence of spurious correlations. In *Advances in Neural Information Processing Systems*, 2022.

Pritish Kamath, Akilesh Tangella, Danica Sutherland, and Nathan Srebro. Does invariant risk minimization capture invariance? In *International Conference on Artificial Intelligence and Statistics*, pp. 4069–4077, 2021.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*, 2022.

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 Earnshaw, Imran Haque, Sara M. Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning,*, pp. 5637–5664, 2021.

Masanori Koyama and Shoichiro Yamaguchi. Out-of-distribution generalization with maximal invariant predictor. *arXiv preprint arXiv:2008.01883*, 2020.

David Krueger, Ethan Caballero, Jörn-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Rémi Le Priol, and Aaron C. Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pp. 5815–5826, 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.

Hongseok Namkoong and John C. Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *Advances in Neural Information Processing Systems*, pp. 2208–2216, 2016.

Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing*, pp. 188–197, 2019.

Giambattista Parascandolo, Alexander Neitz, Antonio Orvieto, Luigi Gresele, and Bernhard Schölkopf. Learning explanations that are hard to vary. In *International Conference on Learning Representations*, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pp. 8024–8035, 2019.

Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron C. Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. In *Advances in Neural Information Processing Systems*, pp. 1256–1272, 2021.

Alexandre Rame, Corentin Dancette, and Matthieu Cord. Fishr: Invariant gradient variances for out-of-distribution generalization. *arXiv preprint arXiv:2109.02934*, 2021.

Alexandre Rame, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, patrick gallinari, and Matthieu Cord. Diverse weight averaging for out-of-distribution generalization. In *Advances in Neural Information Processing Systems*, 2022.

Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. Invariant models for causal transfer learning. *Journal of Machine Learning Research*, 19(36):1–34, 2018.

F. Rosenblatt. The perceptron - a perceiving and recognizing automaton. Technical Report 85-460-1, Cornell Aeronautical Laboratory, 1957.

Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. Domain-adjusted regression or: Erm may already learn features sufficient for out-of-distribution generalization. *arXiv preprint arXiv:2202.06856*, 2022.

Shiori Sagawa*, Pang Wei Koh*, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020.

Wojciech Samek, Gregoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Muller. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer Publishing Company, Incorporated, 1st edition, 2019. ISBN 3030289532.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In *Advances in Neural Information Processing Systems*, pp. 9573–9585, 2020.

Ruoqi Shen, Sebastien Bubeck, and Suriya Gunasekar. Data augmentation as feature manipulation. In *International Conference on Machine Learning*, pp. 19773–19808, 2022.

Yuge Shi, Jeffrey Seely, Philip Torr, Siddharth N, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. In *International Conference on Learning Representations*, 2022.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

James Taylor, Berton Earnshaw, Ben Mabey, Mason Victors, and Jason Yosinski. Rxrx1: An image set for cellular morphological variation across many experimental batches. In *International Conference on Learning Representations*, 2019.

Vladimir Vapnik. Principles of risk minimization for learning theory. In *Advances in Neural Information Processing Systems*, pp. 831–838, 1991.

Yoav Wald, Amir Feder, Daniel Greenfeld, and Uri Shalit. On calibration and out-of-domain generalization. In *Advances in Neural Information Processing Systems*, pp. 2215–2227, 2021.

Zixin Wen and Yuanzhi Li. Toward understanding the feature learning process of self-supervised contrastive learning. In *International Conference on Machine Learning*, pp. 11112–11122, 2021.

Haotian Ye, James Zou, and Linjun Zhang. Freeze then train: Towards provable representation learning under spurious correlations and feature noise. *arXiv preprint arXiv:2210.11075*, 2022.

Jianyu Zhang and Leon Bottou. Learning useful representations for shifting tasks and distributions. 2022.

Jianyu Zhang, David Lopez-Paz, and Leon Bottou. Rich feature construction for the optimization-generalization dilemma. In *International Conference on Machine Learning*, pp. 26397–26411, 2022.

Xiao Zhou, Yong Lin, Renjie Pi, Weizhong Zhang, Renzhe Xu, Peng Cui, and Tong Zhang. Model agnostic sample reweighting for out-of-distribution learning. In *International Conference on Machine Learning*, pp. 27203–27221, 2022.

Difan Zou, Yuan Cao, Yuanzhi Li, and Quanquan Gu. Understanding the generalization of adam in learning neural networks with proper regularization. *arXiv preprint arXiv:2108.11371*, 2021.

A Related Work

On Feature Learning and Generalization. Understanding what features are learned by deep networks is crucial to understanding their generalization (Rosenblatt, 1957; Shwartz-Ziv & Tishby, 2017; Brutzkus et al., 2018; Frei et al., 2021; Allen-Zhu & Li, 2020; Cao et al., 2022b). Beyond the empirical probing (Samek et al., 2019; Gupta et al., 2022), Allen-Zhu & Li (2020) proposed a new theoretical framework that characterizes the feature learning process of deep networks, which has been widely adopted to analyze behaviors of deep networks (Wen & Li, 2021; Zou et al., 2021; Cao et al., 2022b) However, how the learned features from in-distribution data can generalize to OOD data remains elusive. The only exception is Shen et al. (2022), which focuses on how data augmentation helps promote good but hard to learn features and improve OOD generalization. In contrast, we study the direct influence of ERM and OOD objectives to feature learning and aim to provide a theoretical explanation to the phenomenon that ERM may have already learned good features (Rosenfeld et al., 2022; Kirichenko et al., 2022; Izmailov et al., 2022).

Rich Feature Learning. Recently many OOD objectives have been proposed to regularize ERM such that the model can focus on learning invariant features (Arjovsky et al., 2019; Krueger et al., 2021; Pezeshki et al., 2021; Wald et al., 2021; Rame et al., 2021). However, due to the intrinsic conflicts of ERM and OOD objectives, it often requires exhaustive hyperparameter tuning of ERM pre-training epochs and regularization weights (Zhang et al., 2022; Chen et al., 2022b). Especially, the final OOD performance has a large dependence on the number of pre-training epochs. To remedy the issue, Zhang et al. (2022) proposed Bonsai to construct rich feature representations with plentiful potentially useful features as network initialization. Although both Bonsai and FAT perform DRO on grouped subsets, Bonsai rely on multiple initializations of the whole network to capture diverse features from the subsets, and complicated ensembling of the features, which requires much more training epochs for the convergence. In contrast, FAT relieves the requirements by performing direct augmentation-retention on the grouped subsets, and thus obtains better performance. More crucially, although Bonsai and other rich feature learning algorithms such as weight averaging (Rame et al., 2022; Arpit et al., 2022; Zhang & Bottou, 2022) have gained impressive successes in mitigating the dilemma, explanations about the reliance on ERM pre-training and why rich feature learning mitigates the dilemma remain elusive. Our work provides novel theoretical explanations for the success of rich feature learning algorithms for OOD generalization.

B Preliminaries and Problem Definition

Notations. We use bold-faced letters for vectors and matrices otherwise representing scalar. We use $\|\cdot\|_2$ to denote the Euclidean norm of a vector or the spectral norm of a matrix, while denoting $\|\cdot\|_F$ as the Frobenius norm of a matrix. For a neural network, we denote $\sigma(x)$ as the activation function. Let $\mathbf{I}_d$ be the identity matrix with dimension of $\mathbb{R}^{d\times d}$. We denote $[n] = \{1, 2, \ldots, n\}$.

Our data model $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ is adapted from (Allen-Zhu & Li, 2020) and further characterizes each data point $\mathbf{x}_i$ as invariant and spurious feature patches from the two-bit model (Kamath et al., 2021; Chen et al., 2022b).

Definition B.1. $\mathcal{D} = \{\mathcal{D}_e\}_{e\in\mathcal{E}_{\text{all}}}$ is composed of multiple subsets $\mathcal{D}_e$ from different environments $e \in \mathcal{E}_{\text{all}}$, where each $\mathcal{D}_e = \{(\mathbf{x}_i^e, y_i^e)\}_{i=1}^{n_e}$ is composed of i.i.d. samples $(\mathbf{x}_i^e, y_i^e) \sim \mathbb{P}^e$. Each data $(\mathbf{x}^e, y^e) \in \mathcal{D}_e$ with $\mathbf{x}^e \in \mathbb{R}^{2d}$ and $y^e \in \{-1, 1\}$ is generated as follows:

(a) Sample $y^e \in \{-1, 1\}$ uniformly;

(b) Given y^e, each input $\mathbf{x}^e = [\mathbf{x}_1^e, \mathbf{x}_2^e]$ contains a feature patch $\mathbf{x}_1$ and a noise patch $\mathbf{x}_2$, that are sampled as:

$$\mathbf{x}_1 = y \cdot \text{Rad}(\alpha) \cdot \mathbf{v}_1 + y \cdot \text{Rad}(\beta) \cdot \mathbf{v}_2 \quad \mathbf{x}_2 = \boldsymbol{\xi}$$

where $\text{Rad}(\delta)$ is a random variable taking value -1 with probability δ and $+1$ with probability δ, $\mathbf{v}_1 = [1, 0, \ldots 0]^\top$ and $\mathbf{v}_2 = [0, 1, 0, \ldots 0]^\top$.

(c) A noise vector $\boldsymbol{\xi}$ is generated from the Gaussian distribution $\mathcal{N}(\mathbf{0}, \sigma_p^2 \cdot (\mathbf{I}_d - \mathbf{v}_1\mathbf{v}_1^\top - \mathbf{v}_2\mathbf{v}_2^\top))$

Definition B.1 is inspired by the structure of image data, where the inputs consist of different patches, some of the patches consist of features that are related to the class label of the image, and the others are noises that are irrelevant to the label. In particular, $\mathbf{v}_1$ and $\mathbf{v}_2$ are feature vectors, in which $\mathbf{v}_1$ simulates the underlying invariant feature while $\mathbf{v}_2$ simulates the spurious feature. Although our data model focuses on two feature vectors, the discussion and results can be further generalized to multiple invariant and spurious features with fine-grained characteristics (Shen et al., 2022). Besides, we assume that the noise patch is generated from the Gaussian distribution $\mathcal{N}(\mathbf{0}, \sigma_p^2 \cdot (\mathbf{I}_d - \mathbf{v}_1\mathbf{v}_1^\top - \mathbf{v}_2\mathbf{v}_2^\top))$ to ensure that the noise vector is orthogonal to the signal vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ for simplicity. Each environment is denoted as $\mathcal{E}_\alpha = \{(\alpha, \beta_e) : 0 < \beta_e < 1\}$, where $\mathbf{v}_1$ is the invariant feature as α is fixed for different environment e, and $\mathbf{v}_2$ is the spurious feature as β_e varies across different e.

CNN model. We consider training a two-layer convolutional neural network whose filters are applied to $\mathbf{x}_1, \mathbf{x}_2$, respectively, [3] and the second layer parameters of the network are fixed as $\frac{1}{m}$ and $-\frac{1}{m}$, respectively, with m being the width of the hidden layer. Then the network can be written as $f(\mathbf{W}, \mathbf{x}) = F_{+1}(\mathbf{W}_{+1}, \mathbf{x}) - F_{-1}(\mathbf{W}_{-1}, \mathbf{x})$, where $F_{+1}(\mathbf{W}_{+1}, \mathbf{x})$ and $F_{-1}(\mathbf{W}_{-1}, \mathbf{x})$ are defined as follows:

$$F_j(\mathbf{W}_j, \mathbf{x}) = \frac{1}{m} \sum_{r=1}^{m} \left[\sigma(\mathbf{w}_{j,r}^\top \mathbf{x}_1) + \sigma(\mathbf{w}_{j,r}^\top \mathbf{x}_2) \right], \tag{5}$$

where $\sigma(x)$ is the activation function. We assume that all network weights are initialized as $\mathcal{N}(0, \sigma_0^2)$. In this work, we focus on linear activation $\sigma(x) = x$ since it is sufficient to observe the desired feature learning behaviors of ERM and OOD objectives.[4] Nevertheless, our framework can also be extended to non-linear activation functions, such as q-ReLU (Zou et al., 2021; Cao et al., 2022a).

ERM objective. We train the above CNN model by minimizing the empirical cross-entropy loss function

$$L_S(\mathbf{W}) = \sum_{e \in \mathcal{E}_{\text{tr}}} \frac{1}{n_e} \sum_{i=1}^{n_e} \ell(y_i^e \cdot f(\mathbf{W}, \mathbf{x}_i^e)), \tag{6}$$

where $\ell(z) = \log(1 + \exp(-z))$ and $\mathcal{D}_{\text{tr}} = \{\mathcal{D}_e\}_{e \in \mathcal{E}_{\text{tr}}} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ is the training data set with $\sum_{e \in \mathcal{E}_{\text{tr}}} n_e = n$.

OOD objective. The goal of OOD generalization is, given the data from training environments $\{\mathcal{D}_e\}_{e \in \mathcal{E}_{\text{tr}}}$, to find a predictor $f : \mathcal{X} \to \mathcal{Y}$ that generalizes well to all (unseen) environments, or minimizes $\max_{e \in \mathcal{E}_{\text{all}}} L_e(f)$, where L_e is the empirical risk under environment e. The predictor $f = w \circ \varphi$ is usually composed of a featurizer $\varphi : \mathcal{X} \to \mathcal{Z}$ that learns to extract useful features, and a classifier $w : \mathcal{Z} \to \mathcal{Y}$ that makes predictions from the extracted features.

Since we are interested in analyzing the feature learning process where the OOD objective succeeds in learning the invariant features. In the discussion below, without loss of generality, we analyze the IRMv1 objective and the data model where IRMv1 succeeds, as IRMv1 is one of the most widely discussed OOD objective from IRM frameworkArjovsky et al. (2019). Specifically, the IRM framework approaches OOD generalization by finding an invariant representation φ, such that there exists a classifier acting on φ that is simultaneously optimal in $\mathcal{E}_{\text{tr}}$. Hence, IRM leads to a challenging bi-level optimization problem as

$$\min_{w, \varphi} \sum_{e \in \mathcal{E}_{\text{tr}}} L_e(w \circ \varphi), \text{s.t. } w \in \operatorname*{arg\,min}_{\bar{w} : \mathcal{Z} \to \mathcal{Y}} L_e(\bar{w} \circ \varphi), \ \forall e \in \mathcal{E}_{\text{tr}}. \tag{7}$$

Due to the optimization difficulty of Eq. (7), Arjovsky et al. (2019) relax Eq. (7) into IRMv1 as follows:

$$\min_{\varphi} \sum_{e \in \mathcal{E}_{\text{tr}}} L_e(\varphi) + \lambda |\nabla_{w|w=1} L_e(w \cdot \varphi)|^2. \tag{8}$$

[3] When the environment e is not explicitly considered, we will omit it for clarity.

[4] It is also partially due to the complexity of IRMv1 dynamics, though to the best of our knowledge, we are the first to directly study the IRMv1 dynamics.

Given the convolutional neural network (Eq. 5) and logistic loss (Eq. 6), IRMv1 can be written as

$$L_{\text{IRMv1}}(\mathbf{W}) = \sum_{e\in\mathcal{E}_{\text{tr}}} \frac{1}{n_e}\sum_{i=1}^{n_e} \ell\left(y_i^e \cdot f(\mathbf{W},\mathbf{x}_i^e)\right) + \sum_{e\in\mathcal{E}_{\text{tr}}} \frac{\lambda}{n_e^2}\left(\sum_{i=1}^{n_e} \ell_i' \cdot y_i^e \cdot f(\mathbf{W},\mathbf{x}_i^e)\right)^2, \tag{9}$$

where ${\ell'}_i^e = \ell'(y_i^e \cdot f(\mathbf{W},\mathbf{x}_i^e)) = -\frac{\exp(-y_i^e\cdot f(\mathbf{W},\mathbf{x}_i^e))}{1+\exp(-y_i^e\cdot f(\mathbf{W},\mathbf{x}_i^e))}$. In other words, we define the featurizer φ as the whole CNN model and the classifier w as the scalar 1, following the relaxations introduced in IRMv1. Due to the complexity of IRMv1, in the analysis below, we introduce C^e_{IRMv1} for the ease of expressions. Specifically,

$$C^e_{\text{IRMv1}} \triangleq \frac{1}{n_e}\sum_{i=1}^{n_e} \ell'\left(y_i^e\hat{y}_i^e\right)\cdot y_i^e\hat{y}_i^e,$$

where $\hat{y}_i^e \triangleq f(\mathbf{W},\mathbf{x}_i^e)$ is the logit of sample i from environment e. The convergence of C_{IRMv1} indicates the convergence of IRMv1 penalty. The following definition will be useful in our analysis of IRMv1 objective.

Definition B.2. Let $\mathbf{w}_{j,r}(t)$[5] for $j\in\{+1,-1\}$ and $r\in[m]$ be the convolution filters of the CNN at the t-th iteration of gradient descent. Then there exists unique coefficients $\gamma_{j,r,1}(t), \gamma_{j,r,2}(t) \geq 0$ and $\rho_{j,r,i}(t)$ such that,

$$\mathbf{w}_{j,r}(t) = \mathbf{w}_{j,r}(0) + j\cdot\gamma_{j,r,1}(t)\cdot\mathbf{v}_1 + j\cdot\gamma_{j,r,2}(t)\cdot\mathbf{v}_2 + \sum_{i=1}^{n}\rho_{j,r,i}(t)\cdot\|\boldsymbol{\xi}_i\|_2^{-2}\cdot\boldsymbol{\xi}_i. \tag{10}$$

We refer Eq.(10) as the *signal-noise decomposition* of $\mathbf{w}_{j,r}(t)$. We add normalization factor $\|\boldsymbol{\xi}_i\|_2^{-2}$ in the definition so that $\rho_{j,r}^{(t)} \approx \langle \mathbf{w}_{j,r}^{(t)}, \boldsymbol{\xi}_i\rangle$. Note that $\|\mathbf{v}_1\|_2 = \|\mathbf{v}_2\|_2 = 1$, the corresponding normalization factors are thus neglected.

C Proofs for theoretical results

C.1 Implementation details of the synthetic CNN experiments

The logit $\hat{y}_i^e$ (which is a function of $\mathbf{W}$) of sample i in the environment e can be explicitly written as

$$\hat{y}_i^e = f(\mathbf{W},\mathbf{x}_i^e) = F_{+1}(\mathbf{W}_{+1},\mathbf{x}_i^e) - F_{-1}(\mathbf{W}_{-1},\mathbf{x}_i^e) = \sum_{j\in\{\pm1\}}\frac{j}{m}\sum_{r=1}^{m}\left[\mathbf{w}_{j,r}^\top(\mathbf{x}_{i,1}^e + \mathbf{x}_{i,2}^e)\right],$$

where $\mathbf{W} \triangleq \{\mathbf{W}_{+1},\mathbf{W}_{-1}\}$ and $\mathbf{W}_j \triangleq \begin{bmatrix}\mathbf{w}_{j,1}^\top\\ \vdots \\ \mathbf{w}_{j,m}^\top\end{bmatrix}$ for $j\in\{\pm1\}$. We initialized all the network weights as $\mathcal{N}(0,\sigma_0^2)$ and we set $\sigma_0 = 0.01$.

The test dataset $(\mathbf{x}, y)$ is generated through

$$\mathbf{x}_{i,1} = y_i\cdot\mathbf{v}_1 + y_i\cdot\text{Rad}(1-\beta_e)\cdot\mathbf{v}_2, \qquad \mathbf{x}_{i,2} = \boldsymbol{\xi},$$

where half of the dataset uses $\text{Rad}(1-\beta_1)$ and the other half uses $\text{Rad}(1-\beta_2)$. Here $\boldsymbol{\xi}\sim\mathcal{N}(0,\sigma_p^2\cdot(\mathbf{I}_d - \mathbf{v}_1\mathbf{v}_1^\top - \mathbf{v}_2\mathbf{v}_2^\top))$ and we chose $\sigma_p = 0.01$.

From the definition of IRMv1, we take derivative wrt the scalar 1 of the logit $1\cdot\hat{y}_i^e$. Thus, for environment e, the penalty is

$$\left(\frac{1}{n_e}\sum_{i=1}^{n_e}\nabla_{w|w=1}\ell\left(y_i^e(w\cdot\hat{y}_i^e)\right)\right)^2 = \left(\frac{1}{n_e}\sum_{i=1}^{n_e}\ell'\left(y_i^e\hat{y}_i^e\right)\cdot y_i^e\hat{y}_i^e\right)^2.$$

[5]We use $\mathbf{w}_{j,r}(t)$, $\mathbf{w}_{j,r}^{(t)}$ and $\mathbf{w}_{j,r}^t$ interchangeably.

Then, the IRMv1 objective is (we set $n_1 = n_2 = 2500$ in the simulation)

$$L_{\text{IRMv1}}(\mathbf{W}) = \sum_{e\in\mathcal{E}_{tr}} \frac{1}{n_e}\sum_{i=1}^{n_e} \ell\big(y_i^e \hat{y}_i^e\big) + \lambda \sum_{e\in\mathcal{E}_{tr}} \left(\frac{1}{n_e}\sum_{i=1}^{n_e} \ell'\big(y_i^e \hat{y}_i^e\big)\cdot y_i^e \hat{y}_i^e\right)^2.$$

We used constant stepsize GD to minimize $L_{\text{IRMv1}}(\mathbf{W})$, and we chose $\lambda = 10^8$ (heavy regularization setup).

Let $C^e_{\text{IRMv1}} \triangleq \frac{1}{n_e}\sum_{i=1}^{n_e} \ell'\big(y_i^e \hat{y}_i^e\big)\cdot y_i^e \hat{y}_i^e$. The gradient of $L_{\text{IRMv1}}(\mathbf{W})$ with respect to each $\mathbf{w}_{j,r}$ can be explicitly written as

$$\begin{aligned}
\nabla_{\mathbf{w}_{j,r}} L_{\text{IRMv1}}(\mathbf{W}) = &\sum_{e\in\mathcal{E}_{tr}} \frac{1}{n_e}\sum_{i=1}^{n_e} \ell'\big(y_i^e \hat{y}_i^e\big)\cdot y_i^e \cdot \frac{j}{m}(\mathbf{x}_{i,1}^e + \mathbf{x}_{i,2}^e) \\
&+ 2\lambda \sum_{e\in\mathcal{E}_{tr}} \frac{C^e_{\text{IRMv1}}}{n_e}\sum_{i=1}^{n_e}\Big(\ell''\big(y_i^e \hat{y}_i^e\big)\cdot \hat{y}_i^e \cdot \frac{j}{m}(\mathbf{x}_{i,1}^e + \mathbf{x}_{i,2}^e) + \ell'\big(y_i^e \hat{y}_i^e\big)\cdot y_i^e \cdot \frac{j}{m}(\mathbf{x}_{i,1}^e + \mathbf{x}_{i,2}^e)\Big) \\
= &\sum_{e\in\mathcal{E}_{tr}} \frac{j}{n_e m}\sum_{i=1}^{n_e} \ell'\big(y_i^e \hat{y}_i^e\big)\cdot y_i^e \cdot (\mathbf{x}_{i,1}^e + \mathbf{x}_{i,2}^e) + 2\lambda \sum_{e\in\mathcal{E}_{tr}} \frac{jC^e_{\text{IRMv1}}}{n_e m}\sum_{i=1}^{n_e} \ell''\big(y_i^e \hat{y}_i^e\big)\cdot \hat{y}_i^e \cdot (\mathbf{x}_{i,1}^e + \mathbf{x}_{i,2}^e) \\
&+ 2\lambda \sum_{e\in\mathcal{E}_{tr}} \frac{jC^e_{\text{IRMv1}}}{n_e m}\sum_{i=1}^{n_e} \ell'\big(y_i^e \hat{y}_i^e\big)\cdot y_i^e \cdot (\mathbf{x}_{i,1}^e + \mathbf{x}_{i,2}^e) \\
= &\sum_{e\in\mathcal{E}_{tr}} \frac{j(1+2\lambda C^e_{\text{IRMv1}})}{n_e m}\sum_{i=1}^{n_e} \ell'\big(y_i^e \hat{y}_i^e\big)\cdot y_i^e \cdot (\mathbf{x}_{i,1}^e + \mathbf{x}_{i,2}^e) \\
&+ 2\lambda \sum_{e\in\mathcal{E}_{tr}} \frac{jC^e_{\text{IRMv1}}}{n_e m}\sum_{i=1}^{n_e} \ell''\big(y_i^e \hat{y}_i^e\big)\cdot \hat{y}_i^e \cdot (\mathbf{x}_{i,1}^e + \mathbf{x}_{i,2}^e).
\end{aligned}$$

Observe that C^e_{IRMv1} is in fact the scalar gradient $C^e_{\text{IRMv1}} = \nabla_{w|w=1} L^e_{\text{ERM}}(\mathbf{W})$ that we want to force zero, whose effect can be understood as a dynamic re-weighting of the ERM gradient. Due to its importance in the analysis and interpretation of IRMv1, we tracked C^e_{IRMv1} in our simulations.

The invariant and spurious feature learning terms that we tracked are the mean of $\langle \mathbf{w}_{j,r}, j\mathbf{v}_1\rangle$ and $\langle \mathbf{w}_{j,r}, j\mathbf{v}_2\rangle$ for $j \in \{\pm 1\}, r \in [m]$, respectively.

C.2 PROOF FOR THEOREM 3.1

Theorem C.1 (Formal statement of Theorem 3.1)**.** *For $\rho > 0$, denote $\underline{n} \triangleq \min_{e\in\mathcal{E}_{tr}} n_e$, $n \triangleq \sum_{e\in\mathcal{E}_{tr}} n_e$, $\epsilon_C \triangleq \sqrt{\frac{2\log(16/\rho)}{\underline{n}}}$ and $\delta \triangleq \exp\{O(\underline{n}^{-1})\} - 1$. Define the feature learning terms $\Lambda^t_{j,r} \triangleq \langle \mathbf{w}^t_{j,r}, j\mathbf{v}_1\rangle$ and $\Gamma^t_{j,r} \triangleq \langle \mathbf{w}^t_{j,r}, j\mathbf{v}_2\rangle$ for $j \in \{\pm 1\}, r \in [m]$. Suppose we run T iterations of GD for the ERM objective. With sufficiently large $\underline{n}$, assuming that*

$$\alpha, \beta_1, \beta_2 < \frac{1 - \epsilon_C - \delta(\frac{1}{4} + \frac{\epsilon_C}{2})}{2} \quad (\alpha, \beta_1, \beta_2 \text{ are sufficiently smaller than } \frac{1}{2}),$$
$$\alpha > \frac{\beta_1 + \beta_2}{2} + \epsilon_C + \frac{\delta(1+\epsilon_C)}{2} \quad (\alpha \text{ is sufficiently larger than } \frac{\beta_1+\beta_2}{2}),$$

and choosing

$$\sigma_0^2 = O\left(\underline{n}^{-2}\log^{-1}(m/\rho)\right),$$
$$\sigma_p^2 = O\left(\min\left\{d^{-1/2}\log^{-1/2}(nm/\rho), T^{-1}\eta^{-1}m\left(d + n\sqrt{d\log(n^2/\rho)}\right)^{-1}\right\}\right),$$

there exists a constant η, such that for any $j \in \{\pm 1\}, r \in [m]$, with probability at least $1 - 2\rho$, $\Lambda^t_{j,r}$ and $\Gamma^t_{j,r}$ are converging and the increment of the spurious feature $\Gamma^{t+1}_{j,r} - \Gamma^t_{j,r}$ is larger than that of the invariant feature $\Lambda^{t+1}_{j,r} - \Lambda^t_{j,r}$ at any iteration $t \in \{0, \ldots, T-1\}$.

Proof of Theorem C.1. We begin with checking the feature learning terms in the ERM stage using constant stepsize GD: $\mathbf{W}^{t+1} = \mathbf{W}^t - \eta \cdot \nabla_{\mathbf{W}} L_{\text{IRMv1}}(\mathbf{W}^t)$. Note that the update rule for each $\mathbf{w}_{j,r}, \forall j \in \{+1, -1\}, r \in [m]$ can be written as

$$\begin{aligned}\mathbf{w}^{t+1}_{j,r} &= \mathbf{w}^t_{j,r} - \frac{j\eta}{m}\sum_{e\in\mathcal{E}_{tr}}\frac{1}{n_e}\sum_{i=1}^{n_e}\ell'\left(y^e_i\hat{y}^e_i\right)\cdot y^e_i\cdot(\mathbf{x}^e_{i,1} + \mathbf{x}^e_{i,2})\\ &= \mathbf{w}^t_{j,r} - \frac{j\eta}{m}\sum_{e\in\mathcal{E}_{tr}}\frac{1}{n_e}\sum_{i=1}^{n_e}\ell'\left(y^e_i\hat{y}^e_i\right)\cdot(\text{Rad}(\alpha)_i\cdot\mathbf{v}_1 + \text{Rad}(\beta_e)_i\cdot\mathbf{v}_2 + y^e_i\boldsymbol{\xi}^e_i).\end{aligned}$$

Define the quantities of interest (the feature learning terms): $\Lambda^t_{j,r} \triangleq \langle \mathbf{w}^t_{j,r}, j\mathbf{v}_1\rangle, \Gamma^t_{j,r} \triangleq \langle \mathbf{w}^t_{j,r}, j\mathbf{v}_2\rangle, \Xi^{t,e}_{j,r,i} \triangleq \langle \mathbf{w}^t_{j,r}, j\boldsymbol{\xi}^e_i\rangle$. From our data generating procedure (Definition B.1), we know that the first two coordinates of $\boldsymbol{\xi}^e_i$ are zero. Thus, we can write down the update rule for each feature learning term as follows.

$$\Lambda^{t+1}_{j,r} = \Lambda^t_{j,r} - \frac{\eta}{m}\sum_{e\in\mathcal{E}_{tr}}\frac{1}{n_e}\sum_{i=1}^{n_e}\ell'\left(y^e_i\hat{y}^e_i\right)\cdot\text{Rad}(\alpha)_i,$$
$$\Gamma^{t+1}_{j,r} = \Gamma^t_{j,r} - \frac{\eta}{m}\sum_{e\in\mathcal{E}_{tr}}\frac{1}{n_e}\sum_{i=1}^{n_e}\ell'\left(y^e_i\hat{y}^e_i\right)\cdot\text{Rad}(\beta_e)_i,$$
$$\Xi^{t+1,e'}_{j,r,i'} = \Xi^{t,e'}_{j,r,i'} - \frac{\eta}{m}\sum_{e\in\mathcal{E}_{tr}}\frac{1}{n_e}\sum_{i=1}^{n_e}\ell'\left(y^e_i\hat{y}^e_i\right)\cdot y^e_i\cdot\langle\boldsymbol{\xi}^e_i, \boldsymbol{\xi}^{e'}_{i'}\rangle.$$

More explicitly, we can write

$$\Lambda^{t+1}_{j,r} = \Lambda^t_{j,r} + \frac{\eta}{m}\sum_{e\in\mathcal{E}_{tr}}\frac{1}{n_e}\sum_{i=1}^{n_e}\frac{\text{Rad}(\alpha)_i}{1+\exp\{y^e_i\hat{y}^e_i\}}, \tag{11}$$

$$\Gamma^{t+1}_{j,r} = \Gamma^t_{j,r} + \frac{\eta}{m}\sum_{e\in\mathcal{E}_{tr}}\frac{1}{n_e}\sum_{i=1}^{n_e}\frac{\text{Rad}(\beta_e)_i}{1+\exp\{y^e_i\hat{y}^e_i\}}, \tag{12}$$

$$\Xi^{t+1,e'}_{j,r,i'} = \Xi^{t,e'}_{j,r,i'} + \frac{\eta}{m}\sum_{e\in\mathcal{E}_{tr}}\frac{1}{n_e}\sum_{i=1}^{n_e}\frac{y^e_i\cdot\langle\boldsymbol{\xi}^e_i, \boldsymbol{\xi}^{e'}_{i'}\rangle}{1+\exp\{y^e_i\hat{y}^e_i\}}. \tag{13}$$

Notice that the updates (11), (12) for $\Lambda_{j,r}, \Gamma_{j,r}$ are independent of j, r. Denoting

$$\Delta_\Lambda^t \triangleq \frac{1}{m}\sum_{e\in\mathcal{E}_{tr}}\frac{1}{n_e}\sum_{i=1}^{n_e}\frac{\text{Rad}(\alpha)_i}{1+\exp\{y_i^e\hat{y}_i^e\}},$$
$$\Delta_\Gamma^t \triangleq \frac{1}{m}\sum_{e\in\mathcal{E}_{tr}}\frac{1}{n_e}\sum_{i=1}^{n_e}\frac{\text{Rad}(\beta_e)_i}{1+\exp\{y_i^e\hat{y}_i^e\}},$$

we can conclude that for any $j \in \{+1, -1\}, r \in [m]$,

$$\begin{aligned}\Lambda_{j,r}^{t+1} &= \Lambda_{j,r}^t + \eta\cdot\Delta_\Lambda^t = \eta\cdot\sum_{k=0}^t \Delta_\Lambda^k + \Lambda_{j,r}^0,\\ \Gamma_{j,r}^{t+1} &= \Gamma_{j,r}^t + \eta\cdot\Delta_\Gamma^t = \eta\cdot\sum_{k=0}^t \Delta_\Gamma^k + \Gamma_{j,r}^0.\end{aligned} \tag{14}$$

Then, we write the logit $\hat{y}_i^e$ as

$$\begin{aligned}\hat{y}_i^e &= \sum_{j\in\{\pm1\}}\frac{j}{m}\sum_{r=1}^m\left[\left\langle \mathbf{w}_{j,r}^t, y_i^e\cdot\text{Rad}(\alpha)_i\cdot\mathbf{v}_1 + y_i^e\cdot\text{Rad}(\beta_e)_i\cdot\mathbf{v}_2 + \mathbf{x}_{i,2}^e\right\rangle\right]\\
&= \sum_{j\in\{\pm1\}}\frac{j}{m}\sum_{r=1}^m\left[jy_i^e\cdot\text{Rad}(\alpha)_i\cdot\Lambda_{j,r}^t + jy_i^e\cdot\text{Rad}(\beta_e)_i\cdot\Gamma_{j,r}^t + j\cdot\Xi_{j,r,i}^{t,e}\right]\\
&= \sum_{j\in\{\pm1\}}\frac{1}{m}\sum_{r=1}^m\left[y_i^e\cdot\text{Rad}(\alpha)_i\cdot\Lambda_{j,r}^t + y_i^e\cdot\text{Rad}(\beta_e)_i\cdot\Gamma_{j,r}^t + \Xi_{j,r,i}^{t,e}\right]\\
&= y_i^e\cdot\text{Rad}(\alpha)_i\cdot\sum_{j\in\{\pm1\}}\sum_{r=1}^m\frac{\Lambda_{j,r}^t}{m} + y_i^e\cdot\text{Rad}(\beta_e)_i\cdot\sum_{j\in\{\pm1\}}\sum_{r=1}^m\frac{\Gamma_{j,r}^t}{m} + \sum_{j\in\{\pm1\}}\sum_{r=1}^m\frac{\Xi_{j,r,i}^{t,e}}{m}\\
&= y_i^e\cdot\text{Rad}(\alpha)_i\cdot 2\eta\cdot\sum_{k=0}^{t-1}\Delta_\Lambda^k + y_i^e\cdot\text{Rad}(\beta_e)_i\cdot 2\eta\cdot\sum_{k=0}^{t-1}\Delta_\Gamma^k\\
&\quad + y_i^e\cdot\text{Rad}(\alpha)_i\cdot\sum_{j\in\{\pm1\}}\sum_{r=1}^m\frac{\Lambda_{j,r}^0}{m} + y_i^e\cdot\text{Rad}(\beta_e)_i\cdot\sum_{j\in\{\pm1\}}\sum_{r=1}^m\frac{\Gamma_{j,r}^0}{m} + \sum_{j\in\{\pm1\}}\sum_{r=1}^m\frac{\Xi_{j,r,i}^{t,e}}{m}.\end{aligned}$$

Denoting $\mathbb{Q}_i^e \triangleq \text{Rad}(\alpha)_i\sum_{j\in\{\pm1\}}\sum_{r=1}^m\frac{\Lambda_{j,r}^0}{m} + \text{Rad}(\beta_e)_i\sum_{j\in\{\pm1\}}\sum_{r=1}^m\frac{\Gamma_{j,r}^0}{m} + y_i^e\cdot\sum_{j\in\{\pm1\}}\sum_{r=1}^m\frac{\Xi_{j,r,i}^{t,e}}{m}$, we have

$$\hat{y}_i^e = y_i^e\cdot\left(\text{Rad}(\alpha)_i\cdot 2\eta\cdot\sum_{k=0}^{t-1}\Delta_\Lambda^k + \text{Rad}(\beta_e)_i\cdot 2\eta\cdot\sum_{k=0}^{t-1}\Delta_\Gamma^k + \mathbb{Q}_i^e\right),$$

We need the following concentration lemma to control the scale of $\mathbb{Q}_i^e$, whose proof is given in Appendix C.2.1.

Lemma C.2. *Denote $\underline{n} \triangleq \min_{e\in\mathcal{E}_{tr}} n_e, n \triangleq \sum_{e\in\mathcal{E}_{tr}} n_e$. For $\rho > 0$, if*

$$\begin{aligned}\sigma_0^2 &= O\left(\underline{n}^{-2}\log^{-1}(m/\rho)\right),\\
\sigma_p^2 &= O\left(\min\left\{d^{-1/2}\log^{-1/2}(nm/\rho), T^{-1}\eta^{-1}m\left(d + n\sqrt{d\log(n^2/\rho)}\right)^{-1}\right\}\right),\end{aligned}$$

then with probability at least $1-\rho$, for any $e \in \mathcal{E}_{tr}, i \in [n_e]$, it holds that $|\mathbb{Q}_i^e| = O(\underline{n}^{-1})$.

Then Δ_Λ^t and Δ_Γ^t can be explicitly written as

$$\Delta_\Lambda^t = \sum_{e\in\mathcal{E}_{tr}} \frac{1}{n_e m} \sum_{i=1}^{n_e} \frac{\text{Rad}(\alpha)_i}{1+\exp\left\{\text{Rad}(\alpha)_i \cdot 2\eta \cdot \sum_{k=0}^{t-1}\Delta_\Lambda^k\right\}\cdot \exp\left\{\text{Rad}(\beta_e)_i \cdot 2\eta \cdot \sum_{k=0}^{t-1}\Delta_\Gamma^k\right\}\cdot \exp\left\{\mathbb{Q}_i^e\right\}},$$

$$\Delta_\Gamma^t = \sum_{e\in\mathcal{E}_{tr}} \frac{1}{n_e m} \sum_{i=1}^{n_e} \frac{\text{Rad}(\beta_e)_i}{1+\exp\left\{\text{Rad}(\alpha)_i \cdot 2\eta \cdot \sum_{k=0}^{t-1}\Delta_\Lambda^k\right\}\cdot \exp\left\{\text{Rad}(\beta_e)_i \cdot 2\eta \cdot \sum_{k=0}^{t-1}\Delta_\Gamma^k\right\}\cdot \exp\left\{\mathbb{Q}_i^e\right\}}.$$

We are going to analyze the convergences of two sequences $\{\Delta_\Gamma^t + \Delta_\Lambda^t\}$ and $\{\Delta_\Gamma^t - \Delta_\Lambda^t\}$. Notice that

$$\Delta_\Gamma^t + \Delta_\Lambda^t = \sum_{e\in\mathcal{E}_{tr}} \frac{1}{n_e m} \sum_{i=1}^{n_e} \frac{\text{Rad}(\beta_e)_i + \text{Rad}(\alpha)_i}{1+\exp\left\{\text{Rad}(\alpha)_i \cdot 2\eta \cdot \sum_{k=0}^{t-1}\Delta_\Lambda^k\right\}\cdot \exp\left\{\text{Rad}(\beta_e)_i \cdot 2\eta \cdot \sum_{k=0}^{t-1}\Delta_\Gamma^k\right\}\cdot \exp\left\{\mathbb{Q}_i^e\right\}},$$

$$\Delta_\Gamma^t - \Delta_\Lambda^t = \sum_{e\in\mathcal{E}_{tr}} \frac{1}{n_e m} \sum_{i=1}^{n_e} \frac{\text{Rad}(\beta_e)_i - \text{Rad}(\alpha)_i}{1+\exp\left\{\text{Rad}(\alpha)_i \cdot 2\eta \cdot \sum_{k=0}^{t-1}\Delta_\Lambda^k\right\}\cdot \exp\left\{\text{Rad}(\beta_e)_i \cdot 2\eta \cdot \sum_{k=0}^{t-1}\Delta_\Gamma^k\right\}\cdot \exp\left\{\mathbb{Q}_i^e\right\}}.$$

We can further write these two terms as

$$\begin{aligned}
\Delta_\Gamma^t + \Delta_\Lambda^t = &\sum_{e\in\mathcal{E}_{tr}} \frac{2}{n_e m} \sum_{\substack{i\in[n_e]\\ \text{Rad}(\beta_e)_i=+1\\ \text{Rad}(\alpha)_i=+1}} \frac{1}{1+\exp\left\{2\eta\cdot\sum_{k=0}^{t-1}\left(\Delta_\Gamma^k+\Delta_\Lambda^k\right)\right\}\cdot\exp\left\{\mathbb{Q}_i^e\right\}} \\
&- \sum_{e\in\mathcal{E}_{tr}} \frac{2}{n_e m} \sum_{\substack{i\in[n_e]\\ \text{Rad}(\beta_e)_i=-1\\ \text{Rad}(\alpha)_i=-1}} \frac{1}{1+\exp\left\{-2\eta\cdot\sum_{k=0}^{t-1}\left(\Delta_\Gamma^k+\Delta_\Lambda^k\right)\right\}\cdot\exp\left\{\mathbb{Q}_i^e\right\}}, \\
\Delta_\Gamma^t - \Delta_\Lambda^t = &\sum_{e\in\mathcal{E}_{tr}} \frac{2}{n_e m} \sum_{\substack{i\in[n_e]\\ \text{Rad}(\beta_e)_i=+1\\ \text{Rad}(\alpha)_i=-1}} \frac{1}{1+\exp\left\{2\eta\cdot\sum_{k=0}^{t-1}\left(\Delta_\Gamma^k-\Delta_\Lambda^k\right)\right\}\cdot\exp\left\{\mathbb{Q}_i^e\right\}} \\
&- \sum_{e\in\mathcal{E}_{tr}} \frac{2}{n_e m} \sum_{\substack{i\in[n_e]\\ \text{Rad}(\beta_e)_i=-1\\ \text{Rad}(\alpha)_i=+1}} \frac{1}{1+\exp\left\{-2\eta\cdot\sum_{k=0}^{t-1}\left(\Delta_\Gamma^k-\Delta_\Lambda^k\right)\right\}\cdot\exp\left\{\mathbb{Q}_i^e\right\}}.
\end{aligned}$$

According to Lemma C.2, for all $e \in \mathcal{E}_{tr}, i \in [n_e], \rho > 0$, letting $\delta \triangleq \exp\{O(\underline{n}^{-1})\} - 1$, we have $1+\delta \geq \exp\{\mathbb{Q}_i^e\} \geq (1+\delta)^{-1}$ with probability at least $1-\rho$. Let $C_{j\ell}^e \triangleq |\{i \mid \text{Rad}(\alpha)_i = j, \text{Rad}(\beta_e)_i = \ell, i \in \mathcal{E}_e\}|$ for any $j \in \{\pm 1\}, \ell \in \{\pm 1\}, e \in \mathcal{E}_{tr}$, and then define $\overline{C}_{j\ell} \triangleq \sum_{e\in\mathcal{E}_{tr}} \frac{C_{j\ell}^e}{n_e}$. We can upper bound and formulate $\Delta_\Gamma^t + \Delta_\Lambda^t$ and $\Delta_\Gamma^t - \Delta_\Lambda^t$ as

$$\begin{aligned}
\Delta_\Gamma^t + \Delta_\Lambda^t &\leq \frac{2}{m}\left(\frac{\overline{C}_{+1+1}}{1+\exp\left\{2\eta\cdot\sum_{k=0}^{t-1}\left(\Delta_\Gamma^k+\Delta_\Lambda^k\right)\right\}\cdot(1+\delta)^{-1}} - \frac{\overline{C}_{-1-1}}{1+\exp\left\{-2\eta\cdot\sum_{k=0}^{t-1}\left(\Delta_\Gamma^k+\Delta_\Lambda^k\right)\right\}\cdot(1+\delta)}\right) \\
&= \frac{2}{m}\cdot\frac{\overline{C}_{+1+1}(1+\delta) - \overline{C}_{-1-1}\cdot\exp\left\{2\eta\cdot\sum_{k=0}^{t-1}\left(\Delta_\Gamma^k+\Delta_\Lambda^k\right)\right\}}{1+\delta+\exp\left\{2\eta\cdot\sum_{k=0}^{t-1}\left(\Delta_\Gamma^k+\Delta_\Lambda^k\right)\right\}}, \qquad (15)
\end{aligned}$$

$$\begin{aligned}
\Delta_\Gamma^t - \Delta_\Lambda^t &\leq \frac{2}{m}\left(\frac{\overline{C}_{-1+1}}{1+\exp\left\{2\eta\cdot\sum_{k=0}^{t-1}\left(\Delta_\Gamma^k-\Delta_\Lambda^k\right)\right\}\cdot(1+\delta)^{-1}} - \frac{\overline{C}_{+1-1}}{1+\exp\left\{-2\eta\cdot\sum_{k=0}^{t-1}\left(\Delta_\Gamma^k-\Delta_\Lambda^k\right)\right\}\cdot(1+\delta)}\right) \\
&= \frac{2}{m}\cdot\frac{\overline{C}_{-1+1}(1+\delta) - \overline{C}_{+1-1}\cdot\exp\left\{2\eta\cdot\sum_{k=0}^{t-1}\left(\Delta_\Gamma^k-\Delta_\Lambda^k\right)\right\}}{1+\delta+\exp\left\{2\eta\cdot\sum_{k=0}^{t-1}\left(\Delta_\Gamma^k-\Delta_\Lambda^k\right)\right\}}. \qquad (16)
\end{aligned}$$

Based on similar arguments, we can also establish lower bounds for these two terms,

$$\Delta_\Gamma^t + \Delta_\Lambda^t \geq \frac{2}{m} \cdot \frac{\overline{C}_{+1+1} - \overline{C}_{-1-1}(1+\delta) \cdot \exp\left\{2\eta \cdot \sum_{k=0}^{t-1} (\Delta_\Gamma^k + \Delta_\Lambda^k)\right\}}{1 + \exp\left\{2\eta \cdot \sum_{k=0}^{t-1} (\Delta_\Gamma^k + \Delta_\Lambda^k)\right\} \cdot (1+\delta)}, \tag{17}$$

$$\Delta_\Gamma^t - \Delta_\Lambda^t \geq \frac{2}{m} \cdot \frac{\overline{C}_{-1+1} - \overline{C}_{+1-1}(1+\delta) \cdot \exp\left\{2\eta \cdot \sum_{k=0}^{t-1} (\Delta_\Gamma^k - \Delta_\Lambda^k)\right\}}{1 + \exp\left\{2\eta \cdot \sum_{k=0}^{t-1} (\Delta_\Gamma^k - \Delta_\Lambda^k)\right\} \cdot (1+\delta)}. \tag{18}$$

The upper and lower bounds (15), (16), (17) and (18) imply that the convergences of $\{\Delta_\Gamma^t + \Delta_\Lambda^t\}$ and $\{\Delta_\Gamma^t - \Delta_\Lambda^t\}$ are determined by recursive equations of the form $\mathcal{Q}^t = \frac{C_1 - C_2 \cdot \exp\{\eta \sum_{k=0}^{t-1} \mathcal{Q}^k\}}{1 + C_3 \cdot \exp\{\eta \sum_{k=0}^{t-1} \mathcal{Q}^k\}}$. We first establish that with suitably chosen η, the sequences $\{\Delta_\Gamma^t + \Delta_\Lambda^t\}$ and $\{\Delta_\Gamma^t - \Delta_\Lambda^t\}$ are guaranteed to be positive. Observed that for the $\mathcal{Q}^t$-type recursive equation, the sign of $\mathcal{Q}^0$ is independent of η, and only determined by the constants C_1, C_2, C_3. At iteration 0, (17) and (18) give

$$\Delta_\Gamma^0 + \Delta_\Lambda^0 \geq \frac{2}{m} \cdot \frac{\overline{C}_{+1+1} - \overline{C}_{-1-1}(1+\delta)}{2+\delta}, \tag{19}$$

$$\Delta_\Gamma^0 - \Delta_\Lambda^0 \geq \frac{2}{m} \cdot \frac{\overline{C}_{-1+1} - \overline{C}_{+1-1}(1+\delta)}{2+\delta}. \tag{20}$$

To proceed, we need the following concentration lemma to control the deviations of the constants $\overline{C}_{+1+1}, \overline{C}_{+1-1}, \overline{C}_{-1+1}$ and $\overline{C}_{-1-1}$ from their expectations, whose proof is given in Appendix C.2.2.

Lemma C.3. *For $\rho > 0$, considering two environments and denoting $\epsilon_C \triangleq \sqrt{\frac{2\log(16/\rho)}{\underline{n}}}$, with probability at least $1-\rho$, we have*

$$\begin{aligned} \left|\overline{C}_{+1+1} - (1-\alpha)(2-\beta_1-\beta_2)\right| &\leq \epsilon_C, \\ \left|\overline{C}_{+1-1} - (1-\alpha)(\beta_1+\beta_2)\right| &\leq \epsilon_C, \\ \left|\overline{C}_{-1+1} - \alpha(2-\beta_1-\beta_2)\right| &\leq \epsilon_C, \\ \left|\overline{C}_{-1-1} - \alpha(\beta_1+\beta_2)\right| &\leq \epsilon_C. \end{aligned} \tag{21}$$

Using Lemma C.3, with probability at least $1-\rho$, the constants $\overline{C}_{+1+1}, \overline{C}_{+1-1}, \overline{C}_{-1+1}$ and $\overline{C}_{-1-1}$ are close to their expectations.

Based on our assumptions that

$$\alpha, \beta_1, \beta_2 < \frac{1 - \epsilon_C - \delta(\frac{1}{4} + \frac{\epsilon_C}{2})}{2} \quad (\alpha, \beta_1, \beta_2 \text{ are sufficiently smaller than } \frac{1}{2}),$$

$$\alpha > \frac{\beta_1+\beta_2}{2} + \epsilon_C + \frac{\delta(1+\epsilon_C)}{2} \quad (\alpha \text{ is sufficiently larger than } \frac{\beta_1+\beta_2}{2}),$$

it can be verified that with probability at least $1-2\rho$, $\Delta_\Gamma^0 + \Delta_\Lambda^0 > 0, \Delta_\Gamma^0 - \Delta_\Lambda^0 > 0$.

Then, at iteration 1, from (17) and (18), we see that as long as we require

$$\eta < \min\left\{\frac{1}{2(\Delta_\Gamma^0 + \Delta_\Lambda^0)} \log \frac{\overline{C}_{+1+1}}{\overline{C}_{-1-1}(1+\delta)}, \frac{1}{2(\Delta_\Gamma^0 - \Delta_\Lambda^0)} \log \frac{\overline{C}_{-1+1}}{\overline{C}_{+1-1}(1+\delta)}\right\},$$

it holds that $\Delta_\Gamma^1 + \Delta_\Lambda^1 > 0, \Delta_\Gamma^1 - \Delta_\Lambda^1 > 0$. By recursively applying this argument, we see the requirement for η to ensure that $\Delta_\Gamma^t + \Delta_\Lambda^t > 0$ and $\Delta_\Gamma^t - \Delta_\Lambda^t > 0$ for any $t \in \{0, \ldots, T\}$ is

$$\eta < \min\left\{\frac{1}{2\sum_{k=0}^{T-1} (\Delta_\Gamma^k + \Delta_\Lambda^k)} \log \frac{\overline{C}_{+1+1}}{\overline{C}_{-1-1}(1+\delta)}, \frac{1}{2\sum_{k=0}^{T-1} (\Delta_\Gamma^k - \Delta_\Lambda^k)} \log \frac{\overline{C}_{-1+1}}{\overline{C}_{+1-1}(1+\delta)}\right\}. \tag{22}$$

In other words, for the $\mathcal{Q}^t$-type recursive equation, as long as $\mathcal{Q}^0 \geq 0$, there always exists a sufficiently small η to guarantee that the whole sequence $\{\mathcal{Q}^t\}$ is positive. From now on, we will focus on the case where the two sequences $\{\Delta_\Gamma^t + \Delta_\Lambda^t\}$ and $\{\Delta_\Gamma^t - \Delta_\Lambda^t\}$ decrease to an $\epsilon_\Delta > 0$ error, i.e., $\min_{t \in \{0,\ldots,T\}} \{\Delta_\Gamma^t + \Delta_\Lambda^t, \Delta_\Gamma^t - \Delta_\Lambda^t\} = \epsilon_\Delta$.

Then, we show that the two sequences $\{\Delta_\Gamma^t + \Delta_\Lambda^t\}$ and $\{\Delta_\Gamma^t - \Delta_\Lambda^t\}$ decrease monotonically, which thus leads to a more refined upper bound for η at (22). Inspect the upper bounds (15), (16) at iteration $t+1$, which can be written as

$$\Delta_\Gamma^{t+1} + \Delta_\Lambda^{t+1} \leq \frac{2}{m} \cdot \frac{\overline{C}_{+1+1} - \overline{C}_{-1-1} \cdot \exp\left\{2\eta \cdot \sum_{k=0}^{t-1} (\Delta_\Gamma^k + \Delta_\Lambda^k)\right\} \cdot \exp\{2\eta \cdot (\Delta_\Gamma^t + \Delta_\Lambda^t)\}(1+\delta)^{-1}}{1 + \exp\left\{2\eta \cdot \sum_{k=0}^{t-1} (\Delta_\Gamma^k + \Delta_\Lambda^k)\right\} \cdot \exp\{2\eta \cdot (\Delta_\Gamma^t + \Delta_\Lambda^t)\}(1+\delta)^{-1}} \triangleq \spadesuit^{t+1},$$

$$\Delta_\Gamma^{t+1} - \Delta_\Lambda^{t+1} \leq \frac{2}{m} \cdot \frac{\overline{C}_{-1+1} - \overline{C}_{+1-1} \cdot \exp\left\{2\eta \cdot \sum_{k=0}^{t-1} (\Delta_\Gamma^k - \Delta_\Lambda^k)\right\} \cdot \exp\{2\eta \cdot (\Delta_\Gamma^t - \Delta_\Lambda^t)\}(1+\delta)^{-1}}{1 + \exp\left\{2\eta \cdot \sum_{k=0}^{t-1} (\Delta_\Gamma^k - \Delta_\Lambda^k)\right\} \cdot \exp\{2\eta \cdot (\Delta_\Gamma^t - \Delta_\Lambda^t)\}(1+\delta)^{-1}} \triangleq \clubsuit^{t+1}.$$

Requiring that $\eta > \max\left\{\frac{1}{\Delta_\Gamma^t + \Delta_\Lambda^t}\log(1+\delta), \frac{1}{\Delta_\Gamma^t - \Delta_\Lambda^t}\log(1+\delta)\right\}, \forall t \in \{0, \ldots, T\} \Rightarrow \eta > \epsilon_\Delta^{-1}\log(1+\delta)$, we have

$$\begin{aligned}
\spadesuit^{t+1} &< \frac{2}{m} \cdot \frac{\overline{C}_{+1+1} - \overline{C}_{-1-1} \cdot \exp\left\{2\eta \cdot \sum_{k=0}^{t-1} (\Delta_\Gamma^k + \Delta_\Lambda^k)\right\} \cdot \exp\{2\eta \cdot (\Delta_\Gamma^t + \Delta_\Lambda^t)\}(1+\delta)^{-1}}{1 + \exp\left\{2\eta \cdot \sum_{k=0}^{t-1} (\Delta_\Gamma^k + \Delta_\Lambda^k)\right\} \cdot (1+\delta)} \\
&< \Delta_\Gamma^t + \Delta_\Lambda^t, \\
\clubsuit^{t+1} &< \frac{2}{m} \cdot \frac{\overline{C}_{-1+1} - \overline{C}_{+1-1} \cdot \exp\left\{2\eta \cdot \sum_{k=0}^{t-1} (\Delta_\Gamma^k - \Delta_\Lambda^k)\right\} \cdot \exp\{2\eta \cdot (\Delta_\Gamma^t - \Delta_\Lambda^t)\}(1+\delta)^{-1}}{1 + \exp\left\{2\eta \cdot \sum_{k=0}^{t-1} (\Delta_\Gamma^k - \Delta_\Lambda^k)\right\} \cdot (1+\delta)} \\
&< \Delta_\Gamma^t - \Delta_\Lambda^t,
\end{aligned}$$

where the last inequalities use the lower bounds (17) and (18).

Based on the above discussion and (22), we can now clarify the requirements of η for the sequences $\{\Delta_\Gamma^t + \Delta_\Lambda^t\}$ and $\{\Delta_\Gamma^t - \Delta_\Lambda^t\}$ to be positive and monotonically decreasing:

$$\begin{aligned}
\epsilon_\Delta^{-1}\log(1+\delta) < \eta < \min\Bigg\{ &\frac{m(2+\delta)}{4T(\overline{C}_{+1+1}(1+\delta) - \overline{C}_{-1-1})}\log\frac{\overline{C}_{+1+1}}{\overline{C}_{-1-1}(1+\delta)}, \\
&\frac{m(2+\delta)}{4T(\overline{C}_{-1+1}(1+\delta) - \overline{C}_{+1-1})}\log\frac{\overline{C}_{-1+1}}{\overline{C}_{+1-1}(1+\delta)}\Bigg\},
\end{aligned} \tag{23}$$

which uses the upper bounds (15) and (16) at iteration 0. The constants $\overline{C}_{+1+1}, \overline{C}_{+1-1}, \overline{C}_{-1+1}$ and $\overline{C}_{-1-1}$ can be substituted using the concentration bounds at (21) to generate an upper bound for η that only involves $\alpha, \beta_1, \beta_2, m, \delta, T, \epsilon_C$. Here we omit the precise upper bound for clarity. Note that the left hand side of (23) approaches 0 if $\delta \to 0$, which means that there exists a constant choice of η in (23) if $\underline{n}$ is sufficiently large in Lemma C.2 and C.3.

To conclude, in view of (14), the convergences of the sequences $\{\Delta_\Gamma^t + \Delta_\Lambda^t\}$ and $\{\Delta_\Gamma^t - \Delta_\Lambda^t\}$ imply that $\Lambda_{j,r}^t$ and $\Gamma_{j,r}^t$ are converging, and the positive sequence $\{\Delta_\Gamma^t - \Delta_\Lambda^t\}$ indicates that the increment of the spurious feature $\Gamma_{j,r}^{t+1} - \Gamma_{j,r}^t$ is larger than that of the invariant feature $\Lambda_{j,r}^{t+1} - \Lambda_{j,r}^t$ at any iteration $t \in \{0, \ldots, T-1\}$. □

C.2.1 Proof of Lemma C.2

First, we recall some concentration inequalities for sub-Gaussian random variables. Since $\boldsymbol{\xi}_i^e \sim \mathcal{N}(0, \sigma_p^2 \cdot (\mathbf{I}_d - \mathbf{v}_1\mathbf{v}_1^\top - \mathbf{v}_2\mathbf{v}_2^\top))$, for $(i', e') \neq (i, e)$, using Bernstein's inequality for sub-exponential random variables, we have for sufficiently small $a \geq 0$,

$$\Pr\left\{|\langle \boldsymbol{\xi}_i^e, \boldsymbol{\xi}_{i'}^{e'}\rangle| \geq a\right\} \leq 2\exp\left\{-\frac{a^2}{4\sigma_p^4(d-2)}\right\},$$

$$\Pr\left\{\left|\|\boldsymbol{\xi}_i^e\|_2^2 - \sigma_p^2(d-2)\right| \geq a\right\} \leq 2\exp\left\{-\frac{a^2}{512\sigma_p^4(d-2)}\right\}.$$

Moreover, for $\xi_r \sim \mathcal{N}(0, \sigma_0^2)$ (indicating the initial weights $\mathbf{w}_{j,r}^0$), the standard Gaussian tail gives

$$\Pr\left\{\left|\frac{1}{m}\sum_{r=1}^{m}\xi_r\right| \geq a\right\} \leq 2\exp\left\{-\frac{ma^2}{2\sigma_0^2}\right\}.$$

Denote $n \triangleq \sum_{e\in\mathcal{E}_{tr}} n_e, \underline{n} \triangleq \min_{e\in\mathcal{E}_{tr}} n_e$, by properly choosing a for each tail bound and applying a union bound, we can conclude that for $\rho > 0$, with probability at least $1-\rho$, it holds that $\forall i, e, i', e', r$,

$$|\langle \boldsymbol{\xi}_i^e, \boldsymbol{\xi}_{i'}^{e'}\rangle| \leq 2\sigma_p^2\sqrt{(d-2)\log\frac{8n^2}{\rho}}, \quad \|\boldsymbol{\xi}_i^e\|_2^2 \leq \sigma_p^2(d-2) + 16\sigma_p^2\sqrt{2(d-2)\log\frac{8n}{\rho}},$$

$$\left|\frac{1}{m}\sum_{r=1}^{m}\xi_r\right| \leq \sigma_0\sqrt{\frac{2}{m}\log\frac{32m}{\rho}}, \qquad |\langle \boldsymbol{\xi}_r, \boldsymbol{\xi}_{i'}^{e'}\rangle| \leq 2\sigma_p\sigma_0\sqrt{(d-2)\log\frac{16nm}{\rho}}.$$

We start with bound the growth of $\Xi_{j,r,i}^{t,e}$. By bounding the update rule (13), with probability at least $1-\rho$, we have

$$\begin{aligned}
\left|\Xi_{j,r,i'}^{t+1,e'}\right| &\leq \left|\Xi_{j,r,i'}^{t,e'}\right| + \frac{\eta}{m}\sum_{e\in\mathcal{E}_{tr}}\frac{1}{n_e}\sum_{i=1}^{n_e}\frac{1}{1+\exp\{y_i^e\hat{y}_i^e\}}\cdot|\langle \boldsymbol{\xi}_i^e, \boldsymbol{\xi}_{i'}^{e'}\rangle| \\
&\leq \left|\Xi_{j,r,i'}^{t,e'}\right| + \frac{\eta}{m}\sum_{e\in\mathcal{E}_{tr}}\frac{1}{n_e}\sum_{i=1}^{n_e}|\langle \boldsymbol{\xi}_i^e, \boldsymbol{\xi}_{i'}^{e'}\rangle| \\
&= \left|\Xi_{j,r,i'}^{0,e'}\right| + (t+1)\cdot\frac{\eta}{m}\sum_{e\in\mathcal{E}_{tr}}\frac{1}{n_e}\sum_{i=1}^{n_e}|\langle \boldsymbol{\xi}_i^e, \boldsymbol{\xi}_{i'}^{e'}\rangle| \\
&= |\langle \boldsymbol{\xi}_r, \boldsymbol{\xi}_{i'}^{e'}\rangle| + (t+1)\cdot\left(\frac{\eta}{mn_{e'}}\|\boldsymbol{\xi}_{i'}^{e'}\|_2^2 + \sum_{(i,e)\neq(i',e')}\frac{\eta}{mn_e}|\langle \boldsymbol{\xi}_i^e, \boldsymbol{\xi}_{i'}^{e'}\rangle|\right) \\
&\leq 2\sigma_p\sigma_0\sqrt{(d-2)\log\frac{16nm}{\rho}} \\
&\quad + \frac{T\eta\sigma_p^2}{m\underline{n}}\left((d-2) + 16\sqrt{2(d-2)\log\frac{8n}{\rho}} + 2n\sqrt{(d-2)\log\frac{8n^2}{\rho}}\right).
\end{aligned}$$

Then, we can bound $|\mathbb{Q}_i^e|$ as

$$\begin{aligned}
|\mathbb{Q}_i^e| &\leq 2\cdot\left|\frac{1}{m}\sum_{r=1}^{m}\xi_r\right| + 2\cdot\left|\frac{1}{m}\sum_{r=1}^{m}\xi_r\right| + \frac{2}{m}\sum_{r=1}^{m}\left|\Xi_{j,r,i}^{t,e}\right| \\
&\leq 4\sigma_0\sqrt{\frac{2}{m}\log\frac{32m}{\rho}} + 4\sigma_p\sigma_0\sqrt{(d-2)\log\frac{16nm}{\rho}} \\
&\quad + \frac{2T\eta\sigma_p^2}{m\underline{n}}\left((d-2) + 16\sqrt{2(d-2)\log\frac{8n}{\rho}} + 2n\sqrt{(d-2)\log\frac{8n^2}{\rho}}\right).
\end{aligned}$$

Thus, with sufficient small σ_0, σ_p, i.e.,

$$\begin{aligned}
\sigma_0^2 &= O\left(\underline{n}^{-2}\log^{-1}(m/\rho)\right), \\
\sigma_p^2 &= O\left(\min\left\{d^{-1/2}\log^{-1/2}(nm/\rho), T^{-1}\eta^{-1}m\left(d + n\sqrt{d\log(n^2/\rho)}\right)^{-1}\right\}\right),
\end{aligned}$$

we ensured that $|\mathbb{Q}_i^e| = O(\underline{n}^{-1})$.

C.2.2 Proof of Lemma C.3

For $e \in \mathcal{E}_{tr}$, using Hoeffding's inequality, it holds that

$$\Pr\left\{\left|\frac{1}{n_e}\sum_{i=1}^{n_e}\mathbf{1}_{\{\mathrm{Rad}(\alpha)_i=+1, \mathrm{Rad}(\beta_e)_i=+1\}} - (1-\alpha)(1-\beta_e)\right| \geq a\right\} \leq 2\exp\{-2a^2 n_e\}.$$

Considering two environments, using a union bound, we can conclude that

$$\Pr\left\{\left|\overline{C}_{+1+1} - (1-\alpha)(2-\beta_1-\beta_2)\right| \leq a\right\} \geq 1 - 4\exp\left\{-\frac{a^2\underline{n}}{2}\right\}.$$

Thus, for $\rho > 0$, with probability at least $1 - \frac{\rho}{4}$, we can conclude that

$$\left|\overline{C}_{+1+1} - (1-\alpha)(2-\beta_1-\beta_2)\right| \leq \sqrt{\frac{2\log(16/\rho)}{\underline{n}}}.$$

Using the above arguments for other constants $\overline{C}_{+1-1}$, $\overline{C}_{-1+1}$ and $\overline{C}_{-1-1}$, and applying a union bound, we obtain the desired results.

C.3 PROOF FOR THEOREM 3.2

Theorem C.4 (Restatement of Theorem 3.2)**.** *Consider training a CNN model with the same data as in Theorem 3.1, define*

$$\mathbf{c}(t) \triangleq \left[C^1_{IRMv1}(\mathbf{W},t), C^2_{IRMv1}(\mathbf{W},t), \cdots, C^{|\mathcal{E}_{tr}|}_{IRMv1}(\mathbf{W},t)\right],$$

and $\lambda_0 = \lambda_{\min}(\mathbf{H}^\infty)$*, where* $\mathbf{H}^\infty_{e,e'} \triangleq \frac{1}{2mn_e n_{e'}} \sum_{i=1}^{n_e} \mathbf{x}^{e\top}_{1,i} \sum_{i'=1}^{n_{e'}} \mathbf{x}^{e'}_{1,i}$*. Suppose that dimension* $d = \Omega(\log(m/\delta))$*, network width* $m = \Omega(1/\delta)$*, regularization factor* $\lambda \geq 1/\sigma_0$*, noise variance* $\sigma_p = O(d^{-2})$*, weight initial scale* $\sigma_0 = O(\min\{\frac{\lambda_0^2 m^2}{\log(1/\epsilon)}, \frac{\lambda_0 m}{\sqrt{d}\log(1/\epsilon)}\})$*, then with probability at least* $1-\delta$*, after training time* $T = \Omega\left(\frac{\log(1/\epsilon)}{\eta\lambda\lambda_0}\right)$*, we have:*

$$\|\mathbf{c}(T)\|_2 \leq \epsilon, \quad \gamma_{j,r,1}(T) = o_d(1), \quad \gamma_{j,r,2}(T) = o_d(1).$$

Before proving Theorem C.4, we first provide some useful lemmas as follows:

Lemma C.5. *Suppose that* $\delta > 0$ *and* $d = \Omega(\log(4n/\delta))$*. Then with probability at least* $1-\delta$*,*

$$\sigma_p^2 d/2 \leq \|\boldsymbol{\xi}_i\|_2^2 \leq 3\sigma_p^2 d/2$$

for all $i, i' \in [n]$*.*

Proof of Lemma C.5. By Bernstein's inequality, with probability at least $1-\delta/(2n)$ we have

$$\left|\|\boldsymbol{\xi}_i\|_2^2 - \sigma_p^2 d\right| = O(\sigma_p^2 \cdot \sqrt{d\log(4n/\delta)}).$$

Therefore, as long as $d = \Omega(\log(4n/\delta))$, we have

$$\sigma_p^2 d/2 \leq \|\boldsymbol{\xi}_i\|_2^2 \leq 3\sigma_p^2 d/2.$$

□

Lemma C.6. *Suppose that* $d \geq \Omega(\log(mn/\delta))$*,* $m = \Omega(\log(1/\delta))$*. Then with probability at least* $1-\delta$*,*

$$|\langle \mathbf{w}^{(0)}_{j,r}, \mathbf{v}_1\rangle| \leq \sqrt{2\log(8m/\delta)} \cdot \sigma_0 \|\mathbf{v}_1\|_2,$$
$$|\langle \mathbf{w}^{(0)}_{j,r}, \mathbf{v}_2\rangle| \leq \sqrt{2\log(8m/\delta)} \cdot \sigma_0 \|\mathbf{v}_2\|_2,$$
$$|\langle \mathbf{w}^{(0)}_{j,r}, \boldsymbol{\xi}_i\rangle| \leq 2\sqrt{\log(8mn/\delta)} \cdot \sigma_0 \sigma_p \sqrt{d}$$

for all $r \in [m]$*,* $j \in \{\pm 1\}$ *and* $i \in [n]$*.*

Proof of Lemma C.6. It is clear that for each $r \in [m]$, $j \cdot \langle \mathbf{w}^{(0)}_{j,r}, \mathbf{v}_1\rangle$ is a Gaussian random variable with mean zero and variance $\sigma_0^2\|\mathbf{v}_1\|_2^2$. Therefore, by Gaussian tail bound and union bound, with probability at least $1-\delta$,

$$j \cdot \langle \mathbf{w}^{(0)}_{j,r}, \mathbf{v}_1\rangle \leq |\langle \mathbf{w}^{(0)}_{j,r}, \mathbf{v}_1\rangle| \leq \sqrt{2\log(8m/\delta)} \cdot \sigma_0 \|\mathbf{v}_1\|_2.$$

Similarly, we have

$$j \cdot \langle \mathbf{w}^{(0)}_{j,r}, \mathbf{v}_2\rangle \leq |\langle \mathbf{w}^{(0)}_{j,r}, \mathbf{v}_2\rangle| \leq \sqrt{2\log(8m/\delta)} \cdot \sigma_0 \|\mathbf{v}_2\|_2.$$

By Lemma C.5, with probability at least $1-\delta$, $\sigma_p\sqrt{d}/\sqrt{2} \leq \|\boldsymbol{\xi}_i\|_2 \leq \sqrt{3/2} \cdot \sigma_p\sqrt{d}$ for all $i \in [n]$. Therefore, the result for $\langle \mathbf{w}^{(0)}_{j,r}, \boldsymbol{\xi}_i\rangle$ follows the same proof as $j \cdot \langle \mathbf{w}^{(0)}_{j,r}, \mathbf{v}_1\rangle$. □

Proof. The proof is by induction method. First we show the gradient flow of weights by IRMv1 objective function (3):

$$
\begin{aligned}
\frac{d\mathbf{w}_{j,r}(t)}{dt} &= -\eta \cdot \nabla_{\mathbf{w}_{j,r}} L_{\text{IRMv1}}(\mathbf{W}(t)) \\
&= -\frac{\eta}{nm} \cdot \sum_{e\in\mathcal{E}_{\text{tr}}} \sum_{i=1}^{n_e} \ell_i'(t)\sigma'(\langle \mathbf{w}_{j,r}(t), y_i^e \mathbf{v}_i^e\rangle \cdot j\mathbf{v}_i^e - \frac{\eta}{nm} \cdot \sum_{e\in\mathcal{E}_{\text{tr}}} \sum_{i=1}^{n_e} \ell_i'(t)\sigma'(\langle \mathbf{w}_{j,r}(t), \boldsymbol{\xi}_i\rangle \cdot jy_i^e \boldsymbol{\xi}_i \\
&\quad - \frac{\eta\lambda}{nm} \cdot \sum_{e\in\mathcal{E}_{\text{tr}}} C^e_{\text{IRMv1}} \sum_{i=1}^{n_e} \ell_i'' \hat{y}_i^e \sigma'(\langle \mathbf{w}_{j,r}(t), y_i^e \mathbf{v}_i^e\rangle \cdot jy_i^e \mathrm{v}_i^e - \frac{\eta\lambda}{nm} \cdot \sum_{e\in\mathcal{E}_{\text{tr}}} C^e_{\text{IRMv1}} \sum_{i=1}^{n_e} \ell_i'' \hat{y}_i^e \sigma'(\langle \mathbf{w}_{j,r}(t), \boldsymbol{\xi}_i\rangle \cdot j\boldsymbol{\xi}_i \\
&\quad - \frac{\eta\lambda}{nm} \cdot \sum_{e\in\mathcal{E}_{\text{tr}}} C^e_{\text{IRMv1}} \sum_{i=1}^{n_e} \ell_i'(t)\sigma'(\langle \mathbf{w}_{j,r}(t), y_i^e \mathbf{v}_i^e\rangle \cdot j\mathbf{v}_i^e - \frac{\eta\lambda}{nm} \cdot \sum_{e\in\mathcal{E}_{\text{tr}}} C^e_{\text{IRMv1}} \sum_{i=1}^{n_e} \ell_i'(t)\sigma'(\langle \mathbf{w}_{j,r}(t), \boldsymbol{\xi}_i\rangle \cdot jy_i^e \boldsymbol{\xi}_i \\
&= -\frac{\eta}{nm} \cdot \sum_{e\in\mathcal{E}_{\text{tr}}} (1 + 2\lambda C^e_{\text{IRMv1}}(t)) \sum_{i=1}^{n_e} \ell_i'(t)\sigma'(\langle \mathbf{w}_{j,r}(t), y_i^e \mathbf{v}_i^e\rangle \cdot j\mathbf{v}_i^e \\
&\quad - \frac{\eta}{nm} \cdot \sum_{e\in\mathcal{E}_{\text{tr}}} (1 + 2\lambda C^e_{\text{IRMv1}}(t)) \sum_{i=1}^{n_e} \ell_i'(t)\sigma'(\langle \mathbf{w}_{j,r}(t), \boldsymbol{\xi}_i\rangle \cdot jy_i^e \boldsymbol{\xi}_i \\
&\quad - \frac{\eta\lambda}{nm} \cdot \sum_{e\in\mathcal{E}_{\text{tr}}} C^e_{\text{IRMv1}} \sum_{i=1}^{n_e} \ell_i'' \hat{y}_i^e \sigma'(\langle \mathbf{w}_{j,r}(t), y_i^e \mathbf{v}_i^e\rangle \cdot jy_i^e \mathrm{v}_i^e - \frac{\eta\lambda}{nm} \cdot \sum_{e\in\mathcal{E}_{\text{tr}}} C^e_{\text{IRMv1}} \sum_{i=1}^{n_e} \ell_i'' \hat{y}_i^e \sigma'(\langle \mathbf{w}_{j,r}(t), \boldsymbol{\xi}_i\rangle \cdot j\boldsymbol{\xi}_i,
\end{aligned}
$$

where $C^e_{\text{IRMv1}} = \frac{1}{n_e}\sum_{i=1}^{n_e} \ell_i' \hat{y}_i^e y_i^e$ and $\mathbf{v}_i^e = \text{Rad}(\alpha)_i \cdot \mathbf{v}_1 + \text{Rad}(\beta_e)_i \cdot \mathbf{v}_2$. Note that ℓ'' has the opposite sign to ℓ'.

Then we look at the dynamics of $C^e_{\text{IRMv1}}(t)$ according to the gradient flow update rule:

$$
\begin{aligned}
\frac{dC^e_{\text{IRMv1}}(\mathbf{W},t)}{dt} &= \sum_{j=\pm1} \sum_{r=1}^{m} \left\langle \frac{\partial C^e_{\text{IRMv1}}(\mathbf{W},t)}{\partial \mathbf{w}_{j,r}(t)}, \frac{d\mathbf{w}_{j,r}(t)}{dt} \right\rangle \\
&= \sum_{e'} 2\lambda C^{e'}_{\text{IRMv1}}(\mathbf{W},t) \sum_{j} \sum_{r=1}^{m} \left\langle \frac{\partial C^e_{\text{IRMv1}}(\mathbf{W},t)}{\partial \mathbf{w}_{j,r}(t)}, \frac{\partial C^{e'}_{\text{IRMv1}}(\mathbf{W},t)}{\partial \mathbf{w}_{j,r}(t)} \right\rangle \\
&\quad + \sum_{j=\pm1} \sum_{r=1}^{m} \left\langle \frac{\partial C^e_{\text{IRMv1}}(\mathbf{W},t)}{\partial \mathbf{w}_{j,r}(t)}, \frac{\partial L_s(\mathbf{W},t)}{\partial \mathbf{w}_{j,r}(t)} \right\rangle \\
&= 2\lambda \sum_{e'} C^{e'}_{\text{IRMv1}}(\mathbf{W},t) \cdot \mathbf{H}_{e,e'}(t) + \mathbf{g}_e(t),
\end{aligned}
$$

where we define $\mathbf{H}_{e,e'}(t) = \sum_j \sum_{r=1}^m \left\langle \frac{\partial C^e_{\text{IRMv1}}(\mathbf{W},t)}{\partial \mathbf{w}_{j,r}(t)}, \frac{\partial C^{e'}_{\text{IRMv1}}(\mathbf{W},t)}{\partial \mathbf{w}_{j,r}(t)} \right\rangle$ and $\mathbf{g}_e(t) = \sum_{j=\pm1} \sum_{r=1}^m \left\langle \frac{\partial C^e_{\text{IRMv1}}(\mathbf{W},t)}{\partial \mathbf{w}_{j,r}(t)}, \frac{\partial L_s(\mathbf{W},t)}{\partial \mathbf{w}_{j,r}(t)} \right\rangle$. Thus $\mathbf{H}(t)$ is an $|\mathcal{E}_{\text{tr}}| \times |\mathcal{E}_{\text{tr}}|$ matrix. We can write the dynamics of $\mathbf{c}(t) = \left[C^1_{\text{IRMv1}}(\mathbf{W},t), C^2_{\text{IRMv1}}(\mathbf{W},t), \cdots, C^{|\mathcal{E}_{\text{tr}}|}_{\text{IRMv1}}(\mathbf{W},t)\right]$ in a compact way:

$$
\frac{d\mathbf{c}(t)}{dt} = 2\lambda \cdot \mathbf{H}(t)\mathbf{c}(t) + \mathbf{g}(t).
$$

Our next step is to show $\mathbf{H}(t)$ is stable in terms of $\mathbf{W}(t)$. To proceed with the analysis, we write down the expression for $\frac{\partial C^e_{\text{IRMv1}}(\mathbf{W},t)}{\partial \mathbf{w}_{j,r}(t)} \in \mathbb{R}^d$:

$$
\begin{aligned}
\frac{\partial C^e_{\text{IRMv1}}(\mathbf{W}(t))}{\partial \mathbf{w}_{j,r}(t)} &= \frac{\eta}{n_e m} \cdot \sum_{i=1}^{n_e} \ell_i'(t)\sigma'(\langle \mathbf{w}_{j,r}(t), y_i^e \mathbf{v}_i^e\rangle) \cdot j\mathbf{v}_i^e + \frac{\eta}{n_e m} \sum_{i=1}^{n_e} \ell_i'(t)\sigma'(\langle \mathbf{w}_{j,r}(t), \boldsymbol{\xi}_i\rangle) \cdot jy_i^e \boldsymbol{\xi}_i \\
&\quad + \frac{\eta}{n_e m} \cdot \sum_{i=1}^{n_e} \ell_i'' y_i^e \sigma'(\langle \mathbf{w}_{j,r}(t), y_i^e \mathbf{v}_i^e\rangle) \cdot jy_i^e \mathbf{v}_i^e + \frac{\eta}{n_e m} \cdot \sum_{i=1}^{n_e} \ell_i'' y_i^e \sigma'(\langle \mathbf{w}_{j,r}(t), \boldsymbol{\xi}_i\rangle) \cdot j\boldsymbol{\xi}_i.
\end{aligned}
$$

When we consider linear activation function $\sigma(x) = x$, the entry of matrix $\mathbf{H}(t)$ reduces to:

$$
\begin{aligned}
\mathbf{H}_{e,e'}(t) &= \sum_j \sum_{r=1}^m \left\langle \frac{\partial C^e_{\text{IRMv1}}(\mathbf{W},t)}{\partial \mathbf{w}_{j,r}(t)}, \frac{\partial C^{e'}_{\text{IRMv1}}(\mathbf{W},t)}{\partial \mathbf{w}_{j,r}(t)} \right\rangle \\
&= \sum_j \sum_{r=1}^m \left(\frac{1}{n_e m}\right)\left(\frac{1}{n_{e'} m}\right) \left[\sum_{i=1}^{n_e} \ell_i'(t) \cdot j\mathbf{v}_i^{e\top} \sum_{i=1}^{n_{e'}} \ell_i'(t) \cdot j\mathbf{v}_i^e + \sum_{i=1}^{n_e} \ell_i''(t) y_i^e(t) \cdot j y_i^e \mathbf{v}_i^{e\top} \sum_{i=1}^{n_{e'}} \ell_i''(t) y_i^e(t) \cdot j y_i^e \mathbf{v}_i^e\right] \\
&+ \sum_j \sum_{r=1}^m \left(\frac{1}{n_e m}\right)\left(\frac{1}{n_{e'} m}\right) \left[\sum_{i=1}^{n_e} \ell_i''(t) \hat{y}_i^e(t) \cdot j y_i^e \mathbf{v}_i^{e\top} \sum_{i=1}^{n_{e'}} \ell_i'(t) \cdot j\mathbf{v}_i^e + \sum_{i=1}^{n_e} \ell_i'(t) \cdot j\mathbf{v}_i^{e\top} \sum_{i=1}^{n_{e'}} \ell_i''(t) \hat{y}_i^e(t) \cdot j\mathbf{v}_i^e\right] \\
&+ \sum_j \sum_{r=1}^m \left(\frac{1}{n_e m}\right)\left(\frac{1}{n_{e'} m}\right) \left[\sum_{i=1}^{n_e} \ell_i'(t) \cdot j y_i^e \boldsymbol{\xi}_i^{e\top} \sum_{i=1}^{n_{e'}} \ell_i'(t) \cdot j y_i^e \boldsymbol{\xi}_i^e + \sum_{i=1}^{n_e} \ell_i''(t) y_i^e(t) \cdot j\boldsymbol{\xi}_i^{e\top} \sum_{i=1}^{n_{e'}} \ell_i''(t) \cdot j\boldsymbol{\xi}_i^e\right] \\
&+ \sum_j \sum_{r=1}^m \left(\frac{1}{n_e m}\right)\left(\frac{1}{n_{e'} m}\right) \left[\sum_{i=1}^{n_e} \ell_i''(t) \cdot j\boldsymbol{\xi}_i^{e\top} \sum_{i=1}^{n_{e'}} \ell_i'(t) \cdot j y_i^e \boldsymbol{\xi}_i^e + \sum_{i=1}^{n_e} y_i^e \ell_i'(t) \cdot j\boldsymbol{\xi}_i^{e\top} \sum_{i=1}^{n_{e'}} \ell_i''(t) \cdot j\boldsymbol{\xi}_i^e\right] \\
&\triangleq \mathbf{H}^1_{e,e'}(t) + \mathbf{H}^2_{e,e'}(t) + \mathbf{H}^3_{e,e'}(t) + \mathbf{H}^4_{e,e'}(t) + \mathbf{H}^5_{e,e'}(t) + \mathbf{H}^6_{e,e'}(t) + \mathbf{H}^7_{e,e'}(t) + \mathbf{H}^8_{e,e'}(t).
\end{aligned}
$$

Define

$$
\begin{aligned}
\mathbf{H}^{1,\infty}_{e,e'} &= \sum_j \sum_{r=1}^m \left(\frac{1}{n_e m}\right)\left(\frac{1}{n_{e'} m}\right) \left[\sum_{i=1}^{n_e} -\frac{1}{2} \cdot j\mathbf{v}_i^{e\top} \sum_{i=1}^{n_{e'}} -\frac{1}{2} \cdot j\mathbf{v}_i^e\right] \\
&= \frac{1}{2mn_e n_{e'}} \sum_{i=1}^{n_e} \mathbf{v}_i^{e\top} \sum_{i'=1}^{n_{e'}} \mathbf{v}_i^{e'}.
\end{aligned}
$$

Then we can show that

$$
\begin{aligned}
\left|\mathbf{H}^1_{e,e'}(t) - \mathbf{H}^{1,\infty}_{e,e'}\right| &= \frac{2}{mn_e n_{e'}} \left|\sum_{i=1}^{n_e} \ell_i'(t) \mathbf{v}_i^{e\top} \sum_{i'=1}^{n_{e'}} \ell_i'(t) \mathbf{v}_i^{e'} - \sum_{i=1}^{n_e} \frac{1}{2} \mathbf{v}_i^{e\top} \sum_{i'=1}^{n_{e'}} \frac{1}{2} \mathbf{v}_i^{e'}\right| \\
&\le \frac{2}{mn_e n_{e'}} \left|\sum_{i=1}^{n_e} \ell_i'(t) \mathbf{v}_i^{e\top} \sum_{i'=1}^{n_{e'}} \ell_i'(t) \mathbf{v}_i^{e'} - \sum_{i=1}^{n_e} \ell_i' \mathbf{v}_i^{e\top} \sum_{i'=1}^{n_{e'}} \frac{1}{2} \mathbf{v}_i^{e'}\right| \\
&\quad + \frac{2}{mn_e n_{e'}} \left|\sum_{i=1}^{n_e} \ell_i'(t) \mathbf{v}_i^{e\top} \sum_{i'=1}^{n_{e'}} \frac{1}{2} \mathbf{v}_i^{e'} - \sum_{i=1}^{n_e} \frac{1}{2} \mathbf{v}_i^{e\top} \sum_{i'=1}^{n_{e'}} \frac{1}{2} \mathbf{v}_i^{e'}\right| \\
&\le \frac{2}{mn_e n_{e'}} \left|\sum_{i=1}^{n_e} \ell_i'(t) \mathbf{v}_i^{e\top} \sum_{i'=1}^{n_{e'}} \left(\ell_i'(t) + \frac{1}{2}\right) \mathbf{v}_i^{e'}\right| + \frac{2}{mn_e n_{e'}} \left|\sum_{i=1}^{n_e} \left(\ell_i'(t) + \frac{1}{2}\right) \mathbf{v}_i^{e\top} \sum_{i'=1}^{n_{e'}} \frac{1}{2} \mathbf{v}_i^{e'}\right| \\
&\le C\frac{2\gamma}{m},
\end{aligned}
$$

where C is an absolute constant, γ is defined as follows:

$$
|\hat{y}_i^e(t)| = |f(\mathbf{x}_i, t)| = \left|\frac{1}{m} \sum_j \sum_{r=1}^m \left[\sigma(\mathbf{w}_{j,r}^\top(t)\mathbf{x}_1) + \sigma(\mathbf{w}_{j,r}^\top(t)\mathbf{x}_2)\right]\right| \le \gamma,
$$

and we have used the bound for $\ell_i'(t) + \frac{1}{2}$:

$$
\begin{aligned}
\left|\ell_i'(t) + \frac{1}{2}\right| &= \left|-\frac{\exp(-y_i^e \cdot f(\mathbf{W}, \mathbf{x}_i, t))}{1 + \exp(-y_i^e \cdot f(\mathbf{W}, \mathbf{x}_i, t))} + \frac{1}{2}\right| \\
&= \left|\frac{1}{2} - \frac{1}{1 + \exp(y_i^e \cdot f(\mathbf{W}, \mathbf{x}_i, t))}\right| \\
&\le \max\left\{\left|\frac{1}{2} - \frac{1}{1 + \exp(\gamma)}\right|, \left|\frac{1}{2} - \frac{1}{1 + \exp(-\gamma)}\right|\right\} \\
&\le \max\left\{\left|\frac{1}{2} - \frac{1}{2 + \frac{7}{4}\gamma}\right|, \left|\frac{1}{2} - \frac{1}{2 - \gamma}\right|\right\} = \Theta(\gamma).
\end{aligned}
$$

and we provide the bound of $\ell_i''(t) - \frac{1}{4}$:

$$\begin{aligned}\left|\ell_i''(t) - \frac{1}{4}\right| &= \left|\frac{\exp(-y_i^e \cdot f(\mathbf{W}, \mathbf{x}_i, t))}{(1+\exp(-y_i^e \cdot f(\mathbf{W}, \mathbf{x}_i, t)))^2} - \frac{1}{4}\right| \\ &= \left|\frac{1}{\exp(y_i^e \cdot f(\mathbf{W}, \mathbf{x}_i, t)) + 2 + \exp(-y_i^e \cdot f(\mathbf{W}, \mathbf{x}_i, t))} - \frac{1}{4}\right| \\ &\leq \left|\frac{1}{4} - \frac{1}{2+2\exp(\gamma^2/2)}\right| = \Theta(\gamma^2).\end{aligned}$$

Similarly, we have:

$$\begin{aligned}\left|\mathbf{H}_{e,e'}^2(t)\right| &= \frac{2}{mn_e n_{e'}}\left|\sum_{i=1}^{n_e} \ell_i''(t)\hat{y}_i^e(t) \cdot jy_i^e \mathbf{v}_i^{e\top} \sum_{i=1}^{n_{e'}} \ell_i''(t)\hat{y}_i^e(t) \cdot jy_i^e \mathbf{v}_i^e\right| \leq C\frac{2\gamma^2}{m}, \\ \left|\mathbf{H}_{e,e'}^3(t)\right| &= \frac{2}{mn_e n_{e'}}\left|\sum_{i=1}^{n_e} \ell_i''(t)\hat{y}_i^e(t) \cdot jy_i^e \mathbf{v}_i^{e\top} \sum_{i=1}^{n_{e'}} \ell_i'(t) \cdot j\mathbf{v}_i^e\right| \leq C\frac{2\gamma}{m}, \\ \left|\mathbf{H}_{e,e'}^4(t)\right| &= \frac{2}{mn_e n_{e'}}\left|+\sum_{i=1}^{n_e} \ell_i'(t) \cdot j\mathbf{v}_i^{e\top} \sum_{i=1}^{n_{e'}} \ell_i''(t)\hat{y}_i^e(t) \cdot j\mathbf{v}_i^e\right| \leq C\frac{2\gamma}{m}, \\ \left|\mathbf{H}_{e,e'}^5(t)\right| &= \frac{2}{mn_e n_{e'}}\left|\sum_{i=1}^{n_e} \ell_i'(t) \cdot jy_i^e \boldsymbol{\xi}_i^{e\top} \sum_{i=1}^{n_{e'}} \ell_i'(t) \cdot jy_i^e \boldsymbol{\xi}_i^e\right| \leq C\frac{2}{m}\sigma_q^2 d, \\ \left|\mathbf{H}_{e,e'}^6(t)\right| &= \frac{2}{mn_e n_{e'}}\left|\sum_{i=1}^{n_e} \ell_i''(t) y_i^e(t) \cdot j\boldsymbol{\xi}_i^{e\top} \sum_{i=1}^{n_{e'}} \ell_i''(t) \cdot j\boldsymbol{\xi}_i^e\right| \leq C\frac{2}{m}\sigma_q^2 d\gamma^2, \\ \left|\mathbf{H}_{e,e'}^7(t)\right| &= \frac{2}{mn_e n_{e'}}\left|\sum_{i=1}^{n_e} \ell_i''(t) \cdot j\boldsymbol{\xi}_i^{e\top} \sum_{i=1}^{n_{e'}} \ell_i'(t) \cdot jy_i^e \boldsymbol{\xi}_i^e\right| \leq C\frac{2}{m}\sigma_q^2 d\gamma, \\ \left|\mathbf{H}_{e,e'}^8(t)\right| &= \frac{2}{mn_e n_{e'}}\left|\sum_{i=1}^{n_e} y_i^e \ell_i'(t) \cdot j\boldsymbol{\xi}_i^{e\top} \sum_{i=1}^{n_{e'}} \ell_i''(t) \cdot j\boldsymbol{\xi}_i^e\right| \leq C\frac{2}{m}\sigma_q^2 d\gamma.\end{aligned}$$

To summarize, we have that,

$$\left|\mathbf{H}_{e,e'}(t) - \mathbf{H}_{e,e'}^\infty\right| \leq C_1\frac{2}{m}\gamma + C_2\frac{2}{m}\sigma_q^2\gamma.$$

Furthermore, we have that

$$|\mathbf{g}_e(t)| \leq C\frac{2}{m}\max\{\sigma_q^2 d, \gamma\}.$$

Finally, we have the dynamics for $\|\mathbf{c}(t)\|_2^2$

$$\frac{d\|\mathbf{c}(t)\|_2^2}{dt} = -2\lambda\mathbf{c}^\top(t)\mathbf{H}(t)\mathbf{c}(t) - \mathbf{c}(t)\mathbf{g}(t) \leq -\lambda_0\lambda\|\mathbf{c}(t)\|_2^2,$$

which requires that:

$$C_1\frac{2}{m}\gamma < \lambda_0; \quad \lambda > \frac{1}{\|\mathbf{c}(0)\|_2}.$$

According to the gradient descent for IRMV1 objective function, the evolution of coefficients can be expressed as:

$$\begin{aligned}\gamma_{j,r,1}(t+1) &= \gamma_{j,r,1}(t) - \frac{\eta}{m} \cdot \sum_{e\in\mathcal{E}_{\text{tr}}}(1+2\lambda C_{\text{IRMv1}}^e(t))\frac{1}{n_e}\sum_{i=1}^{n_e}\ell_i'(t)\text{Rad}(\alpha)_i - \frac{\eta\lambda}{m} \cdot \sum_{e\in\mathcal{E}_{\text{tr}}} 2C_{\text{IRMv1}}^e \frac{1}{n_e}\sum_{i=1}^{n_e}\ell_i''\hat{y}_i^e \cdot y_i^e \text{Rad}(\alpha)_i, \\ \gamma_{j,r,2}(t+1) &= \gamma_{j,r,2}(t) - \frac{\eta}{m} \cdot \sum_{e\in\mathcal{E}_{\text{tr}}}(1+2\lambda C_{\text{IRMv1}}^e(t))\frac{1}{n_e}\sum_{i=1}^{n_e}\ell_i'(t)\text{Rad}(\beta_e) - \frac{\eta\lambda}{m} \cdot \sum_{e\in\mathcal{E}_{\text{tr}}} 2C_{\text{IRMv1}}^e \frac{1}{n_e}\sum_{i=1}^{n_e}\ell_i''\hat{y}_i^e \cdot y_i^e \text{Rad}(\beta_e)_i.\end{aligned}$$

Then we have,

$$
\begin{aligned}
|\gamma_{j,r,1}(t+1)| &\leq |\gamma_{j,r,1}(t)| + \left| \frac{\eta}{m} \cdot \sum_{e \in \mathcal{E}_{\text{tr}}} (1 + 2\lambda C^e_{\text{IRMv1}}(t)) \frac{1}{n_e} \sum_{i=1}^{n_e} \ell_i'(t) \text{Rad}(\alpha)_i \right| \\
&+ \left| \frac{\eta\lambda}{m} \cdot \sum_{e \in \mathcal{E}_{\text{tr}}} 2C^e_{\text{IRMv1}} \frac{1}{n_e} \sum_{i=1}^{n_e} \ell_i'' \hat{y}_i^e \cdot y_i^e \text{Rad}(\alpha)_i \right| \\
&\leq |\gamma_{j,r,1}(t)| + C\frac{\eta\lambda}{m} \|\mathbf{c}(t)\|_2.
\end{aligned}
$$

Similarly, we have,

$$
|\gamma_{j,r,2}(t+1)| \leq |\gamma_{j,r,2}(t)| + C\frac{\eta\lambda}{m} \|\mathbf{c}(t)\|_2.
$$

Taking the convergence time $T = \Omega\left(\frac{\log(\sigma_0/\epsilon)}{\eta\lambda\lambda_0}\right)$ we have that:

$$
\|\mathbf{c}(T)\|_2 \leq \epsilon.
$$

Besides, at the time step T, the feature learning satisfies that:

$$
\gamma_{j,r,1}(T) \leq C\frac{\eta\lambda T}{m} \|\mathbf{c}(0)\|_2; \quad \gamma_{j,r,1}(T) \leq C\frac{\eta\lambda T}{m} \|\mathbf{c}(0)\|_2.
$$

To make sure that $\gamma_{j,r1,1}(T) = o_d(1)$ and $\gamma_{j,r1,2}(T) = o_d(1)$, we need the following condition:

$$
C\frac{\eta\lambda T}{m} \|\mathbf{c}(0)\|_2 \leq d^{-\frac{1}{2}},
$$

which results in $\sigma_0 \leq \frac{m\lambda_0}{d^{-1/2}\log(1/\epsilon)}$. Furthermore, we have that:

$$
|f(\mathbf{x}_i, T)| \leq \gamma_{j,r,1}(T) + \gamma_{j,r,2}(T) \leq \gamma.
$$

Combined with the condition that $C\frac{2\gamma}{m} < \lambda_0$, we obtain that:

$$
\sigma_0 \leq \frac{\lambda_0^2 m^2}{\log(1/\epsilon)}.
$$

□

C.4 Poor invariant feature + IRMv1

Corollary C.7 (Restatement of Corollary 3.4)**.** *Consider training the CNN model with the same data as Theorem 3.1, suppose that $\gamma_{j,r,1}(t_1) = o(1)$ and $\gamma_{j,r,2}(t_1) = \Theta(1)$ at the end of ERM pre-train t_1 and $\mathcal{E}_{tr} = \{(0.25, 0.1), (0.25, 0.2)\}$. Then, in the limit of $n \to \infty$, we have*

$$
\gamma_{j,r,1}(t_1 + 1) < \gamma_{j,r,1}(t_1).
$$

Proof of Corollary C.7. Recall that the feature learning update rule:

$$
\gamma_{j,r,1}(t+1) = \gamma_{j,r,1}(t) - \frac{\eta}{m} \cdot \sum_{e \in \mathcal{E}_{\text{tr}}} (1 + 2\lambda C^e_{\text{IRMv1}}(t)) \frac{1}{n_e} \sum_{i=1}^{n_e} \ell_i'(t) \text{Rad}(\alpha)_i - \frac{\eta\lambda}{m} \cdot \sum_{e \in \mathcal{E}_{\text{tr}}} 2C^e_{\text{IRMv1}} \frac{1}{n_e} \sum_{i=1}^{n_e} \ell_i'' \hat{y}_i^e \cdot y_i^e \text{Rad}(\alpha)_i,
$$

$$
\gamma_{j,r,2}(t+1) = \gamma_{j,r,2}(t) - \frac{\eta}{m} \cdot \sum_{e \in \mathcal{E}_{\text{tr}}} (1 + 2\lambda C^e_{\text{IRMv1}}(t)) \frac{1}{n_e} \sum_{i=1}^{n_e} \ell_i'(t) \text{Rad}(\beta_e) - \frac{\eta\lambda}{m} \cdot \sum_{e \in \mathcal{E}_{\text{tr}}} 2C^e_{\text{IRMv1}} \frac{1}{n_e} \sum_{i=1}^{n_e} \ell_i'' \hat{y}_i^e \cdot y_i^e \text{Rad}(\beta_e)_i,
$$

Taking the value of $\gamma_{j,r,1}(t_1)$, $\gamma_{j,r,2}(t_1)$ and, we can conclude that:

$$\begin{aligned}
\lim_{n\to\infty} A_1^1(t_1) &= -1/(1+e^{\gamma_2})(1-\alpha)(1-\beta_1) - 1/(1+e^{-\gamma_2})(1-\alpha)\beta_1 + 1/(1+e^{\gamma_2})\alpha(1-\beta_1) + 1/(1+e^{-\gamma_2})\alpha\beta_1 \\
&= 1/(1+e^{\gamma_2})(2\alpha-1)(1-\beta_1) + 1/(1+e^{-\gamma_2})(2\alpha-1)(\beta_1) \\
&= (2\alpha-1)[1/(1+e^{\gamma_2})(1-\beta_2) + 1/(1+e^{-\gamma_2})\beta_1)] \\
\lim_{n\to\infty} A_1^2(t_1) &= 1/(1+e^{\gamma_2})(2\alpha-1)(1-\beta_2) + 1/(1+e^{-\gamma_2})(2\alpha-1)(\beta_2) \\
&= (2\alpha-1)[1/(1+e^{\gamma_2})(1-\beta_2) + 1/(1+e^{-\gamma_2})\beta_2)] \\
\lim_{n\to\infty} B_1^1(t_1) &= -1/(1+e^{\gamma_2})(1-\alpha)(1-\beta_1) + 1/(1+e^{-\gamma_2})(1-\alpha)\beta_1 - 1/(1+e^{\gamma_2})\alpha(1-\beta_1) + 1/(1+e^{-\gamma_2})\alpha\beta_1 \\
&= -1/(1+e^{\gamma_2})(1-\beta_1) + 1/(1+e^{-\gamma_2})\beta_1 \\
\lim_{n\to\infty} B_1^2(t_1) &= -1/(1+e^{\gamma_2})(1-\alpha)(1-\beta_2) + 1/(1+e^{-\gamma_2})(1-\alpha)\beta_2 - 1/(1+e^{\gamma_2})\alpha(1-\beta_2) + 1/(1+e^{-\gamma_2})\alpha\beta_2 \\
&= -1/(1+e^{\gamma_2})(1-\beta_2) + 1/(1+e^{-\gamma_2})\beta_2
\end{aligned}$$

On the other hand,

$$\begin{aligned}
\lim_{n\to\infty} A_2^1(t_1) &= e^{\gamma_2}/(1+e^{\gamma_2})^2(1-\alpha)(1-\beta_1)(\gamma_2) + e^{-\gamma_2}/(1+e^{-\gamma_2})^2(1-\alpha)\beta_1(-\gamma_2) \\
&\quad + e^{+\gamma_2}/(1+e^{\gamma_2})^2\alpha(1-\beta_1)(-\gamma_2) + e^{-\gamma_2}/(1+e^{-\gamma_2})^2\alpha\beta_1(\gamma_2) \\
&= e^{\gamma_2}/(1+e^{\gamma_2})^2(1-2\alpha)(1-\beta_1) + e^{-\gamma_2}/(1+e^{-\gamma_2})^2(2\alpha-1)\beta_1\gamma_2 \\
\lim_{n\to\infty} A_2^2(t_1) &= e^{\gamma_2}/(1+e^{\gamma_2})^2(1-\alpha)(1-\beta_2)(\gamma_2) + e^{-\gamma_2}/(1+e^{-\gamma_2})^2(1-\alpha)\beta_2(-\gamma_2) \\
&\quad + e^{\gamma_2}/(1+e^{\gamma_2})^2\alpha(1-\beta_2)(-\gamma_2) + e^{-\gamma_2}/(1+e^{-\gamma_2})^2\alpha\beta_2(\gamma_2) \\
&= e^{\gamma_2}/(1+e^{\gamma_2})^2(1-2\alpha)(1-\beta_2) + e^{-\gamma_2}/(1+e^{-\gamma_2})^2(2\alpha-1)\beta_2\gamma_2 \\
\lim_{n\to\infty} B_2^1(t_1) &= e^{\gamma_2}/(1+e^{\gamma_2})^2(1-\alpha)(1-\beta_1)(\gamma_2) + e^{-\gamma_2}/(1+e^{-\gamma_2})^2(1-\alpha)\beta_1(\gamma_2) \\
&\quad + e^{\gamma_2}/(1+e^{\gamma_2})^2\alpha(1-\beta_1)(\gamma_2) + e^{-\gamma_2}/(1+e^{-\gamma_2})^2\alpha\beta_1(\gamma_2), \\
\lim_{n\to\infty} B_2^2(t_1) &= e^{\gamma_2}/(1+e^{\gamma_2})^2(1-\alpha)(1-\beta_2)(\gamma_2) + e^{-\gamma_2}/(1+e^{-\gamma_2})^2(1-\alpha)\beta_2(\gamma_2) \\
&\quad + e^{\gamma_2}/(1+e^{\gamma_2})^2\alpha(1-\beta_2)(\gamma_2) + e^{-\gamma_2}/(1+e^{-\gamma_2})^2\alpha\beta_2(\gamma_2).
\end{aligned}$$

Finally, taking the value of environment of $(\alpha, \beta_1, \beta_2) = (0.25, 0.1, 0.2)$, we complete the proof:

$$\gamma_{j,r,1}(t_1+1) < \gamma_{j,r,1}(t_1).$$

□

D Feature Augmentated Training

D.1 Rich Features for OOD Generalization

The results in Sec. 3 imply that the model is expected to learn all potentially useful features during the pre-training in order to achieve the optimal OOD generalization performance. Otherwise, the OOD training is less likely to exacerbate the poorly learned features. It also explains the success of learning diverse and rich features by weight averaging (Rame et al., 2022) and rich feature construction (or Bonsai) (Zhang et al., 2022), and adopting PCA to learn all the potentially useful features (Ye et al., 2022).

Despite the empirical success, however, the learning of new features in both Bonsai and weight averaging is uncontrolled, which may discard previously learned useful features or fail to explore all the desired features. Besides, they also need multiple initializations and training of the whole networks with different random seeds to encourage the diversity of feature learning, which in turn requires much more computational overhead to reach convergence.

D.2 The FAT Algorithm

To overcome the limitations of previous rich feature learning algorithms, we propose **F**eature **A**ugmented **T**raining (FAT), that adopts an iterative data-centric strategy to enforce the model to learn all useful features directly.

Algorithm 1 FAT: **F**eature **A**ugmented **T**raining

Input: Training data $\mathcal{D}_{\text{tr}}$; the maximum augmentation rounds K; predictor $f := w \circ \varphi$; length of inner training epochs e; termination threshold p;
Initialize groups $G^a \leftarrow \mathcal{D}_{\text{tr}}, G^r \leftarrow \{\}$;
for $k \in [1, \ldots, K]$ **do**
  Randomly initialize w_k;
  for $j \in [1, \ldots, e]$ **do**
    Obtain ℓ_{FAT} with G via Eq. 24;
    Update w_k, φ with ℓ_{FAT};
  end for
  `// Early Stop if` $f_k = w_k \circ \varphi$ `fails to find new features.`
  if Training accuracy of f_k is smaller than p **then**
    Set $K = k - 1$ and terminate the loop;
  end if
  Split $\mathcal{D}_{\text{tr}}$ into groups $\mathcal{D}_k^a, \mathcal{D}_k^r$ according to f_k;
  Update groups $G^a \leftarrow G^a \cup \{\mathcal{D}_k^a\}, G^r \leftarrow G^r \cup \{\mathcal{D}_k^r\}$;
end for
Synthesize the final classifier $w \leftarrow \frac{1}{K}\sum_{i=1}^{K} w_i$;
return $f = w \circ \varphi$;

Intuitively, the potentially useful features presented in the training data are features that have non-trivial correlations with labels, or using the respective feature to predict the labels is able to achieve a *non-trivial training performance*. Moreover, the invariance principle assumes that the training data comes from different environments (Arjovsky et al., 2019), which implies that each set of features can only dominate the correlations with labels in a *subset* of data. Therefore, it is possible to differentiate the distinct sets of useful features entangled in the training data into different subsets, where simple ERM can effectively learn the dominant features presented in the data as shown in Theorem 3.1.

The intuition naturally motivates an iterative rich feature learning algorithm, i.e., FAT, that differentiates the subsets and explores new features in multiple rounds. We list FAT in Algorithm 1, where we are given a randomly initialized or pre-trained model $f = w \circ \varphi$ that consists a featurizer φ and a classifier w. In round k, FAT first identifies the subset that contains the already learned features by collecting the data points where f yields the correct prediction, denoted as G_k^r, and the subset that contains the other samples as G_k^a.

Given a grouped datasets $G = \{G^r, G^a\}$ with $2k-1$ groups, where $G^a = \{\mathcal{D}_i^a\}_{i=0}^{k-1}$ is the grouped sets for new feature augmentation, and $G^r = \{\mathcal{D}_i^r\}_{i=1}^{k-1}$ is the grouped sets for already learned feature retention (notice that $\mathcal{D}_0^r$ is the empty set), FAT performs distributionally robust optimization (DRO) (Namkoong & Duchi, 2016; Zhang et al., 2022) on G^a to explore new features that have not been learned in previous rounds. Meanwhile, FAT also needs to *retain* the already learned features by minimizing the empirical risk at G^r. The FAT objective at round k then is

$$\ell_{\text{FAT}} = \max_{\mathcal{D}_i^a \in G^a} \ell_{\mathcal{D}_i^a}(w_k \circ \varphi) + \lambda \cdot \sum_{\mathcal{D}_i^r \in G^r} \ell_{\mathcal{D}_i^r}(w_i \circ \varphi), \tag{24}$$

where $\ell_{\mathcal{D}_i}(w \circ \varphi)$ refers to the empirical risk of $w \circ \varphi$ evaluated at the subset $\mathcal{D}_i$, and $\{w_i | 1 \leq i \leq k-1\}$ are the historical classifiers trained in round i.

Relations with previous rich feature learning algorithms. Compared with previous feature learning algorithms, FAT *directly* controls the exploration of new features while keeping retaining the already learned features by identifying and grouping subsets that contain distinct features, which also requires less computational overhead as FAT does not need to train the whole network multiple times. Although Bonsai also adopts the DRO to explore new features, the multiple initializations in Bonsai could lead to uncontrolled behaviors such as forgetting that leads to suboptimal performance, as observed in our experiments.

D.3 PRACTICAL IMPLEMENTATIONS OF FAT

When the featurizer is implemented as a deep net that have a massive amount of parameters, back-propagating through Algorithm 1 can allocate too much memory for propagating with $2K-1$ batches of data. It is common for many realistic benchmarks such as Camelyon17 and FMoW in wilds benchmark (Koh et al., 2021) that adopts a DenseNet (Huang et al., 2017) with 121 layers as the featurizer. To relieve the exceeding computational and memory overhead, we propose a lightweight version of FAT, denoted as FAT. Instead of storing all of historical subsets and classifiers, iFAT iteratively use the augmentation and retention sets and historical classifier from only the last round. In contrast, previous rich feature learning algorithm (Zhang et al., 2022; Rame et al., 2022) incurs a high computational and memory overhead as the round grows. For example, in RxRx1, we have to reduce the batch size of Bonsai to allow the proceeding of rounds ≥ 3.

We elaborate the detailed algorithmic description of iFAT in Algorithm 2. As iFAT stores only the latest augmentation and retention subsets, inspecting the training performance for termination check (line 10 of Algorithm 1) may not be suitable. However, one can still inspect the performance in either an OOD validation set that finds suitable intermediate feature representations, or the retention set that checks whether learning new features leads to a severe contradiction of already learned features (FAT should terminate if so).

Algorithm 2 A practical variant of FAT

1: **Input:** Training data $\mathcal{D}_{\text{tr}}$; the maximum augmentation rounds K; predictor $f := w \circ \varphi$; length of inner training epochs e; termination threshold p;
2: Initialize groups $G^a \leftarrow \mathcal{D}_{\text{tr}}, G^r \leftarrow \{\}$;
3: **for** $k \in [1, \ldots, K]$ **do**
4: Randomly initialize w_k;
5: **for** $j \in [1, \ldots, e]$ **do**
6: Obtain ℓ_{FAT} with G via Eq. 24;
7: Update w_k, φ with ℓ_{FAT};
8: **end for**
9: `// Early Stop if` $f_k = w_k \circ \varphi$ `fails to find new features.`
10: **if** Training accuracy of f_k is smaller than p **then**
11: Set $K = k - 1$ and terminate the loop;
12: **end if**
13: **if** $k > 1$ **then**
14: `// Hence it doesnot need to maintain all historical classifiers.`
15: Update $w_k \leftarrow (w_{k-1}, w_k)$;
16: **end if**
17: Split $\mathcal{D}_{\text{tr}}$ into groups $\mathcal{D}_k^a, \mathcal{D}_k^r$ according to f_k;
18: `// Hence it doesnot need to maintain all historical subsets.`
19: Update groups $G^a \leftarrow \{\mathcal{D}_k^a\}, G^r \leftarrow \{\mathcal{D}_k^r\}$;
20: **end for**
21: **return** $f = w \circ \varphi$;

E MORE EMPIRICAL RESULTS

We provide full details of the empirical verifications in Sec. 3.

E.1 COLOREDMNIST EXPERIMENTS

We conducted extensive experiments on COLOREDMNIST and WILDS to verify the effectiveness of FAT in finding a better OOD solution under objective conflicts.

A controlled study. We first conducted a controlled study using COLOREDMNIST dataset (Arjovsky et al., 2019) to examine the feature learning performance of FAT under various conditions.

Table 2: OOD performance on COLOREDMNIST datasets initialized with different representations.

	COLOREDMNIST-025				COLOREDMNIST-01			
	ERM-NF	ERM	BONSAI	FAT	ERM-NF	ERM	BONSAI	FAT
ERM	17.14 (±0.73)	12.40 (±0.32)	11.21 (±0.49)	**17.27** (±2.55)	73.06 (±0.71)	73.75 (±0.49)	70.95 (±0.93)	**76.05** (±1.45)
IRMv1	67.29 (±0.99)	59.81 (±4.46)	70.28 (±0.72)	**70.57** (±0.68)	76.89 (±3.25)	73.84 (±0.56)	76.71 (±4.10)	**82.33** (±1.77)
V-REX	68.62 (±0.73)	65.96 (±1.29)	70.31 (±0.66)	**70.82** (±0.59)	83.52 (±2.52)	81.20 (±3.27)	82.61 (±1.76)	**84.70** (±0.69)
IRMX	67.00 (±1.95)	64.05 (±0.88)	70.46 (±0.42)	**70.78** (±0.61)	81.61 (±1.98)	75.97 (±0.88)	80.28 (±1.62)	**84.34** (±0.97)
IB-IRM	56.09 (±2.04)	59.81 (±4.46)	70.28 (±0.72)	**70.57** (±0.68)	75.81 (±0.63)	73.84 (±0.56)	76.71 (±4.10)	**82.33** (±1.77)
CLOVE	58.67 (±7.69)	65.78 (±0.00)	65.57 (±3.02)	**65.78** (±2.68)	75.66 (±10.6)	74.73 (±0.36)	72.73 (±1.18)	**75.12** (±1.08)
IGA	51.22 (±3.67)	62.43 (±3.06)	**70.17** (±0.89)	67.11 (±3.40)	74.20 (±2.45)	73.74 (±0.48)	74.72 (±3.60)	**83.46** (±2.17)
FISHR	69.38 (±0.39)	67.74 (±0.90)	68.75 (±1.10)	**70.56** (±0.97)	77.29 (±1.61)	82.23 (±1.35)	84.19 (±0.66)	**84.26** (±0.93)
ORACLE	71.97 (±0.34)				86.55 (±0.27)			

We used both the original COLOREDMNIST with $\mathcal{E}_{\text{tr}} = \{(0.25, 0.1), (0.25, 0.2)\}$ (denoted as COLOREDMNIST-025), where spurious features are better correlated with labels, and the modified COLOREDMNIST (denoted as COLOREDMNIST-01) with $\mathcal{E}_{\text{tr}} = \{(0.1, 0.2), (0.1, 0.25)\}$, where invariant features are better correlated with labels. We compared the OOD performance of the features learned by FAT, with that of ERM and the state-of-the-art rich feature learning algorithm Bonsai (Zhang et al., 2022). We select various state-of-the-art OOD objectives including IRMv1 (Arjovsky et al., 2019), VREx (Krueger et al., 2021), IRMX (Chen et al., 2022b), IB-IRM (Ahuja et al., 2021), CLOvE (Wald et al., 2021), IGA (Koyama & Yamaguchi, 2020) and Fishr (Rame et al., 2021) for evaluating the quality of the learned features. The feature representations are frozen once initialized for the OOD training as fine-tuning the featurizer can distort the pre-trained features (Kumar et al., 2022). We also compared FAT with the common training approach that uses unfrozen ERM features, denoted as ERM-NF. For Bonsai, we trained 2 rounds following Zhang et al. (2022), while for FAT the automatic termination stopped at round 2 in COLOREDMNIST-025 and round 3 in COLOREDMNIST-01. For ERM, we pre-trained the model with the same number of overall epochs as FAT in COLOREDMNIST-01, while early stopping at the number of epochs of 1 round in COLOREDMNIST-025 to prevent over-fitting. All methods adopted the same backbone and the same training protocol following previous works (Zhang et al., 2022; Chen et al., 2022b). More details are given in Appendix F.1.

The results are reported in Table 2, which shows that FAT significantly improves the OOD generalization performance of all OOD objectives for all the COLOREDMNIST datasets over ERM. In contrast, although Bonsai boosts the OOD performance for COLOREDMNIST-025, it sometimes leads to a suboptimal performance when invariant correlations are stronger, which could be attributed to the uncontrolled feature learning with multiple initializations.

E.2 WILDS EXPERIMENTS

Table 3: OOD generalization performances on WILDS benchmark.

INIT.	METHOD	CAMELYON17*	CIVILCOMMENTS	FMOW	IWILDCAM*	AMAZON*	RXRX1
		Avg. acc. (%)	Worst acc. (%)	Worst acc. (%)	Macro F1	10-th per. acc. (%)	Avg. acc. (%)
ERM	DFR	95.14 (±1.96)	**77.34** (±0.50)	41.96 (±1.90)	23.15 (±0.24)	48.00 (±0.00)	-
ERM	DFR-s†	-	82.24 (±0.13)	56.17 (±0.62)	52.44 (±0.34)	-	-
ERM	ERM	74.30 (±5.96)	55.53 (±1.78)	33.58 (±1.02)	28.22 (±0.78)	51.11 (±0.63)	30.21 (±0.09)
ERM	GroupDRO	76.09 (±6.46)	69.50 (±0.15)	33.03 (±0.52)	28.51 (±0.58)	52.00 (±0.00)	29.99 (±0.13)
ERM	IRMv1	75.68 (±7.41)	68.84 (±0.95)	33.45 (±1.07)	28.76 (±0.45)	52.00 (±0.00)	30.10 (±0.05)
ERM	V-REx	71.60 (±7.88)	69.03 (±1.08)	33.06 (±0.46)	28.82 (±0.47)	52.44 (±0.63)	29.88 (±0.35)
ERM	IRMX	73.49 (±9.33)	68.91 (±1.19)	33.13 (±0.86)	28.82 (±0.47)	52.00 (±0.00)	30.10 (±0.05)
Bonsai	DFR	95.17 (±0.18)	77.07 (±0.85)	43.26 (±0.82)	21.36 (±0.41)	46.67 (±0.00)	-
Bonsai	DFR-s†	-	81.26 (±1.86)	58.58 (±1.17)	50.85 (±0.18)	-	-
Bonsai	ERM	73.98 (±5.30)	63.34 (±3.49)	31.91 (±0.51)	28.27 (±1.05)	48.58 (±0.56)	24.22 (±0.44)
Bonsai	GroupDRO	72.82 (±5.37)	70.23 (±1.33)	33.12 (±1.20)	27.16 (±1.18)	42.67 (±1.09)	22.95 (±0.46)
Bonsai	IRMv1	73.59 (±6.16)	68.39 (±2.01)	32.51 (±1.23)	27.60 (±1.57)	47.11 (±0.63)	23.35 (±0.43)
Bonsai	V-REx	76.39 (±5.32)	68.67 (±1.29)	33.17 (±1.26)	25.81 (±0.42)	48.00 (±0.00)	23.34 (±0.42)
Bonsai	IRMX	64.77 (±10.1)	69.56 (±0.95)	32.63 (±0.75)	27.62 (±0.66)	46.67 (±0.00)	23.34 (±0.40)
FAT	DFR	**95.28** (±0.19)	**77.34** (±0.59)	**43.54** (±1.26)	**23.54** (±0.52)	**49.33** (±0.00)	-
FAT	DFR-s†	-	79.56 (±0.38)	57.69 (±0.78)	52.31 (±0.38)	-	-
FAT	ERM	77.80 (±2.48)	68.11 (±2.27)	33.13 (±0.78)	28.47 (±0.67)	**52.89** (±0.63)	**30.66** (±0.42)
FAT	GroupDRO	**80.41** (±3.30)	**71.29** (±0.46)	33.55 (±1.67)	28.38 (±1.32)	52.58 (±0.56)	29.99 (±0.11)
FAT	IRMv1	77.97 (±3.09)	70.33 (±1.14)	**34.04** (±0.70)	**29.66** (±1.52)	**52.89** (±0.63)	29.99 (±0.19)
FAT	V-REx	75.12 (±6.55)	70.97 (±1.06)	34.00 (±0.71)	29.48 (±1.94)	**52.89** (±0.63)	30.57 (±0.53)
FAT	IRMX	76.91 (±6.76)	71.18 (±1.10)	33.99 (±0.73)	29.04 (±2.96)	**52.89** (±0.63)	29.92 (±0.16)

*Indicates using "cheating" protocol in DFR. †Indicates spurious feature learning. The lower the better.

We also compared FAT with ERM and Bonsai in the realistic datasets curated by Koh et al. (2021) that contain complicated features and distribution shifts. The learned features were evaluated with several state-of-the-art OOD objectives in WILDS, including GroupDro (Sagawa* et al., 2020),

IRMv1 (Arjovsky et al., 2019), VREx (Krueger et al., 2021) as well as IRMX (Chen et al., 2022b). In addition, we evaluated the learned features with Deep Feature Reweighting (DFR) (Kirichenko et al., 2022; Izmailov et al., 2022). Specifically, when a OOD validation set was available, DFR performed logistic regression on the OOD validation set based on the learned features andIzmailov et al. (2022) found that it achieves impressive OOD performance. We also reported DFR-s by regression with the environment labels (when available) to evaluate the spurious feature learning of different methods. For datasets without a proper OOD validation set, e.g., CAMELYON17, we followed the "cheating" protocol (Rosenfeld et al., 2022) that performs logistic regression based on both a random split from the training and test data. More details are given in Appendix F.2.2. We trained all ERM, Bonsai and FAT the same number of steps, and kept the rounds of Bonsai and FAT the same. The only exception was in RXRX1 where both Bonsai and FAT required more steps to converge. We used the same evaluation protocol as previous works (Koh et al., 2021; Shi et al., 2022; Zhang et al., 2022; Chen et al., 2022b) to ensure a fair comparison. More details are given in Appendix F.2.

The results are presented in Table 3. FAT consistently achieves the best invariant feature learning performance across various challenging realistic datasets. Meanwhile, compared to ERM and Bonsai, FAT also reduces over-fitting to the spurious feature learning led by spurious correlations. Thus, FAT achieves consistent improvements when the learned features are applied to various OOD objectives. In contrast, Bonsai may lead to a suboptimal performance due to the uncontrolled learning behavior.

E.3 THE TERMINATION CHECK IN FAT.

Table 4: Performances in various sets at different FAT rounds.

COLOREDMNIST-025	ROUND-1	ROUND-2	ROUND-3
TRAINING ACC.	85.08± 0.14	71.87± 0.96	84.93± 1.26
RETENTION ACC.	-	88.11± 4.28	43.82± 0.59
OOD ACC.	11.08± 0.30	70.64± 0.62	10.07± 0.26

A key difference between FAT and previous rich feature learning algorithms is that FAT performs controlled feature learning. As elaborated in Sec. D.2, FAT can terminate automatically by inspecting the retention accuracy. To verify, we list the FAT performances in various subsets of COLOREDMNIST-025 at different rounds. As shown in Table 4, after FAT learns sufficiently good features at Round 2, it is not necessary to proceed with Round 3 as it will destroy the already learned features and lead to degenerated retention and OOD performance.

F MORE DETAILS ABOUT THE EXPERIMENTS

In this section, we provide more details and the implementation, evaluation and hyperparameter setups in complementary to the experiments in Appendix E.

F.1 MORE DETAILS ABOUT COLOREDMNIST EXPERIMENTS

Datasets. In the controlled experiments with COLOREDMNIST, we follow the evaluation settings as previous works (Arjovsky et al., 2019; Zhang et al., 2022; Chen et al., 2022b). In addition to the original COLOREDMNIST with $\mathcal{E}_{\text{tr}} = \{(0.25, 0.1), (0.25, 0.2)\}$ (denoted as COLOREDMNIST-025) where spurious features are better correlated with labels, we also incorporate the modified one (denoted as COLOREDMNIST-01) with $\mathcal{E}_{\text{tr}} = \{(0.1, 0.2), (0.1, 0.25)\}$ where invariant features are better correlated with labels, since both cases can happen at real world.

Architecture and optimization. To ensure a fair comparison, we use 4-Layer MLP with a hidden dimension of 256 as the backbone model for all methods, where we take the first 3 layers as the featurizer and the last layer as the classifier, following the common practice (Gulrajani & Lopez-Paz, 2021; Koh et al., 2021). For the optimization of the models, we use the Adam Kingma & Ba (2015) optimizer with a learning rate of $1e-3$ and a weight decay of $1e-3$. We report the mean and standard deviation of the performances of different methods with each configuration of hyperparameters 10 times with the random seeds from 1 to 10.

Implementation of ERM-NF and OOD objectives. For the common pre-training protocol with ERM, our implementation follows the previous works (Zhang et al., 2022). Specifically, we first train the model with $\{0, 50, 100, 150, 200, 250\}$ epochs and then apply the OOD regularization of various objectives with a penalty weight of $\{1e1, 1e2, 1e3, 1e4, 1e5\}$. We adopt the implementations from Zhang et al. (2022) for various OOD objectives, including IRMv1 (Arjovsky et al., 2019),VREx (Krueger et al., 2021),IB-IRM (Ahuja et al., 2021),CLOvE (Wald et al., 2021),IGA (Koyama & Yamaguchi, 2020) and Fishr (Rame et al., 2021) Besides, we also incorporate the state-of-the-art OOD objective proposed by Chen et al. (2022b) that is able to resolve both COLOREDMNIST-025 and COLOREDMNIST-01.

Evaluation of feature learning methods. For the sake of fairness in comparison, by default, we train all feature learning methods by the same number of epochs and rounds (if applicable). For the implementation Bonsai, we strictly follow the recommended setups provided by Zhang et al. (2022), [6] where we train the model with Bonsai by 2 rounds with 50 epochs for round 1, 500 epochs for round 2, and 500 epochs for the synthesize round in COLOREDMNIST-025. While in COLOREDMNIST-01, round 1 contains 150 epochs, round 2 contains 400 epochs and the synthesize round contains 500 epochs. For the implementation of FAT, we train the model with 2 rounds of FAT in COLOREDMNIST-025, and 3 rounds of FAT in COLOREDMNIST-01, where each round contains 150 epochs. While for the retain penalty, we find using a fixed number of 0.01 already achieved sufficiently good performance. ERM only contains 1 round, for which we train the model with 150 epochs in COLOREDMNIST-025 as we empirically find more epochs will incur severe performance degeneration in COLOREDMNIST-025. While in COLOREDMNIST-01, we train the model with ERM by 500 epochs to match up the overall training epochs of FAT and Bonsai. We provide a detailed distribution of the number of epochs in each round in Table 5. It can be found

Table 5: Number of epochs in each round of various feature learning algorithms.

CMNIST-025	ROUND-1	ROUND-2	ROUND-3	SYN. ROUND	CMNIST-01	ROUND-1	ROUND-2	ROUND-3	SYN. ROUND
ERM	150	-	-	-	ERM	500	-	-	-
BONSAI	50	150	-	500	BONSAI	150	400	-	500
FAT	150	150	-	-	FAT	150	150	150	-

that, although Bonsai costs $2-3$ times of training epochs more than ERM and FAT, Bonsai does not necessarily find better feature representations for OOD training, as demonstrated in Table. 1. In contrast, FAT significantly and consistently learns richer features given both COLOREDMNIST-025 and COLOREDMNIST-01 than ERM, which shows the superiority of FAT.

F.2 MORE DETAILS ABOUT WILDS EXPERIMENTS

In this section, we provide more details about the WILDS datasets used in the experiments as well as the evaluation setups.

F.2.1 DATASET DESCRIPTION.

To evaluate the feature learning performance given data from realistic scenarios, we select 6 challenging datasets from WILDS (Koh et al., 2021) benchmark. The datasets contain various realistic distribution shifts, ranging from domain distribution shifts, subpopulation shifts and the their mixed. A summary of the basic information and statistics of the selected WILDS datasets can be found in Table. 6, Table. 7, respectively. In the following, we will give a brief introduction to each of the datasets. More details can be found in the WILDS paper (Koh et al., 2021).

Amazon. We follow the WILDS splits and data processing pipeline for the Amazon dataset (Ni et al., 2019). It provides 1.4 million comments collected from $7,676$ Amazon customers. The task is to predict the score (1-5 stars) for each review. The domains d are defined according to the reviewer/customer who wrote the product reviews. The evaluation metric used for the task is 10th percentile of per-user accuracies in the OOD test sets, and the backbone model is a DistilBert (Sanh et al., 2019), following the WILDS protocol (Koh et al., 2021).

[6] https://github.com/TjuJianyu/RFC

Table 6: A summary of datasets information from WILDS.

Dataset	Data (x)	Class information	Domains	Metric	Architecture
AMAZON	Product reviews	Star ratings (5 classes)	7,676 reviewers	10-eth percentile acc.	DistillBERT
CAMELYON17	Tissue slides	Tumor (2 classes)	5 hospitals	Avg. acc.	DenseNet-121
CIVILCOMMENTS	Online comments	Toxicity (2 classes)	8 demographic groups	Wr. group acc.	DistillBERT
FMOW	Satellite images	Land use (62 classes)	16 years x 5 regions	Wr. group acc.	DenseNet-121
IWILDCAM	Photos	Animal species (186 classes)	324 locations	Macro F1	ResNet-50
RXRX1	Cell images	Genetic treatments (1,139 classes)	51 experimental batches	Avg. acc	ResNet-50

Table 7: A summary of datasets statistics from WILDS.

Dataset	# Examples			# Domains		
	train	val	test	train	val	test
AMAZON	1,000,124	100,050	100,050	5,008	1,334	1,334
CAMELYON17	302,436	34,904	85,054	3	1	1
CIVILCOMMENTS	269,038	45,180	133,782	-	-	-
FMOW	76,863	19,915	22,108	11	3	2
IWILDCAM	129,809	14,961	42,791	243	32	48
RXRX1	40,612	9,854	34,432	33	4	14

Camelyon17. We follow the WILDS splits and data processing pipeline for the Camelyon17 dataset (Bándi et al., 2019). It provides $450,000$ lymph-node scans from 5 hospitals. The task in Camelyon17 is to take the input of 96×96 medical images to predict whether there exists a tumor tissue in the image. The domains d refers to the index of the hospital where the image was taken. The training data are sampled from the first 3 hospitals where the OOD validation and test data are sampled from the 4-th and 5-th hospital, respectively. We will use the average accuracy as the evaluation metric and a DenseNet-121 (Huang et al., 2017) as the backbone for the featurizer.

CivilComments. We follow the WILDS splits and data processing pipeline for the CivilComments dataset (Borkan et al., 2019). It provides $450,000$ comments collected from online articles. The task is to classify whether an online comment text is toxic or non-toxic. The domains d are defined according to the demographic features, including male, female, LGBTQ, Christian, Muslim, other religions, Black, and White. CivilComments is used to study the subpopulation shifts, here we will use the worst group/domain accuracy as the evaluation metric. As for the backbone of the featurizer, we will use a DistillBert (Sanh et al., 2019) following WILDS (Koh et al., 2021).

FMoW. We follow the WILDS splits and data processing pipeline for the FMoW dataset (Christie et al., 2018). It provides satellite images from 16 years and 5 regions. The task in FMoW is to classify the images into 62 classes of building or land use categories. The domain is split according to the year that the satellite image was collected, as well as the regions in the image which could be Africa, America, Asia, Europe or Oceania. Distribution shifts could happen across different years and regions. The training data contains data collected before 2013, while the validation data contains images collected within 2013 to 2015, and the test data contains images collected after 2015. The evaluation metric for FMoW is the worst region accuracy and the backbone model for the featurizer is a DenseNet-121 (Huang et al., 2017).

iWildCam. We follow the WILDS splits and data processing pipeline for the iWildCam dataset (Beery et al., 2020). It is consist of $203,029$ heat or motion-activated photos of animal specifies from 323 different camera traps across different countries around the world. The task of iWildCam is to classify the corresponding animal specifies in the photos. The domains is split according to the locations of the camera traps which could introduce the distribution shifts. We will use the Macro F1 as the evaluation metric and a ResNet-50 (He et al., 2016) as the backbone for the featurizer.

RxRx1. We follow the WILDS splits and data processing pipeline for the RxRx1 dataset (Taylor et al., 2019). The input is an image of cells taken by fluorescent microscopy. The cells can be genetically perturbed by siRNA and the task of RxRx1 is to predict the class of the corresponding siRNA that have treated the cells. There exists $1,139$ genetic treatments and the domain shifts are introduced by the experimental batches. We will use the average accuracy of the OOD experimental batches as the evaluation metric and a ResNet-50 (He et al., 2016) as the backbone for the featurizer.

F.2.2 TRAINING AND EVALUATION DETAILS.

We follow previous works to implement and evaluate different methods used in our experiments (Koh et al., 2021). The information of the referred paper and code is listed as in Table. 8.

Table 8: The information of the referred paper and code.

Paper	Commit	Code
WILDS (Koh et al., 2021)	v2.0.0	https://wilds.stanford.edu/
Fish (Shi et al., 2022)	333efa24572d99da0a4107ab9cc4af93a915d2a9	https://github.com/YugeTen/fish
Bonsai (Zhang et al., 2022)	33b9ecad0ce8b3462793a2da7a9348d053c06ce0	https://github.com/TjuJianyu/RFC
DFR (Kirichenko et al., 2022; Izmailov et al., 2022)	6d098440c697a1175de6a24d7a46ddf91786804c	https://github.com/izmailovpavel/spurious_feature_learning

The general hyperparemter setting inherit from the referred codes and papers, and are as listed in Table 9. We use the same backbone models to implement the featurizer (He et al., 2016; Huang et al., 2017; Sanh et al., 2019). By default, we repeat the experiments by 3 runs with the random seeds of $0, 1, 2$. While for Camelyon17, we follow the official guide to repeat 10 times with the random seeds from 0 to 9.

Table 9: General hyperparameter settings for the experiments on WILDS.

Dataset	AMAZON	CAMELYON17	CIVILCOMMENTS	FMOW	IWILDCAM	RXRX1
Num. of seeds	3	10	3	3	3	3
Learning rate	2e-6	1e-4	1e-5	1e-4	1e-4	1e-3
Weight decay	0	0	0.01	0	0	1e-5
Scheduler	n/a	n/a	n/a	n/a	n/a	Cosine Warmup
Batch size	64	32	16	32	16	72
Architecture	DistilBert	DenseNet121	DistilBert	DenseNet121	ResNet50	ResNet50
Optimizer	Adam	SGD	Adam	Adam	Adam	Adam
Domains in minibatch	5	3	5	5	10	10
Group by	Countries	Hospitals	Demographics× toxicity	Times × regions	Trap locations	Experimental batches
Training epochs	200	10	5	12	9	90

OOD objective implementations. We choose 4 representative OOD objectives to evaluate the quality of learned features, including GroupDRO (Sagawa* et al., 2020), IRMv1 (Arjovsky et al., 2019), VREx (Krueger et al., 2021) and IRMX (Chen et al., 2022b). We implement the OOD objectives based on the code provided by Shi et al. (2022). For each OOD objective, by default, we follow the WILDS practice to sweep the penalty weights from the range of $\{1e-2, 1e-1, 1, 1e1, 1e2\}$, and perform the model and hyperparameter selection via the performance in the provided OOD validation set of each dataset. Due to the overwhelming computational overhead required by large datasets and resource constraints, we tune the penalty weight in iWildCam according to the performance with seed 0, which we empirically find yields similar results as full seed tunning. Besides in Amazon, we adopt the penalty weights tuned from CivilComments since the two datasets share a relatively high similarity, which we empirically find yields similar results as full seed tunning, too. On the other hand, it raises more challenges for feature learning algorithms in iWildCam and Amazon.

Deep Feature Reweighting (DFR) implementations. For the implementation of DFR (Kirichenko et al., 2022; Izmailov et al., 2022), we use the code provided in Izmailov et al. (2022). By default, DFR considers the OOD validation as an unbiased dataset and adopts the OOD validation set to learn a new classifier based on the frozen features from the pre-trained featurizer. We follow the same implementation and evaluation protocol when evaluating feature learning quality in FMoW and CivilComments. However, since Camelyon17 does not have the desired OOD validation set, we follow the "cheating" protocol as in Rosenfeld et al. (2022) to perform the logistic regression based the train and test sets. Note that when "cheating", the model is not able to access the whole test sets. Instead, the logistic regression is conducted on a random split of the concatenated train and test data. Moreover, for Amazon and iWildCam, we find the original implementation fails to converge possibly due to the complexity of the task, and the relatively poor feature learning quality. Hence we implement a new logistic regression based on PyTorch (Paszke et al., 2019) optimized with SGD, and perform DFR using "cheating" protocol based on the OOD validation set and test set. Besides, we find neither the two aforementioned implementations or dataset choices can lead to DFR convergence in RxRx, which we will leave for future investigations.

Feature learning algorithm implementations. We implement all the feature learning methods based on the Fish code framework. For the fairness of comparison, we set all the methods to train the same number of steps or rounds (if applicable) in WILDS datasets. The only exception is in RxRx1, where both Bonsai and FAT require more steps to converge, since the initialized featurizer has a relatively large distance from the desired featurizer in the task. We did not train the model for much too long epochs as Izmailov et al. (2022) find that it only requires $2-5$ epochs for deep nets to learn high-quality invariant features. The final model is selected based on the OOD validation accuracy during the training. Besides, we tune the retain penalty in FAT by searching over $\{1e-2, 1e-1, 0.5, 1, 2, 10\}$, and finalize the retain penalty according to the OOD validation performance. We list the detailed training steps and rounds setups, as well as the used retain penalty in FAT in Table 10.

Table 10: Hyperparameter setups of feature learning algorithms for the experiments on WILDS.

Dataset	AMAZON	CAMELYON17	CIVILCOMMENTS	FMOW	IWILDCAM	RXRX1
Overall steps	31,000	10,000	50,445	9,600	48,000	20,000
Approx. epochs	4	10,000	3	4	10	10
Num. of rounds	3	2	3	2	2	10
Steps per round	10,334	5,000	16,815	4,800	10	10
FAT Retain penalty	2.0	1e-2	1e-2	1.0	0.5	10

For ERM, we train the model simply by the overall number of steps, except for RxRx1 where we train the model by $15,000$ steps following previous setups (Shi et al., 2022). Bonsai and FAT directly adopt the setting listed in the Table 10. Besides, Bonsai will adopt one additional round for synthesizing the pre-trained models from different rounds. Although Zhang et al. (2022) requires Bonsai to train the two rounds for synthesizing the learned features, we empirically find additional training steps in synthesizing will incur overfitting and worse performance. Moreover, as Bonsai requires propagating $2K-1$ batches of the data that may exceed the memory limits, we use a smaller batch size when training Bonsai in iWildCam (8) and RxRx1 (56).

F.3 SOFTWARE AND HARDWARE

We implement our methods with PyTorch (Paszke et al., 2019). For the software and hardware configurations, we ensure the consistent environments for each datasets. We run all the experiments on on Linux servers with NVIDIA V100 graphics cards with CUDA 10.2.