



(a) Worst-group accuracy when progressively removing the smallest singular values and flattening the remaining spectrum. Flattening consistently improves performance and stabilizes the effect of truncation.

(b) Worst-group accuracy when progressively removing the smallest singular values without flattening. While some gains occur, the effect is unstable and inconsistent across truncation levels.

Figure 3: Effect of spectral manipulations on SimCLR-trained feature representations for SpurCIFAR-10. We evaluate trained features with a linear classifier where singular values are incrementally truncated (from 0 to 512) either with (3a) or without (3b) flattening the remaining spectrum. Flattening alone (even with no truncation) significantly improves worst-group accuracy—from 30% to 40%—highlighting the importance of spectral balance for robust representation learning. Shaded bands indicate worst- and best-group accuracies.

## A Ablation Study on Removing Small Singular Values Without Flattening the Spectrum

To test our hypothesis, we manipulate the spectrum of the feature matrix learned by SimCLR on SpurCIFAR-10. Specifically, we compare two interventions: (1) progressively truncating the smallest singular values while flattening the remaining spectrum (Figure 3a; the same as Figure 2b in the main paper), and (2) progressively truncating the smallest singular values without flattening the remaining spectrum (Figure 3b). We then evaluate the quality of the resulting representations using a linear classifier.

While both approaches yield some gains in worst-group accuracy as more low-variance directions are removed, the second approach (truncation without flattening) exhibits instability (see Figure 3b); it is unclear which singular values should be removed to consistently improve performance. In contrast, Figure 3a shows that truncation combined with flattening leads to more robust and consistent improvements in worst-group accuracy. Notably, even without any truncation, simply flattening the full spectrum significantly boosts worst-group performance, from approximately 30% to 40%.

Taken together, these findings suggest that low-rank directions provide limited benefit for generalization. As illustrated in Figures 3, a balanced spectrum plays a crucial role in enabling robust representations, reinforcing the importance of spectral regularization in the presence of spurious correlations.

## B Proof of Corollary 3

**Corollary 3** (Linear classifier generalization bound). *Fix a failure probability  $\delta \in (0, 1)$ . Let  $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$  be drawn i.i.d. from a  $(\lambda_{\min}(FF^\top), \delta, n)$ -non-degenerate distribution over inputs and binary labels  $y_i \in \{\pm 1\}$ . Let  $f(\mathbf{x}) \in \mathbb{R}^d$  be a fixed representation and define the feature matrix  $F \in \mathbb{R}^{n \times d}$  with rows  $f(\mathbf{x}_i)^\top$ . Consider a linear predictor  $g_{\mathbf{w}}(\mathbf{x}) = \langle f(\mathbf{x}), \mathbf{w} \rangle$ , trained using gradient descent. Then, with probability at least  $1 - \delta$ , the population loss  $L_{\mathcal{D}}(g_{\mathbf{w}^{(k)}}) := \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}} [\ell(g_{\mathbf{w}}(\mathbf{x}), y)]$  satisfies*

$$L_{\mathcal{D}}(g_{\mathbf{w}^{(k)}}) \leq \tilde{O} \left( \sqrt{\frac{\mathbf{y}^\top (FF^\top)^{-1} \mathbf{y}}{n}} \right),$$

where  $\mathbf{y} = (y_1, \dots, y_n)^\top$ , and  $\tilde{O}$  hides logarithmic factors and dependence on  $\delta$ .

806 *Proof.* Without loss of generality, assume  $F$  is appropriately normalized, such that  $\lambda_{\max}(FF^T) \leq 1$ .

807 Consider gradient updates of the form

$$\mathbf{w}(k+1) - \mathbf{w}(k) = -\eta \frac{d\Phi}{d\mathbf{w}} = -\eta F^T(\mathbf{g}(k) - \mathbf{y})$$

808 with  $\eta = \mathcal{O}\left(\frac{1}{2\lambda_{\max}(FF^T)}\right)$  and  $\mathbf{w}(0) = 0$ . The outputs of the linear network evolve as

$$\mathbf{g}(k+1) - \mathbf{g}(k) = F(\mathbf{w}(k+1) - \mathbf{w}(k)) = -\eta FF^T(\mathbf{g}(k) - \mathbf{y})$$

809 Thus, the distance of the outputs to the labels evolves as

$$\mathbf{g}(k) - \mathbf{y} = \mathbf{g}(k-1) - \eta FF^T(\mathbf{g}(k-1) - \mathbf{y}) - \mathbf{y} = (\mathbf{I} - \eta FF^T)(\mathbf{g}(k-1) - \mathbf{y}) = (\mathbf{I} - \eta FF^T)^k(\mathbf{g}(0) - \mathbf{y})$$

810 Using the previous result, we can express the change in the weights during training as

$$\begin{aligned} \mathbf{w}(K) - \mathbf{w}(0) &= \sum_{k=0}^{K-1} \mathbf{w}(k+1) - \mathbf{w}(k) \\ &= -\sum_{k=0}^{K-1} \eta F^T(\mathbf{g}(k) - \mathbf{y}) \\ &= -\sum_{k=0}^{K-1} \eta F^T(\mathbf{I} - \eta FF^T)^k(\mathbf{g}(0) - \mathbf{y}) \\ &= \sum_{k=0}^{K-1} \eta F^T(\mathbf{I} - \eta FF^T)^k \mathbf{y} - \sum_{k=0}^{K-1} \eta F^T(\mathbf{I} - \eta FF^T)^k \mathbf{g}(0) \end{aligned}$$

811 To bound the change in the weights, we bound each term individually.

$$\begin{aligned} \|\eta F^T \sum_{k=0}^{K-1} (\mathbf{I} - \eta FF^T)^k \mathbf{y}\|_2^2 &= \mathbf{y}^T \left( \sum_{k=0}^{K-1} (\mathbf{I} - \eta FF^T)^k \right)^T FF^T \left( \sum_{k=0}^{K-1} (\mathbf{I} - \eta FF^T)^k \right) \mathbf{y} \\ &= \mathbf{y}^T \left( \sum_{i=1}^n \frac{1 - (1 - \eta \lambda_i)^K}{\lambda_i} v_i v_i^T \right)^T \sum_{i=1}^n \lambda_i v_i v_i^T \left( \sum_{i=1}^n \frac{1 - (1 - \eta \lambda_i)^K}{\lambda_i} v_i v_i^T \right) \mathbf{y} \\ &= \mathbf{y}^T \left( \sum_{i=1}^n \lambda_i \left( \frac{1 - (1 - \eta \lambda_i)^K}{\lambda_i} \right)^2 v_i v_i^T \right) \mathbf{y} \\ &\leq \mathbf{y}^T \left( \sum_{i=1}^n \lambda_i^{-1} v_i v_i^T \right) \mathbf{y} \\ &= \mathbf{y}^T (FF^T)^{-1} \mathbf{y} \end{aligned}$$

812

$$\left\| \sum_{k=0}^{K-1} \eta F^T(\mathbf{I} - \eta FF^T)^k \mathbf{g}(0) \right\|_2 \leq \eta \sqrt{n} \left( \sum_{k=0}^{K-1} (1 - \eta \lambda_{\min}(FF^T))^k \right) \|\mathbf{g}(0)\|_2 \leq 0$$

813 Taking these bounds together yields

$$\|\mathbf{w}(K) - \mathbf{w}(0)\|_2 \leq \sqrt{\mathbf{y}^T (FF^T)^{-1} \mathbf{y}}$$

814 Let  $\epsilon \in \{\pm 1\}^n$ . Then it holds

$$\langle \epsilon, X\mathbf{w} \rangle = \langle \epsilon, X(\mathbf{w} - \mathbf{w}(0)) \rangle + \langle \epsilon, X\mathbf{w}(0) \rangle \leq \sqrt{n} \|\mathbf{w} - \mathbf{w}(0)\|_2 + \langle \epsilon, X\mathbf{w}(0) \rangle = \sqrt{n} \|\mathbf{w} - \mathbf{w}(0)\|_2$$

815 Let  $\mathcal{F}_R = \{\langle \mathbf{w}, \mathbf{x} \rangle : \|\mathbf{w} - \mathbf{w}(0)\| \leq R\}$ .

$$\mathcal{R}_S(\mathcal{F}_R) = \frac{1}{n} \mathbb{E}_{\epsilon \sim \{\pm 1\}^n} \left[ \sup_{\|\mathbf{w} - \mathbf{w}(0)\| < R} \langle \epsilon, X\mathbf{w} \rangle \right] < \frac{1}{\sqrt{n}} R = \frac{1}{\sqrt{n}} R$$

$$\begin{aligned}
\sup_{f \in \mathcal{F}_R} L_D(f) - L_S(f) &\leq 2\mathcal{R}_S(\mathcal{F}_R) + \mathcal{O}\left(\sqrt{\frac{\log \frac{2}{\delta}}{2n}}\right) \\
&\leq 2\frac{1}{\sqrt{n}}R + \mathcal{O}\left(\sqrt{\frac{\log \frac{2}{\delta}}{2n}}\right) \\
&= 2\sqrt{\frac{\mathbf{y}^T (FF^T)^{-1} \mathbf{y}}{n}} + \mathcal{O}\left(\sqrt{\frac{\log \frac{2}{\delta}}{2n}}\right)
\end{aligned}$$

816 Using

$$\|\mathbf{g} - \mathbf{y}\|_2 \leq \|(\mathbf{I} - \eta FF^T)^k\|_2 \|\mathbf{g}(0) - \mathbf{y}\|_2 \leq (1 - \eta \lambda_{\min}(FF^T))^k \sqrt{n} \leq 1$$

817 for sufficiently large  $k \geq \log \sqrt{n}$ , we can bound  $L_S$  as follows.

$$L_S(f) = \frac{1}{n} \sum_{i=1}^n |g(f(x_i)) - y|^2 = \frac{1}{n} \|\mathbf{g} - \mathbf{y}\|_2^2 \leq \frac{1}{\sqrt{n}}$$

818

□

## 819 C Proof of Theorem 4

820 In this section, we formalize the problem setting, restate Theorem 4 and provide its proof. The  
821 dominant term,  $\mathbf{y}^\top (FF^\top)^{-1} \mathbf{y}$ , shows that generalization improves when the label vector  $\mathbf{y}$  aligns  
822 well with the top eigenspaces of  $FF^\top$ . In contrastive learning, however, the downstream task is not  
823 known during pretraining, so it is unclear which directions in the feature space will ultimately be  
824 important.

825 To address this, we consider downstream tasks that arise by randomly sampling two latent classes  
826  $c^+, c^- \in \mathcal{C}$  according to a distribution  $\rho$ . For each such pair, we assume the existence of class-specific  
827 vectors  $\mathbf{v}_{c^+}, \mathbf{v}_{c^-}$  such that the optimal linear classifier in the feature space is given by  $\mathbf{v} = \mathbf{v}_{c^+} - \mathbf{v}_{c^-}$ .  
828 Specifically, the class posterior is given by

$$\mathbb{P}(Y_i = +1 \mid \mathbf{v}) = \frac{1 + (F\mathbf{v})_i}{2}, \quad \mathbb{P}(Y_i = -1 \mid \mathbf{v}) = \frac{1 - (F\mathbf{v})_i}{2},$$

829 where  $F \in \mathbb{R}^{n \times d}$  is the feature matrix.

830 Since downstream tasks are unknown at pretraining time, designing robust representations for  
831 contrastive learning requires optimizing for generalization over a distribution of tasks. Assuming  $\rho$   
832 is uniform over class pairs, we study which spectral properties of  $FF^\top$  lead to improved average  
833 generalization. Specifically, we aim to minimize the expected surrogate loss:

$$\mathcal{L}(F) := \mathbb{E}_{\mathbf{v}, Y} [Y^\top (FF^\top)^{-1} Y],$$

834 where the expectation is over random task vectors  $\mathbf{v}$  and induced labels  $Y \in \{\pm 1\}^n$ .

835 The following theorem shows the optimal structure of  $F$  to enhance the generalization on a general  
836 downstream task.

837 **Theorem 4** (Restatement of Theorem 4 (informal)). *Let  $F \in \mathbb{R}^{n \times n}$  be a feature matrix. Then,*  
838 *under a fixed trace constraint on  $FF^\top$ , the objective  $\mathcal{L}(F)$  is minimized when  $FF^\top$  has a uniform*  
839 *spectrum; that is, all eigenvalues are equal:  $\lambda_1 = \lambda_2 = \dots = \lambda_n$ .*

840 **Theorem 4** (Optimality of Uniform Spectrum under Trace Constraint). *Let  $F \in \mathbb{R}^{n \times n}$  be full-rank,*  
841 *and define  $G := FF^\top \in \mathbb{R}^{n \times n}$ . Suppose the trace is fixed, i.e.,  $\text{Tr}(FF^\top) = \sum_{i=1}^n \lambda_i = c$  for some*  
842 *constant  $c > 0$ , where  $\lambda_i$  are the eigenvalues of  $FF^\top$ . Then the expected quadratic form*

$$\mathcal{L}(F) := \mathbb{E}_{Y \sim \mathcal{D}} [Y^\top (FF^\top)^{-1} Y]$$

843 *is minimized when  $FF^\top = \lambda I_n$ , i.e., when all eigenvalues are equal.*

844 *Proof.* We first simplify the objective. Conditioning on  $v$ , the second moment of  $Y$  satisfies

$$\mathbb{E}[YY^\top \mid \mathbf{v}] = (F\mathbf{v})(F\mathbf{v})^\top + \text{diag}(1 - (F\mathbf{v})^2).$$

845 Thus,

$$\mathcal{L}(F) = \mathbb{E}_v [\text{tr}((FF^\top)^{-1}(F\mathbf{v})(F\mathbf{v})^\top) + \text{tr}((FF^\top)^{-1} \text{diag}(1 - (F\mathbf{v})^2))].$$

846 Since  $F$  is square and invertible, we have  $(XX^\top)^{-1}F = F^{-\top}$ , and thus

$$\text{tr}((FF^\top)^{-1}(F\mathbf{v})(F\mathbf{v})^\top) = \text{tr}(F^{-\top}\mathbf{v}\mathbf{v}^\top F^\top) = \text{tr}(\mathbf{v}\mathbf{v}^\top) = \|\mathbf{v}\|_2^2,$$

847 where we used the cyclic property of the trace and the fact that  $F^\top F^{-\top} = I$ .

848 Expanding the second term gives

$$\text{tr}((FF^\top)^{-1} \text{diag}(1 - (F\mathbf{v})^2)) = \text{tr}((F^\top)^{-1}) - \text{tr}((FF^\top)^{-1} \text{diag}((F\mathbf{v})^2)).$$

849 Using a similar simplification as above, the contribution from  $\text{diag}((F\mathbf{v})^2)$  exactly cancels the  $\|\mathbf{v}\|_2^2$   
850 term when taking expectation over  $\mathbf{v}$ . Thus, we conclude that

$$\mathcal{L}(F) = \text{tr}((FF^\top)^{-1}).$$

851 Next, we optimize  $\mathcal{L}(F)$  under the constraint  $\text{tr}(FF^\top) = c$ . Let  $G := FF^\top \succ 0$ . Then  $G$  admits an  
852 eigenvalue decomposition  $G = U\Lambda U^\top$ , where  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$  with  $\lambda_i > 0$ . Thus,

$$\mathcal{L}(F) = \text{tr}(G^{-1}) = \sum_{i=1}^n \lambda_i^{-1}, \quad \text{and} \quad \text{tr}(G) = \sum_{i=1}^n \lambda_i = c.$$

853 Define the Lagrangian:

$$\mathcal{L}(\lambda_1, \dots, \lambda_n, \mu) = \sum_{i=1}^n \lambda_i^{-1} + \mu \left( \sum_{i=1}^n \lambda_i - c \right).$$

854 Taking partial derivatives with respect to each  $\lambda_i$  and setting them to zero:

$$\frac{\partial \mathcal{L}}{\partial \lambda_i} = -\lambda_i^{-2} + \mu = 0 \quad \Rightarrow \quad \lambda_i = \frac{1}{\sqrt{\mu}}, \quad \text{for all } i = 1, \dots, n.$$

855 Thus, all  $\lambda_i$  are equal. Substituting into the constraint  $\sum_{i=1}^n \lambda_i = c$  gives:

$$n \cdot \lambda_i = c \quad \Rightarrow \quad \lambda_i = \frac{c}{n}.$$

856

□

857 Theorem 4 suggests that when the downstream task is unknown, learning a feature matrix  $FF^\top$  with  
858 a uniform spectrum is optimal. See also Appendix I for an illustrative example involving spurious  
859 correlations, which further motivates the benefits of a uniform spectrum.

## 860 D Lemma 4

861 **Lemma 4.** Let  $F \in \mathbb{R}^{n \times d}$  be a feature matrix with singular values  $\sigma_1, \dots, \sigma_r$ , where  $r = \min(n, d)$ ,  
862 and let  $\lambda_1, \dots, \lambda_r$  denote the eigenvalues of  $FF^\top$ . Assume the singular values are normalized so  
863 that

$$\sum_{i=1}^r (\sigma_i - 1)^2 \leq \varepsilon \quad \text{for some } \varepsilon \in (0, 1].$$

864 Then,

$$\sum_{i=1}^r (\lambda_i - 1)^2 \leq C\varepsilon \quad \text{for a constant } C.$$

865 **Remark 5.** Lemma 4 shows that flattening the singular values of  $F$  implies a corresponding flattening  
 866 of the eigenvalues of  $FF^\top$ , leading to a more uniform spectrum.

867 **Remark 6.** Thus, to encourage spectrum uniformity of  $FF^\top$ , it suffices to regularize the singular  
 868 values of  $F$ , which is often simpler and more efficient in contrastive learning frameworks.

869 *Proof.* Let  $F \in \mathbb{R}^{n \times d}$  have rank  $r = \min(n, d)$ , and let its singular value decomposition be  
 870  $F = U\Sigma V^\top$ , where  $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r)$  with singular values  $\sigma_i > 0$ . Then the eigenvalues of  
 871  $FF^\top \in \mathbb{R}^{n \times n}$  are exactly  $\lambda_i = \sigma_i^2$  for  $i = 1, \dots, r$ , and 0 otherwise.

872 By assumption, the singular values are normalized and satisfy

$$\sum_{i=1}^r (\sigma_i - 1)^2 \leq \varepsilon.$$

873 Our goal is to bound

$$\sum_{i=1}^r (\lambda_i - 1)^2.$$

874 To relate these two expressions, we use the identity:

$$(\sigma_i^2 - 1)^2 = (\sigma_i - 1)^2(\sigma_i + 1)^2.$$

875 Since  $\sigma_i$  are normalized and  $(\sigma_i - 1)^2 \leq \varepsilon$ , we have  $\sigma_i \in [1 - \sqrt{\varepsilon}, 1 + \sqrt{\varepsilon}]$ . Therefore,

$$(\sigma_i + 1)^2 \leq (1 + \sqrt{\varepsilon} + 1)^2 = (2 + \sqrt{\varepsilon})^2 \leq 9 \quad \text{for } \varepsilon \leq 1.$$

876 Then:

$$(\sigma_i^2 - 1)^2 = (\sigma_i - 1)^2(\sigma_i + 1)^2 \leq 9(\sigma_i - 1)^2.$$

877 Summing over  $i = 1, \dots, r$ , we obtain:

$$\sum_{i=1}^r (\lambda_i - 1)^2 = \sum_{i=1}^r (\sigma_i^2 - 1)^2 \leq 9 \sum_{i=1}^r (\sigma_i - 1)^2 \leq 9\varepsilon.$$

878 □

## 879 **E Algorithmic Details of Contrastive Pretraining with Spectral** 880 **Regularization**

881 This section presents the pseudocode for our contrastive pretraining framework. Algorithm 1 out-  
 882 lines the self-supervised training procedure based on SimCLR with optional spectral regularization.  
 883 Algorithm 2 describes the computation of the spectrum flattening loss.

---

**Algorithm 1** Self-supervised Contrastive Pretraining with Spectrum Regularization

---

**Input:** Encoder  $f_\theta$ , projection head  $g$ , temperature  $\tau$ , augmentation pipeline  $\mathcal{T}$ , spectral weight  $\alpha_{\text{spec}}$ , epochs  $N$ , batch size  $B$   
Initialize parameters of  $f_\theta$  and  $g$   
**for** epoch = 1 to  $N$  **do**  
  **for** each mini-batch  $\{x_i\}_{i=1}^B$  **do**  
    *// Stage 1: Data Augmentation*  
    Sample two augmentations  $t, t' \sim \mathcal{T}$   
     $x_i^1 = t(x_i), x_i^2 = t'(x_i)$  for  $i = 1, \dots, B$   
    *// Stage 2: Feature Extraction*  
     $z_i^1 = g(f_\theta(x_i^1)), z_i^2 = g(f_\theta(x_i^2))$   
    Normalize:  $\tilde{z}_i^1 = z_i^1 / \|z_i^1\|_2, \tilde{z}_i^2 = z_i^2 / \|z_i^2\|_2$   
    Stack all views:  $\mathcal{Z} = \{\tilde{z}_1^1, \tilde{z}_1^2, \dots, \tilde{z}_B^1, \tilde{z}_B^2\} \in \mathbb{R}^{2B \times d}$   
    *// Stage 3: Loss Computation*  
    Contrastive loss:  
    
$$\mathcal{L}_{\text{CL}} = \frac{1}{2B} \sum_{i=1}^{2B} -\log \frac{\exp(\text{sim}(\tilde{z}_i, \tilde{z}_{p(i)})/\tau)}{\sum_{j \neq i} \exp(\text{sim}(\tilde{z}_i, \tilde{z}_j)/\tau)}$$
  
    where  $\text{sim}(a, b) = a^\top b$  and  $p(i)$  is the positive pair index  
    **if** spectral regularization is enabled **then**  
      Compute  $\mathcal{L}_{\text{spec}}$  using Algorithm 2  
    **end if**  
    Total loss:  $\mathcal{L} = \mathcal{L}_{\text{CL}} + \alpha_{\text{spec}} \mathcal{L}_{\text{spec}}$   
    *// Stage 4: Optimization*  
    Update  $f_\theta, g$  via gradient step on  $\nabla_\theta \mathcal{L}$   
  **end for**  
**end for**  
**Output:** Pretrained encoder  $f_\theta$

---

---

**Algorithm 2** Spectrum Flattening Loss Computation ( $\mathcal{L}_{\text{spec}}$ )

---

**Input:** Feature matrix  $Z \in \mathbb{R}^{B \times d}$  from current mini-batch  
**Output:** Spectrum loss  $\mathcal{L}_{\text{spec}}$   
Compute singular values:  $U, S, V = \text{svd}(Z)$  //  $S \in \mathbb{R}^r$  where  $r = \min(B, d)$   
Normalize:  $S_{\text{norm}} \leftarrow S / \max(S)$   
Compute loss:  
$$\mathcal{L}_{\text{spec}} = \frac{1}{r} \sum_{i=1}^r (S_{\text{norm},i} - 1)^2$$
  
**Return:**  $\mathcal{L}_{\text{spec}}$

---

## 884 F Hyperparameters

885 We employed the SimCLR framework to train ResNet encoders for our approach. To ensure a fair  
886 comparison on SimCLR, we adopted the same encoder architectures as those used in Hamidieh  
887 et al. [2024], using ResNet-18 for all datasets except CelebA, where ResNet-50 was used. Detailed  
888 hyperparameter configurations for SimCLR across all datasets are provided in Table 3. To select the  
889 regularization strength  $\alpha$  for the spectral flattening loss, we performed a grid search over the values  
890  $\{0.001, 0.005, 0.01, 0.05\}$  using validation performance on the worst-group accuracy as the selection  
891 criterion. The best-performing value was then fixed for each dataset across all evaluation protocols.

Table 3: Hyperparameter settings and encoder architectures for SimCLR pretraining.

| Dataset      | Encoder   | Learning Rate | Batch Size | Weight Decay | Epochs | Regularizer $\alpha$ |
|--------------|-----------|---------------|------------|--------------|--------|----------------------|
| celebA       | ResNet-50 | 0.01          | 128        | 1e-4         | 400    | 0.01                 |
| cmnist       | ResNet-18 | 0.01          | 256        | 1e-4         | 1000   | 0.01                 |
| metashift    | ResNet-18 | 0.01          | 256        | 1e-4         | 1000   | 0.005                |
| spurcifar-10 | ResNet-18 | 0.01          | 256        | 1e-4         | 1000   | 0.01                 |
| waterbirds   | ResNet-18 | 0.01          | 256        | 1e-4         | 1000   | 0.01                 |

Table 4: Comparison of our pretraining strategy (SimCLR + spectral regularizer) with supervised models in terms of average and worst-group accuracies (%). Our pretraining strategy achieves comparable performance to supervised models, both in terms of average and worst-group accuracies (%), despite not utilizing any ground-truth labels or group information.

| DATASET      | AVERAGE ACCURACY |      |             | WORST-GROUP ACCURACY |             |             |
|--------------|------------------|------|-------------|----------------------|-------------|-------------|
|              | SSRL-BASE        | OURS | SUPERVISED  | SSRL-BASE            | OURS        | SUPERVISED  |
| CELEBA       | 82.1             | 88.5 | <b>91.9</b> | 76.7                 | <b>84.2</b> | 81.7        |
| CMNIST       | 82.5             | 97.0 | <b>98.4</b> | 81.7                 | <b>95.1</b> | 94.9        |
| METASHIFT    | 55.1             | 78.1 | <b>89.8</b> | 45.5                 | 67.4        | <b>83.5</b> |
| SPURCIFAR-10 | 69.3             | 80.1 | <b>89.9</b> | 36.5                 | 59.7        | <b>79.6</b> |
| WATERBIRDS   | 47.5             | 57.9 | <b>67.9</b> | 43.8                 | <b>56.7</b> | 41.1        |

## G Closing the Gap to Supervised Pretraining

SSRL has demonstrated significant potential in narrowing the performance gap with supervised learning approaches, particularly for general representation learning. Similar to Hamidieh et al. [2024], we utilized a consistent encoder model and varied only the pretraining strategies, ensuring that other variables, such as hyperparameter settings and model selection criteria, remained fixed. Notably, supervised pretraining requires labeled data, whereas SSRL methods do not, reducing the overall annotation cost significantly. While this inherently makes the comparison less direct, the goal of this evaluation is to measure how closely SSRL methods, and specifically our proposed approach, can match or surpass supervised pretraining strategies.

Table 4 compares the SSRL-base (SimCLR) method, our proposed method, and the supervised approach. The results highlight how our method narrows the gap with supervised learning in terms of average accuracy. Additionally, in worst-group accuracy, our approach outperforms both SimCLR and the supervised method on datasets such as CelebA, CMNIST, and Waterbirds. Notably, on CelebA, our method achieves a significant improvement in worst-group accuracy—approximately 4% higher than the supervised approach—while maintaining strong overall accuracy.

## H Comparing with More Baselines

We evaluate the effectiveness of our spectral regularization method by comparing it against two representative baselines: Barlow Twins [Zbontar et al., 2021] and DirectDLR [Jing et al., 2021]. In addition, we examine how our regularizer performs when applied on top of SimSiam. We report both average accuracy and worst-group accuracy across five standard spurious correlation benchmarks as shown in Tables 5 and 6.

## I Example Motivating Importance of Uniforming the Spectrum

We present a simple example to highlight the role of the eigenspectrum of the feature matrix. Let  $F$  be a fixed feature matrix with orthonormal eigenvectors  $v^+$ ,  $v^-$ , and  $v^s$ , corresponding to eigenvalues  $\lambda^+$ ,  $\lambda^-$ , and  $\lambda^s$ , respectively. Here,  $v^+$  and  $v^-$  represent class-discriminative directions for labels  $+1$  and  $-1$ , while  $v^s$  is a spurious direction with spurious correlation strength  $\alpha$ .

Table 5: Worst-group accuracy (%) comparison between Barlow Twins, DirectDLR, and our spectral regularization applied to SimCLR and SimSiam.

| DATASET      | BARLOW TWINS | DIRECTDLR | SIMSIAI + SPEC (OURS) | SIMCLR + SPEC (OURS) |
|--------------|--------------|-----------|-----------------------|----------------------|
| CMNIST       | 57.05        | 90.32     | 94.4                  | <b>95.1</b>          |
| SPURCIFAR-10 | 6.0          | 20.58     | 50.1                  | <b>59.7</b>          |
| CELEBA       | 39.99        | 68.68     | 74.17                 | <b>84.2</b>          |
| METASHIFT    | 58.33        | 53.84     | <b>69.23</b>          | 67.4                 |
| WATERBIRDS   | 43.13        | 47.32     | 49.08                 | <b>56.7</b>          |

Table 6: Average accuracy (%) of Barlow Twins, DirectDLR, and our spectral regularization applied to SimCLR and SimSiam.

| DATASET      | BARLOW TWINS | DIRECTDLR | SIMSIAI + SPEC (OURS) | SIMCLR + SPEC (OURS) |
|--------------|--------------|-----------|-----------------------|----------------------|
| CMNIST       | 93.1         | 96.35     | <b>97.2</b>           | 97.0                 |
| SPURCIFAR-10 | 22.0         | 50.2      | 71.78                 | <b>80.1</b>          |
| CELEBA       | 84.61        | 78.23     | <b>89.12</b>          | 88.5                 |
| METASHIFT    | 64.6         | 75.28     | 77.52                 | <b>78.1</b>          |
| WATERBIRDS   | 54.67        | 53.46     | <b>60.55</b>          | 57.9                 |

Specifically, the label generation process is as follows: define  $v := \frac{1}{2}(v^+ - v^-)$ . For each sample  $x_i$ , let  $f_i = f(x_i)$ , and define the perturbed direction:

$$v_i = \begin{cases} v, & \text{with probability } 1 - \alpha, \\ \frac{1}{2}(v + v^s), & \text{with probability } \alpha. \end{cases}$$

The label  $y_i \in \{\pm 1\}$  is sampled according to:

$$\mathbb{P}(y_i = +1 \mid f_i) = \frac{1 + f_i^\top v_i}{2}.$$

Let  $g_i = g_w(f_i)$ , and consider the squared loss:

$$\Phi(\mathbf{g}, \mathbf{y}) = \frac{1}{2} \sum_{i=1}^n (1 - y_i g_i)^2.$$

**Lemma 7.** *The expected gradient flow under the randomness of the labels satisfies:*

$$\mathbb{E}[FF^\top \cdot \nabla_{\mathbf{g}} \Phi] = \left( FF^\top \mathbf{g} - \left[ \left(1 - \frac{\alpha}{2}\right) \cdot \frac{1}{2} \lambda^+ v^+ - \left(1 - \frac{\alpha}{2}\right) \cdot \frac{1}{2} \lambda^- v^- + \frac{\alpha}{2} \lambda^s v^s \right] \right).$$

This result shows that even a weak spurious correlation ( $\alpha \ll 1$ ) can dominate the training dynamics if  $\lambda^s \gg \lambda^+, \lambda^-$ . In contrast, under a flat spectrum (i.e., uniform eigenvalues), the influence of the spurious direction scales linearly with  $\alpha$ , making the model more robust to such noise.

*Proof.* The first step is to compute the loss with respect to each model output  $g_i$  which is given by

$$\frac{d\Phi}{dg_i} = -2y_i(1 - y_i g_i).$$

The sources of randomness are from sampling both  $y$  and the random mixing of the spurious feature. By the law of total expectation, the expectation with respect to  $y$  and  $v$  is given by

$$\mathbb{E}_{v_i | f_i} \left[ \mathbb{E}_{y_i | x_i, v_i} \left[ \frac{d\Phi}{dg_i} \right] \right]$$

The inner expectation is given by

$$\mathbb{E}_{y_i | f_i, v_i} \left[ \frac{d\Phi}{dg_i} \right] = -2\mathbb{E}[y_i - y_i^2 g_i] = 2g_i - 2f_i^\top v_i,$$



930 since  $\mathbb{E}[y_i \mid f_i, v_i] = f_i^\top v_i$  and  $y_i^2 = 1$ . Further observe

$$\mathbb{E}_{v_i}[x_i^\top v_i] = (1 - \alpha)f_i^\top v + \alpha f_i^\top \left( \frac{1}{2}(v + v^s) \right) = \left( 1 - \frac{\alpha}{2} \right) f_i^\top v + \frac{\alpha}{2} f_i^\top v^s.$$

931 Combining the above three equations and letting  $\tilde{v} := \left( 1 - \frac{\alpha}{2} \right) v + \frac{\alpha}{2} v^s$ , we get

$$\mathbb{E}_{v_i, y_i} \left[ \frac{d\Phi}{dg_i} \right] = 2g_i - 2f_i^\top \tilde{v}.$$

932 Stacking across all samples, let  $\mathbf{g} = [g_1, \dots, g_n]^\top$ . Then:

$$\nabla_{\mathbf{g}} \Phi = (\mathbf{g} - F\tilde{v}).$$

933 Applying the data covariance operator  $FF^\top$  gives:

$$\mathbb{E}[FF^\top \cdot \nabla_{\mathbf{g}} \Phi] = (FF^\top \mathbf{g} - FF^\top F\tilde{v}).$$

934 By the assumption that  $v^+$ ,  $v^-$ , and  $v^s$  are orthonormal eigenvectors of  $FF^\top$  with eigenvalues  $\lambda^+$ ,  
935  $\lambda^-$ , and  $\lambda^s$ , and  $v = \frac{1}{2}(v^+ - v^-)$ . Then:

$$F\tilde{v} = \left( 1 - \frac{\alpha}{2} \right) \cdot \frac{1}{2} Fv^+ - \left( 1 - \frac{\alpha}{2} \right) \cdot \frac{1}{2} Fv^- + \frac{\alpha}{2} Fv^s,$$

936 and applying  $FF^\top$ :

$$FF^\top F\tilde{v} = \left( 1 - \frac{\alpha}{2} \right) \cdot \frac{1}{2} \lambda^+ v^+ - \left( 1 - \frac{\alpha}{2} \right) \cdot \frac{1}{2} \lambda^- v^- + \frac{\alpha}{2} \lambda^s v^s.$$

937 Substituting this expression concludes the proof:

$$\mathbb{E}[FF^\top \cdot \nabla_{\mathbf{g}} \Phi] = \frac{2}{n} \left( FF^\top \mathbf{g} - \left[ \left( 1 - \frac{\alpha}{2} \right) \cdot \frac{1}{2} \lambda^+ v^+ - \left( 1 - \frac{\alpha}{2} \right) \cdot \frac{1}{2} \lambda^- v^- + \frac{\alpha}{2} \lambda^s v^s \right] \right).$$

938 □

## 939 J Computational Efficiency of Regularization Terms

940 Several self-supervised learning methods aim to mitigate representation collapse and redundancy by  
941 decorrelating feature dimensions. **Barlow Twins** minimizes the cross-correlation matrix between two  
942 views and enforces it to be close to the identity, effectively promoting invariance while discouraging  
943 redundancy Zbontar et al. [2021]. **VICReg** combines an invariance term with variance and covariance  
944 regularizers, penalizing off-diagonal entries in the covariance matrix Bardes et al. [2021]. Both  
945 methods require computing and differentiating through batch-wise matrices of size  $d \times d$ , incurring a  
946 cost of  $O(nd^2)$  to form the matrix, and an additional  $O(d^2)$  for computing the regularization loss.

947 **Whitening-based methods**, such as ZCA whitening Ermolov et al. [2021], go further by requiring  
948 not only the covariance matrix  $F^\top F \in \mathbb{R}^{d \times d}$  but also its inverse square root, computed via eigende-  
949 composition. This results in a total cost of  $O(nd^2 + d^3)$ , making them significantly more expensive  
950 in high-dimensional settings.

951 In contrast, our **spectral flattening regularizer** only requires access to the singular values of the  
952 feature matrix. These can be obtained by computing the eigenvalues of either  $F^\top F \in \mathbb{R}^{d \times d}$  or  
953  $FF^\top \in \mathbb{R}^{n \times n}$ , or by directly applying SVD to  $F \in \mathbb{R}^{n \times d}$ . Since all three methods yield the  
954 same singular values, one can select the most efficient strategy depending on the dimensions of  $F$ .  
955 Specifically, computing eigenvalues of  $F^\top F$  is preferred when  $d \ll n$ , while  $FF^\top$  may be used  
956 when  $n \ll d$ . Direct SVD provides a balanced alternative with cost  $O(nd^2)$  when  $n \geq d$ . This makes  
957 our method scalable to large batch sizes and embedding dimensions, while remaining competitive  
958 with or cheaper than other decorrelation strategies Shigeto et al. [2023].

959 All experiments using the spectral flattening regularizer were run on a single NVIDIA A100 GPU  
960 with 40 GB memory.

Table 7: Computational complexity of regularization terms for different SSRL methods. Here,  $n$  is the batch size and  $d$  is the feature dimension.

| METHOD                             | FORWARD + BACKWARD COST |
|------------------------------------|-------------------------|
| BARLOW TWINS ZBONTAR ET AL. [2021] | $O(nd^2 + d^2)$         |
| VICREG BARDES ET AL. [2021]        | $O(nd^2 + d^2)$         |
| WHITENING ERMOLOV ET AL. [2021]    | $O(nd^2 + d^3)$         |
| SPECTRAL FLATTENING (OURS)         | $O(nd^2)$               |

## 961 K Limitations

962 While our spectral flattening regularizer is more efficient than full covariance-based methods in many  
 963 practical settings, it does incur some additional computational cost due to the need for singular value  
 964 computation. In our implementation, we compute the singular values of the batch feature matrix  
 965  $F \in \mathbb{R}^{n \times d}$  via eigendecomposition of  $F^\top F$ , which scales as  $O(nd^2)$  when  $n \geq d$ . This cost is  
 966 typically lower than that of Barlow Twins or VICReg, both of which require full covariance matrices  
 967 and gradients through  $d \times d$  terms. However, the quadratic dependence on  $d$  may still pose challenges  
 968 for extremely high-dimensional embeddings. See Appendix J for a detailed comparison of methods  
 969 and costs.