Stabilizing Feature Geometry in Noisy Pretrained Models for Robust Downstream Tasks

Supplementary Material

1. Experiments

1.1. Detailed Setup for Experiments

In this paper, we report the accuracy of various models on ID and OOD datasets under different noise settings. Specifically, Tables 1 and 2 provide detailed training and test set splits for the ID and OOD datasets, while Table 3 summarizes the configurations of large-scale models pretrained on real-world noisy data.

In our experiments, we follow the experimental setup of NMTune to ensure a fair comparison between different methods and to validate the effectiveness of the proposed method across multiple tasks. The experiments involve various mainstream fine-tuning strategies, including Linear Probe (LP), Multilayer Perceptron fine-tuning (MLP), NMTune, and the method proposed in this paper. To ensure the fairness of performance comparisons, all methods adopt as consistent optimizer configurations and training parameters.

Specifically, for LP, we set the learning rate to 0.1 and the weight decay to 0. For MLP, NMTune, and the proposed method, we uniformly use a learning rate of 0.001 and a weight decay of 1e-4. The optimizer employed is AdamW, with a batch size of 64 and 30 training iterations for all methods. All methods are fine-tuned under a black-box setting, where the backbone remains frozen throughout training. The projection module is uniformly designed as a two-layer fully connected network (i.e., an MLP following the structure in [1]): the first layer maps the input feature dimension D to $4D$ with a ReLU activation function, followed by the second layer that projects it back to the original dimension D. This structure strikes a good balance between computational efficiency and expressive power, and is validated as stable and effective in prior research. Unless otherwise specified, the default noise type used in Feature Perturbation Consistency is salt-and-pepper noise, with an injection ratio of 10%. All experiments are conducted on NVIDIA V100 GPUs. Each experiment is repeated three times with different random seeds, and we report the average performance across runs.

1.2. Detailed Results for Experiments

Figures 1, 5, 6, and 7 further demonstrate the detailed experimental results of the two models across different datasets under the synthetic noise setting.

Table 1. In-domain (ID) datasets for ID evaluation.

Dataset	Classes	Train Size	Test Size	Evaluation Metric
CIFAR-10 [9]	10	50,000	10,000	accuracy
CIFAR-100 [9]	100	50,000	10,000	accuracy
Flowers102 [12]	102	2,040	6,149	mean per class
Food101 [4]	101	75,750	25,250	accuracy
OxfordPet [13]	37	3,680	3,669	mean per class
StanfordCars [8]	196	8,144	8,041	accuracy
DTD [2]	47	1,880	1,880	accuracy
Caltech101 [4]	102	3,060	6,084	mean per class

Table 2. Out-of-domain (OOD) datasets for OOD evaluation.

Dataset	Classes	Train Size	Test Size	Evaluation Metric
DomainNet Sketch [14]	345	48,212	20,916	accuracy
DomainNet Real [14]	345	120,906	52,041	accuracy
DomainNet Painting [14]	345	-	21,850	accuracy
DomainNet Clipart [14]	345	-	14,604	accuracy

Table 3. Overview of Pretrained Models under Real-World Noise Settings.

Model	pretrained Data	pretrained Method	Param. Size (M)
EfficientNet-B3 [18]	JFT-300M [6]	Noisy Student [19]	12.23
ResNetv2-152x2 [5]	ImageNet-21K [16]	BiT [7]	236.34
Swin-L [10]	ImageNet-21K [7]	Supervised [10]	196.74
ViT-L [3]	Laion-2B [17]	CLIP [15]	304.20
ConvNext-L [11]	Laion-2B [17]	CLIP [15]	200.13

1.3. Detailed Results for In-Depth Analysis Experiments

1.3.1. Analysis on Mitigating Noisy Data

To gain deeper insights into the noise robustness mechanism of our proposed method, we design a series of feature space visualization experiments to compare the feature distribution differences across various methods on two representative tasks: DomainNetSketch and CIFAR-100. Specifically, we extract the high-dimensional feature representations output by the models during the final training iteration and employ t-SNE to reduce their dimensionality to a 2D space, enabling intuitive observation of the learned feature structures. For each dataset, we randomly select five categories for visualization to enhance image readability and comparability. It is worth noting that the DomainNetSketch dataset contains 345 categories, while CIFAR-100 comprises 100 categories.

We repeat this process under multiple pretraining noise ratio settings, including 0%, 5%, 10%, 20%, and 30%. The results are further presented in Figures 8, 9, 10, and 11. These visualizations demonstrate that, compared to other

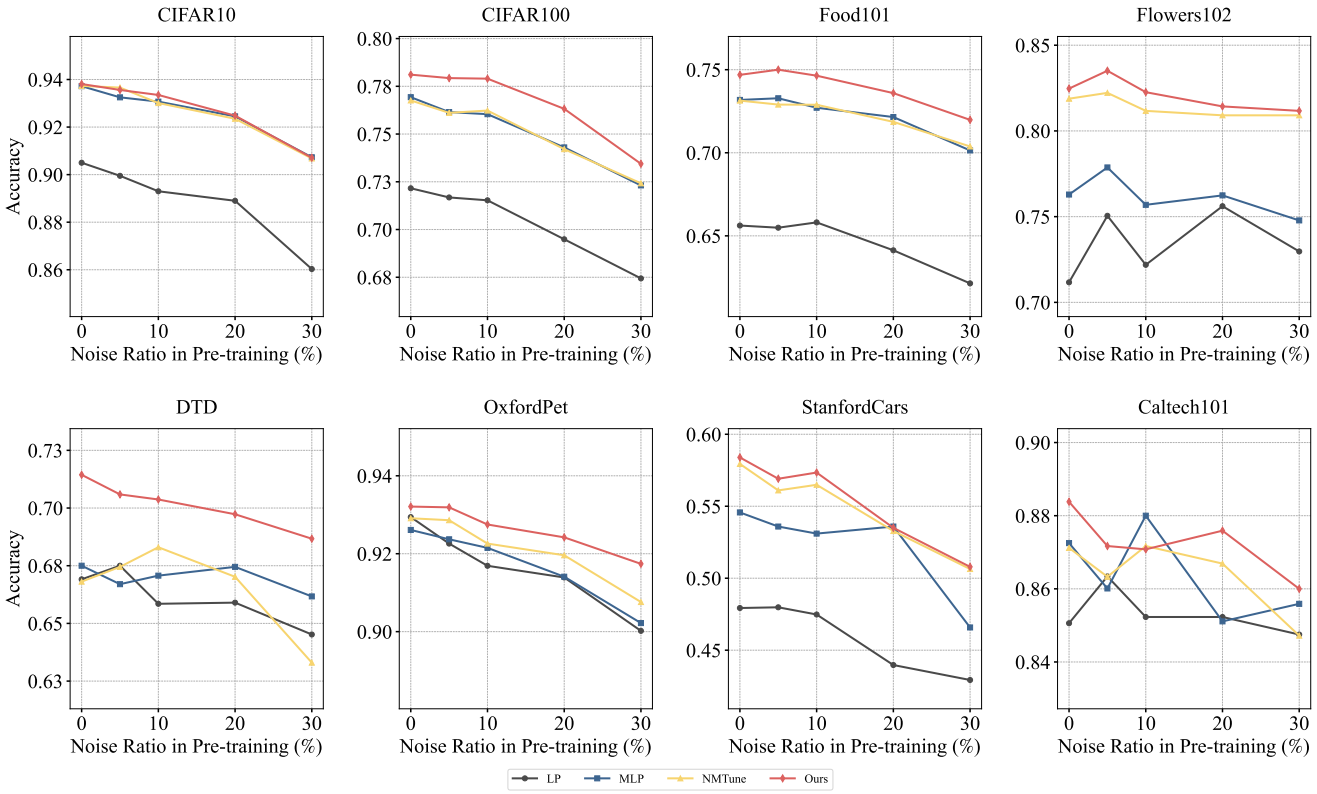
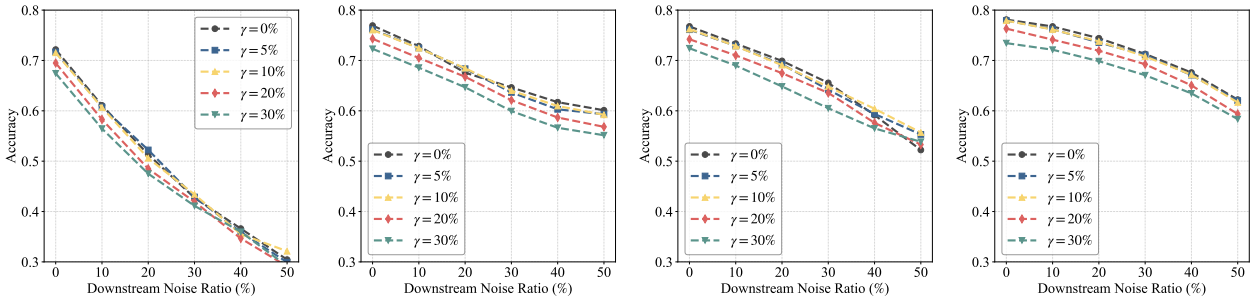


Figure 1. ImageNet-1K pretrained ResNet-50 in-domain (ID) evaluation results



(a) LP, CIFAR100 (b) MLP, CIFAR100 (c) NMTune, CIFAR100 (d) Ours, CIFAR100

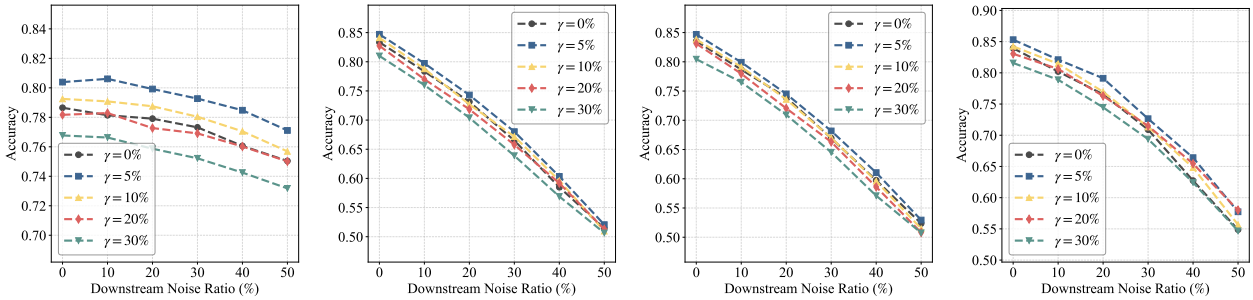
Figure 2. Evaluation of the ResNet-50/ImageNet-1K model on CIFAR-100 at various noise ratios.

methods, our approach consistently exhibits stronger feature structure stability across all pretraining noise levels: it not only maintains compact intra-class sample distributions but also enhances the clarity of boundaries between different categories, resulting in superior inter-class separability.

1.3.2. Analysis on Downstream Noise Robustness

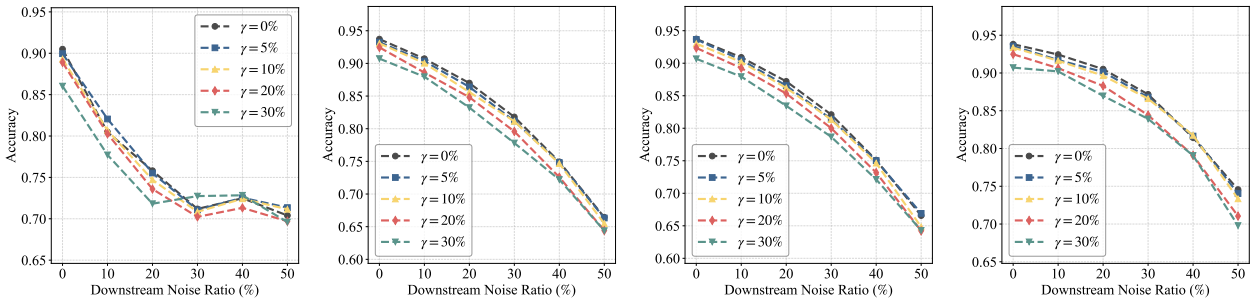
Furthermore, we investigate a more complex scenario where label noise exists simultaneously during both pretraining and downstream task phases. For the downstream phase, we select the CIFAR-10 and CIFAR-100 datasets, which are commonly used for studying label noise robustness. We construct symmetric label noise by randomly and uniformly permuting the labels of each category in the training set, with noise ratios set to {0%, 10%, 20%, 30%, 40%, 50%}. The performance of the two models under different noise ratios is presented in Figures 2, 3, and 4, respectively.

The results show that as the noise ratio in the downstream data increases, our proposed method exhibits significantly more robust performance compared to mainstream approaches. Notably, under high noise levels, the performance degradation of our model is markedly more gradual, further validating the superiority of the proposed method in



(a) LP, CIFAR10 (b) MLP, CIFAR10 (c) NMTune, CIFAR10 (d) Ours, CIFAR10

Figure 3. Evaluation of the ResNet-50/YFCC15M model on CIFAR-10 at various noise ratios.



(a) LP, CIFAR10 (b) MLP, CIFAR10 (c) NMTune, CIFAR10 (d) Ours, CIFAR10

Figure 4. Evaluation of the ResNet-50/ImageNet-1K model on CIFAR-10 at various noise ratios.

complex noise environments.

1.4. Additional Experiments

1.4.1. Visualization of Feature Evolution in Feature Perturbation Consistency

To better demonstrate the learning process in Feature Perturbation Consistency, our experiments further analyze the evolutionary trends of original features and perturbed features (obtained by injecting noise into the original features) on DomainNetSketch and CIFAR-100 (with five randomly selected categories). Specifically, we illustrate the training process of a ResNet-50 model pretrained on ImageNet-1K under varying synthetic noise ratios. The results (Figures 12, 13, 14, 15, 16) show that as training progresses, the model's feature representations gradually evolve toward greater discriminability, rather than merely aligning the original features and perturbed features. Under high noise ratios, the model effectively suppresses noise interference through the geometric direction consistency constraint, making features of the same class more compact and those of different classes better separated, demonstrating enhanced geometric structure stability. This phenomenon consistently appears across both the cross-domain DomainNetSketch and in-domain CIFAR-100 datasets, highlighting the robustness and generalization capability of the proposed method in different noise environments.

1.4.2. Applicability of Other Fine-tuning Methods

To further evaluate the robustness and generality of our approach, we extend the experiments to different fine-tuning strategies, including both full fine-tuning and parameter-efficient fine-tuning. Table 4 presents the comparison results. Specifically, we adopt ResNet-50 models pretrained on ImageNet-1K with varying noise levels (0%, 5%, and 20%) and validate them on OOD generalization tasks, using DomainNetSketch as the training set and all four target domains for evaluation. In the full fine-tuning setting, our method consistently outperforms the vanilla full fine-tuning baseline across all tasks. The averaged results show stable gains under each configuration, demonstrating that our method can also effectively adapt to the full fine-tuning paradigm.

We also investigate the applicability under parameter-efficient fine-tuning using Swin-L with LoRA. As shown in the lower block of Table 4, our method significantly improves over LoRA on all tasks, with the average score rising from 0.6386 to 0.6759. This indicates that the proposed approach is not only compatible with parameter-efficient tuning but also enhances its effectiveness.

1.4.3. Computational Complexity

We analyze the computational overhead introduced by the three proposed modules. The time and memory complexi-

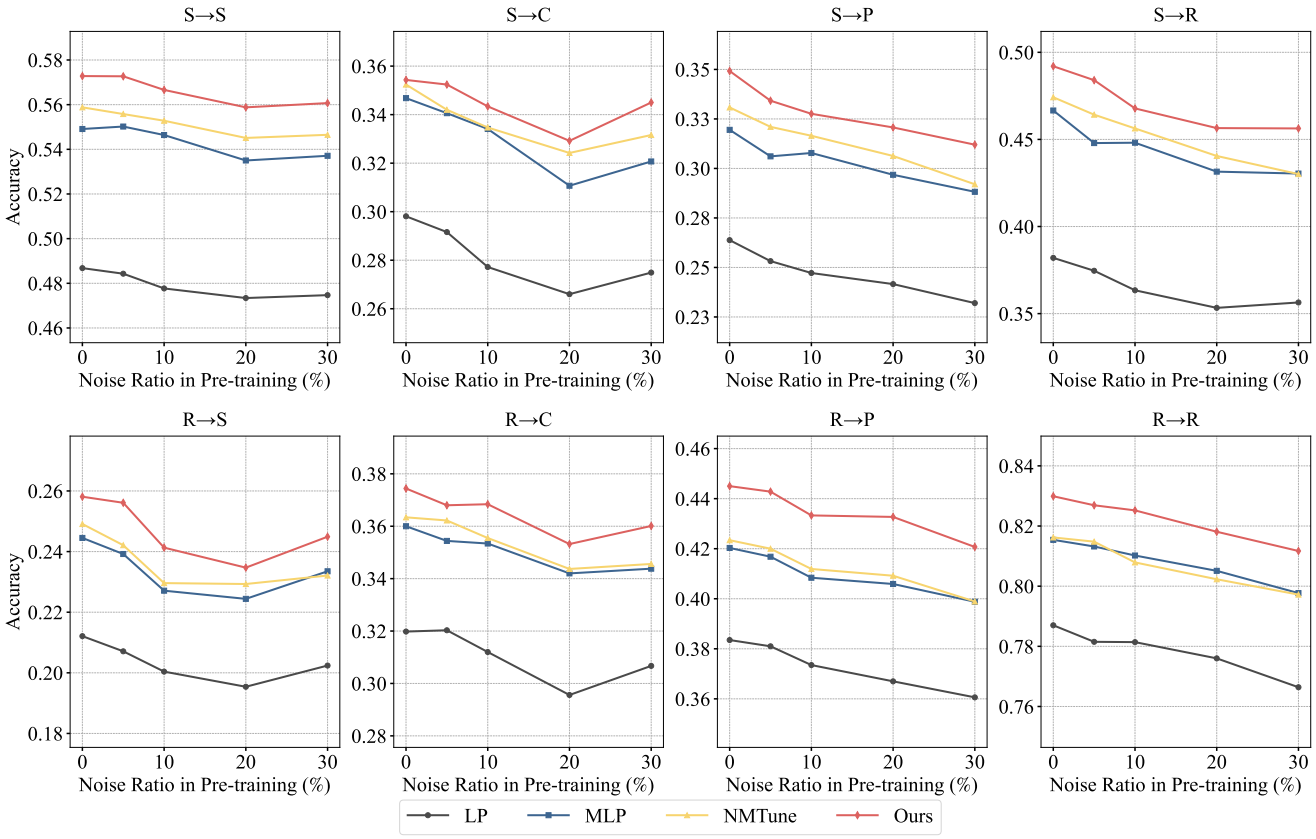


Figure 5. ImageNet-1K pretrained ResNet-50 out-of-domain (OOD) evaluation results

Table 4. Comparison with other fine-tuning methods under different settings. We compare different methods on 4 tasks for out-of-domain (OOD) evaluation. We perform training on DomainNetSketch (S), and evaluate on DomainNetSketch (S), DomainNetReal (R), DomainNetPainting (P), and DomainNetClipart (C) without the training set.

Model	Method	S→S	S→C	S→P	S→R	Avg
RN50/IN-1K-0%	Full FT	0.6533	0.4057	0.2506	0.3062	0.4039
	Ours	0.6665	0.4388	0.2883	0.3355	0.4323
RN50/IN-1K-5%	Full FT	0.6516	0.3996	0.2575	0.3130	0.4054
	Ours	0.6659	0.4380	0.2887	0.3315	0.4310
RN50/IN-1K-20%	Full FT	0.6503	0.3961	0.2502	0.3097	0.4016
	Ours	0.6690	0.4365	0.2843	0.3330	0.4307
Swin-L	LoRA	0.7632	0.6413	0.5004	0.6496	0.6386
	Ours	0.7735	0.6770	0.5556	0.6975	0.6759

ties are summarized in Table 5.

As shown, FPC requires additional class-aware operations, leading to slightly higher complexity compared to FCD and VAR. However, both FCD and VAR are lightweight, with linear dependence on the batch size and feature dimension. Overall, the overhead remains marginal and does not hinder training scalability.

Table 5. Time and memory complexity of different modules. B denotes the batch size, D the feature dimension, and C the number of classes.

Module	Time Complexity	Memory Complexity
FPC	$\mathcal{O}(B \times D) + \mathcal{O}(B \times C)$	$\mathcal{O}(B \times D + B \times C)$
FCD	$\mathcal{O}(B \times D)$	$\mathcal{O}(B \times D)$
VAR	$\mathcal{O}(B \times D)$	$\mathcal{O}(B \times D)$

1.4.4. Hyperparameter Sensitivity Analysis

We further conduct a comprehensive analysis on the sensitivity of loss weights for the three modules in our framework. The evaluation is carried out on OOD tasks using ResNet-50 models pretrained on ImageNet-1K and YFCC15M, under clean, 5% noise, and 20% noise pretraining settings. Training is performed on DomainNetSketch and tested on the remaining domains (DomainNetSketch, DomainNetReal, DomainNetPainting, and DomainNetClipart), excluding the training domain.

The results in Table 6 provide an overview of the combined effects of different weight configurations. In addition, we supplement the analysis with three dedicated exper-

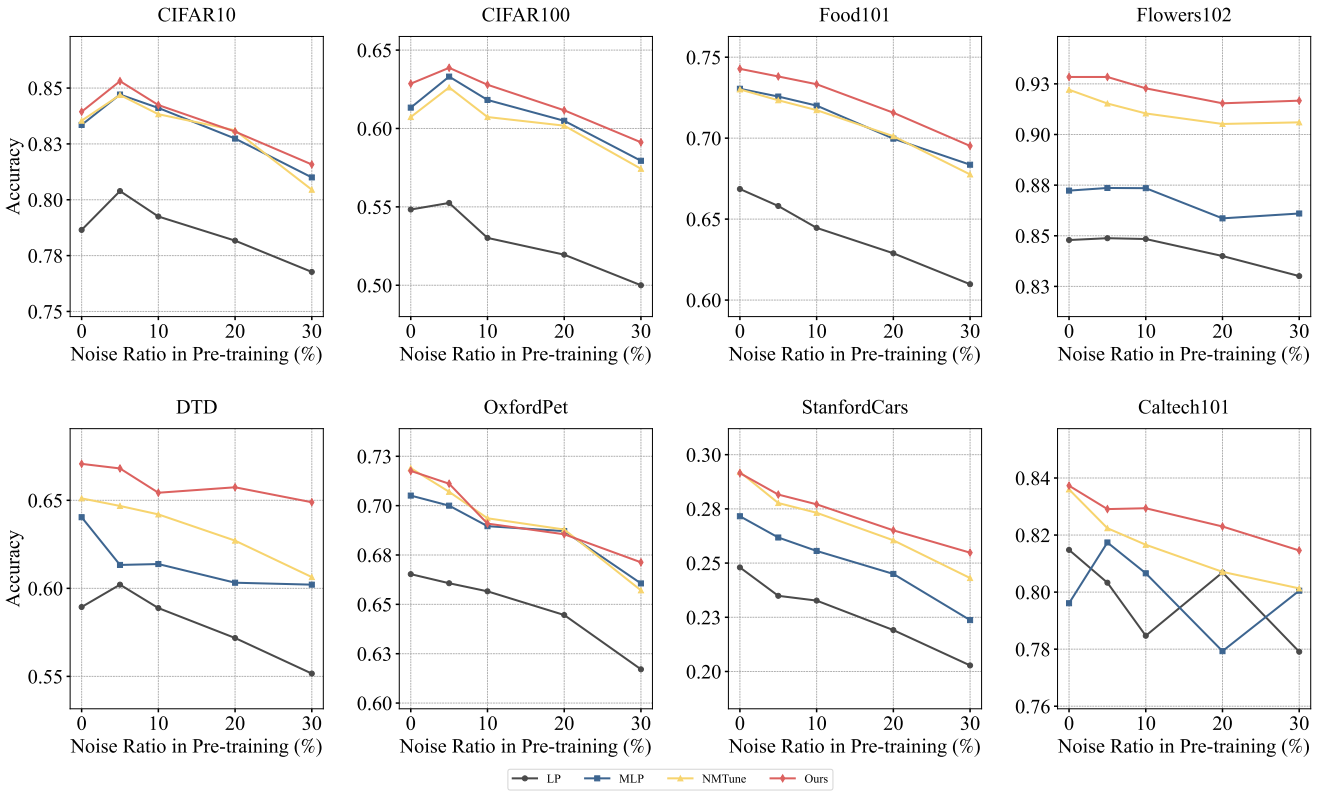


Figure 6. YFCC15M pretrained ResNet-50 in-domain (ID) evaluation results

iments, each varying only one module's loss weight while fixing the others, to better disentangle their individual contributions. The results are summarized in Table 7, Table 8, and Table 9, respectively.

Overall, the findings can be summarized as follows: (i) Larger weights for $\mathcal{L}_{\text{FCD}}$ (e.g., 0.4–0.8) generally yield better results on YFCC15M, showing its stronger role under noisy pretraining. (ii) $\mathcal{L}_{\text{FPC}}$ achieves stable and robust performance within the range 0.03–0.05, particularly in the presence of high pretraining noise. (iii) $\mathcal{L}_{\text{VAR}}$ exhibits relatively minor sensitivity, with smaller values (e.g., 0.0001) often leading to better outcomes, consistent with the intuition that spurious features are less influential, especially in low-noise cases. Taken together, these observations indicate that the three modules provide complementary benefits, and our method maintains strong tolerance to variations in loss weight settings.

Regarding the hyperparameter search process, we adopted a staged strategy. Specifically, we first tuned the weight of $\mathcal{L}_{\text{FPC}}$ independently, and then fixed its optimal value while jointly searching the hyperparameters of $\mathcal{L}_{\text{VAR}}$ and $\mathcal{L}_{\text{FCD}}$. This design was intended to better simulate practical scenarios where multiple loss functions need to operate collaboratively.

It is worth noting that the absolute scales of the three losses differ significantly: $\mathcal{L}_{\text{FCD}}$ is approximately on the order of 10^{-1}, $\mathcal{L}_{\text{FPC}}$ is around 10^{0}, while $\mathcal{L}_{\text{VAR}}$ tends to be the largest, close to 10^{1}. Therefore, the relative magnitude of the assigned weights should not be interpreted as the relative importance of the losses. Instead, these values should be understood in conjunction with the inherent scales of the loss terms.

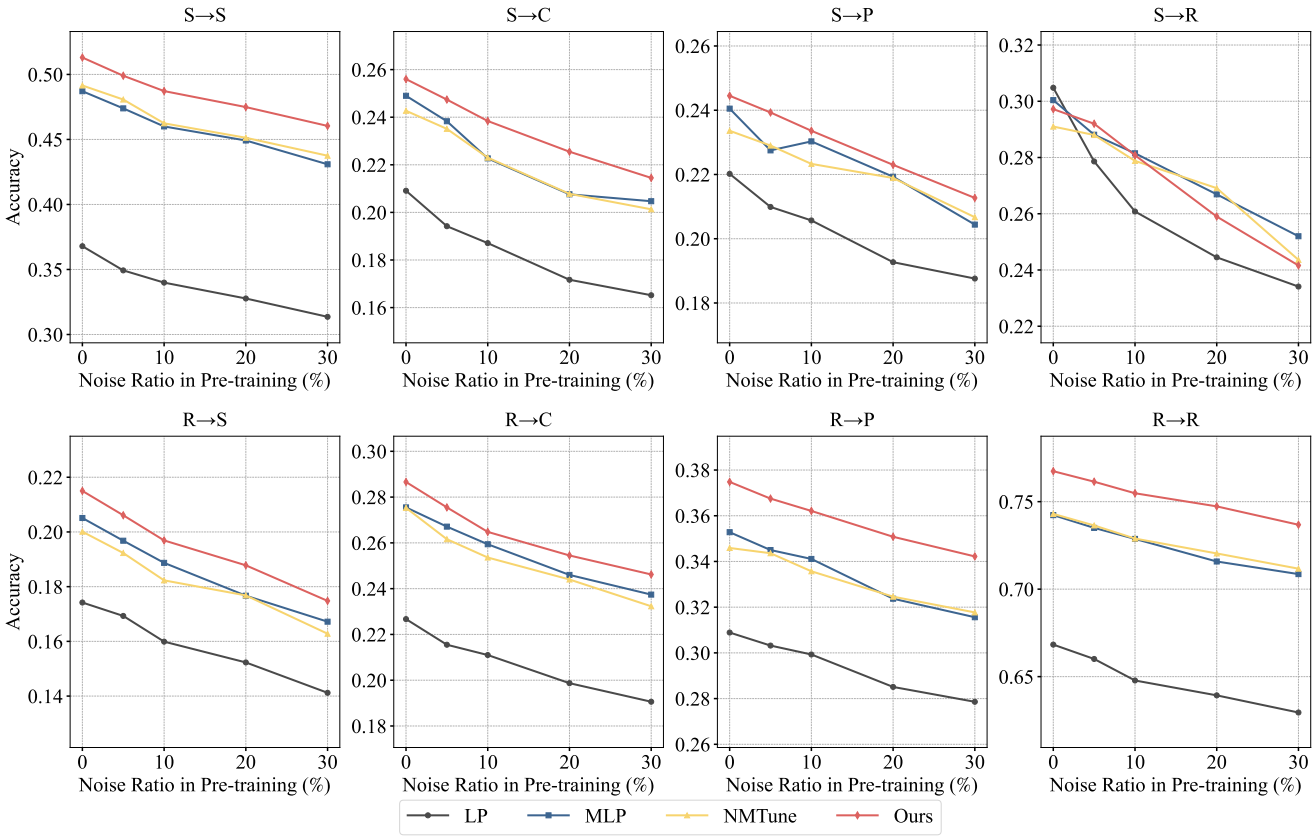


Figure 7. YFCC15M pretrained ResNet-50 out-of-domain (OOD) evaluation results

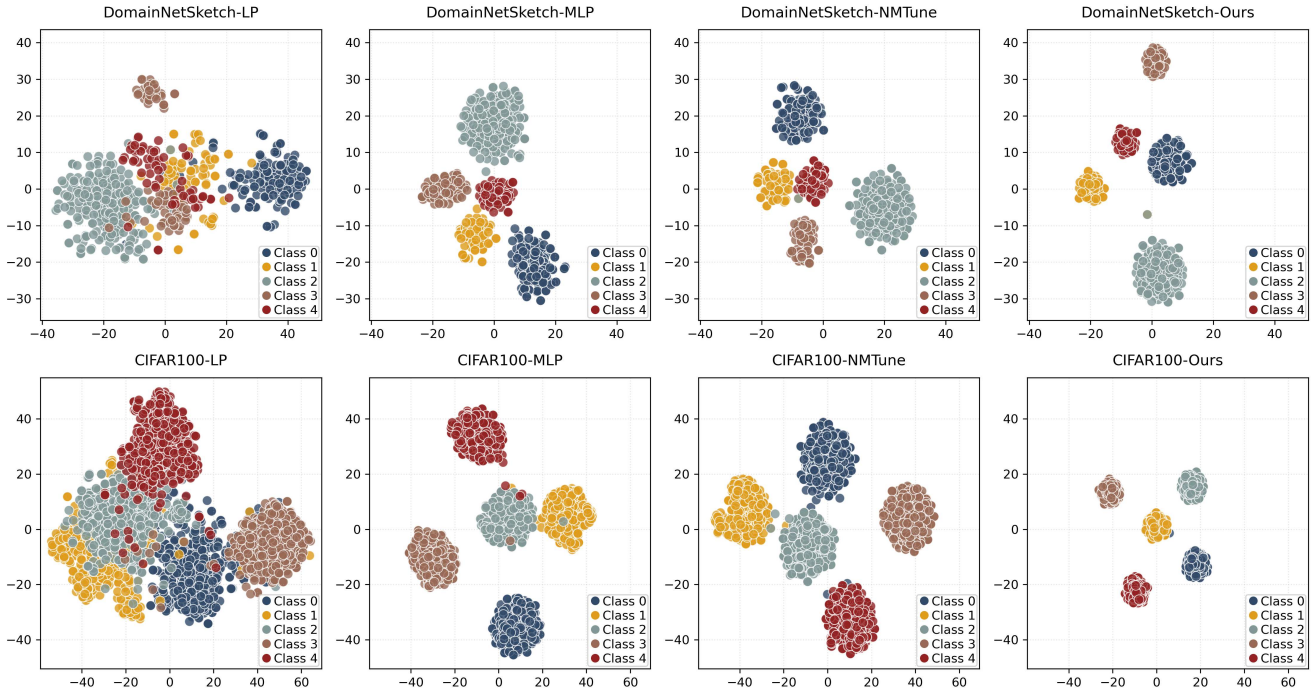


Figure 8. t-SNE visualization comparing the projected features of ResNet-50/ImageNet-1K model ($\gamma = 0\%$) fine-tuned with LP, MLP, NMTune, and the proposed method.

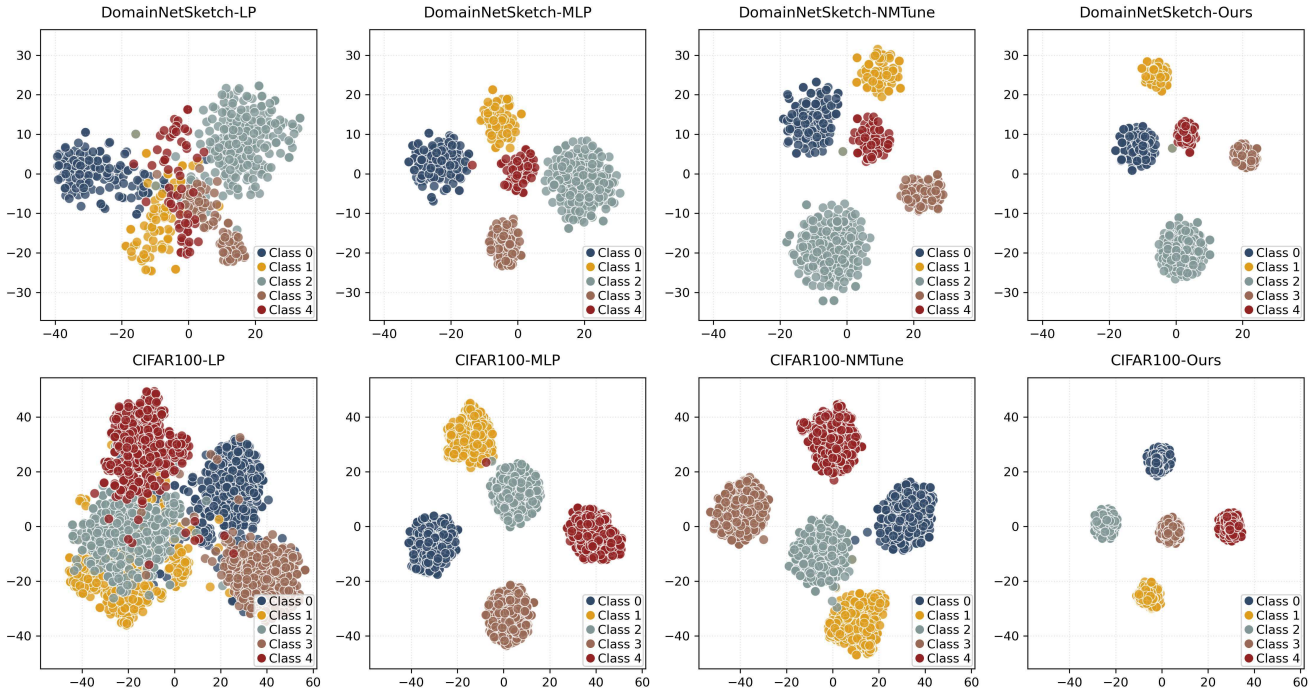


Figure 9. t-SNE visualization comparing the projected features of ResNet-50/ImageNet-1K model ($\gamma = 5\%$) fine-tuned with LP, MLP, NMTune, and the proposed method.

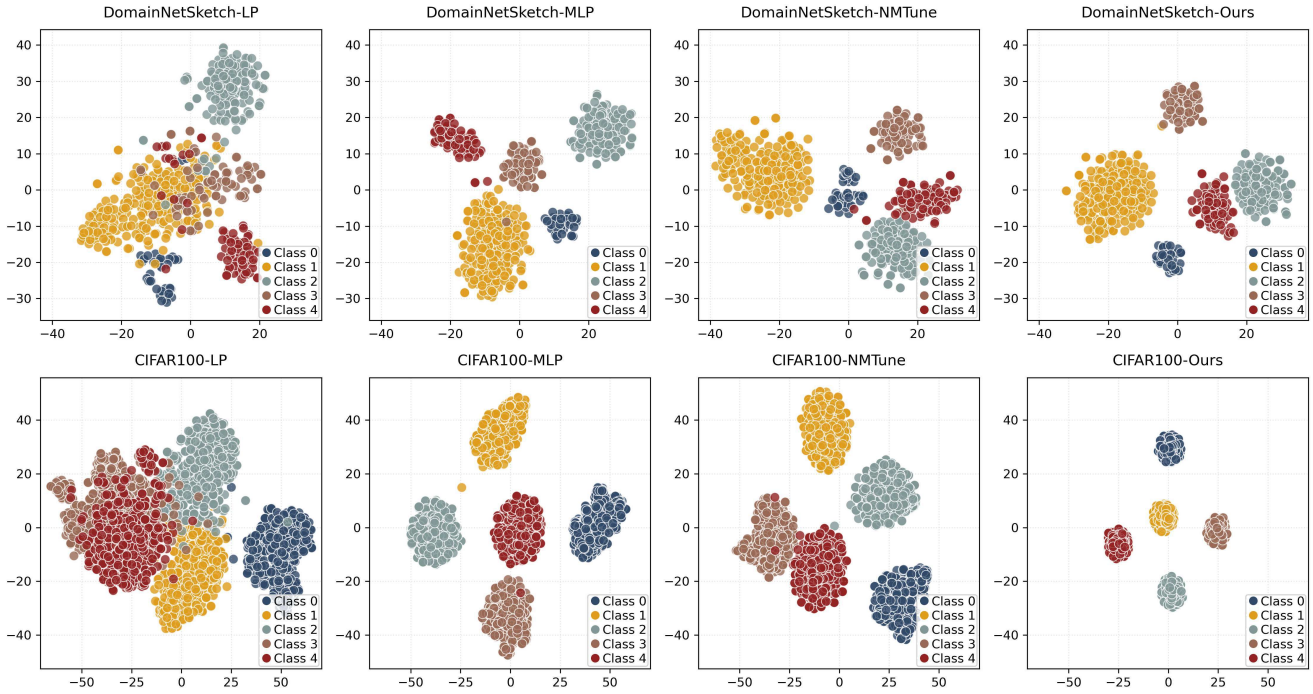


Figure 10. t-SNE visualization comparing the projected features of ResNet-50/ImageNet-1K model ($\gamma = 10\%$) fine-tuned with LP, MLP, NMTune, and the proposed method.

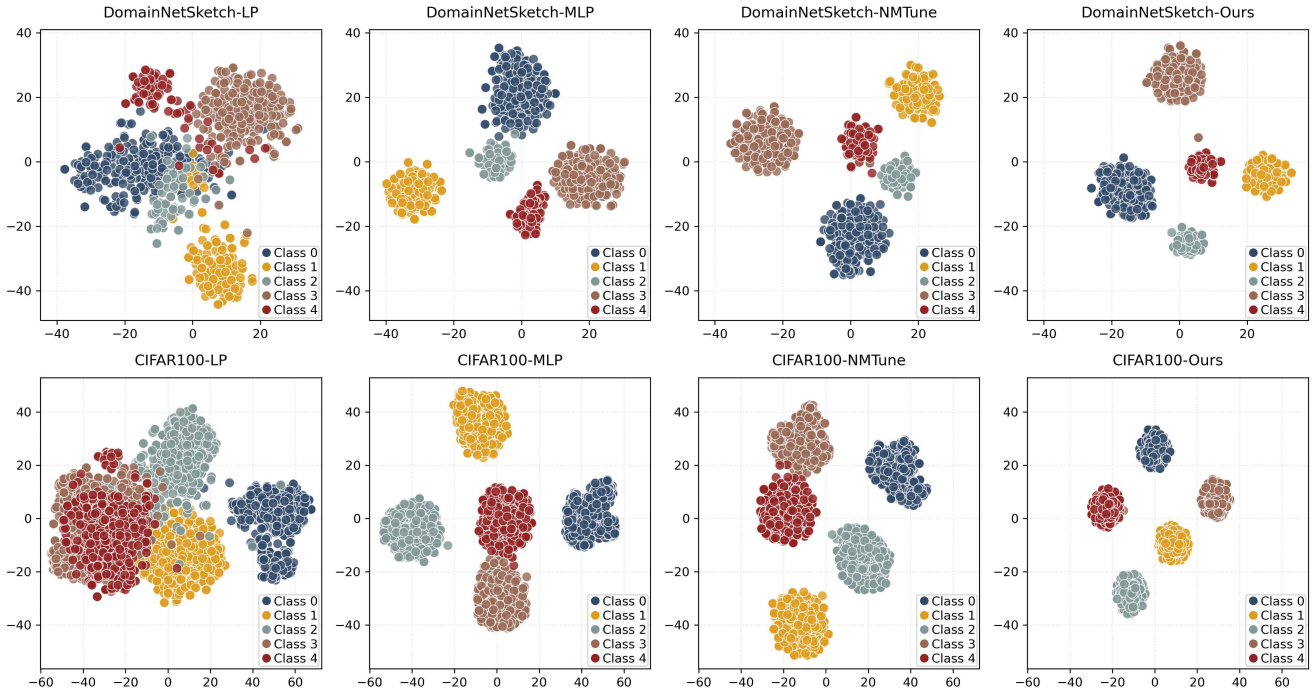


Figure 11. t-SNE visualization comparing the projected features of ResNet-50/ImageNet-1K model ($\gamma = 30\%$) fine-tuned with LP, MLP, NMTune, and the proposed method.

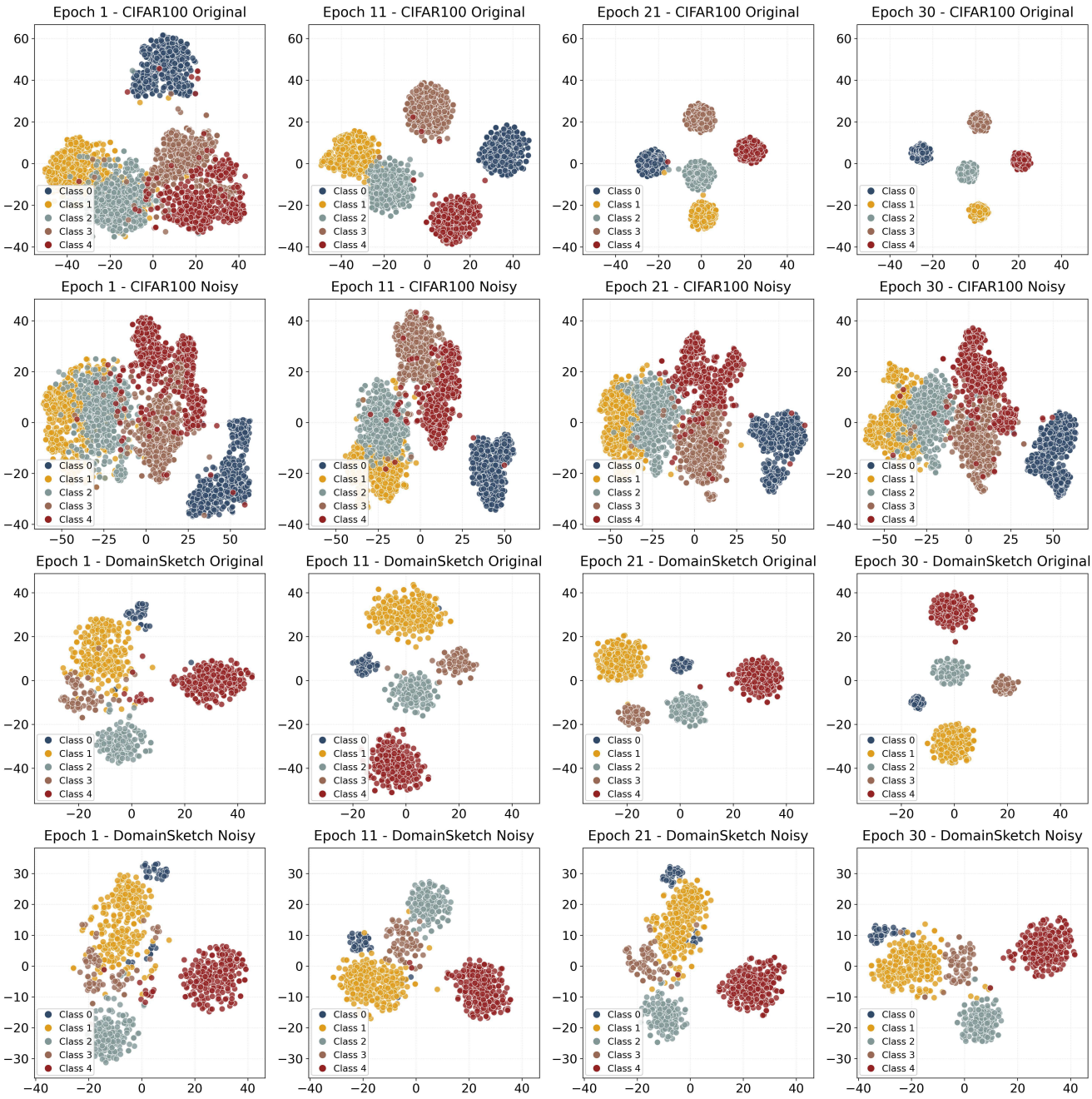


Figure 12. t-SNE trajectories of projected original features (Original) and perturbed features (Noisy) during fine-tuning on DomainNetSketch and CIFAR-100, using ResNet-50/ImageNet-1K ($\gamma = 0\%$).

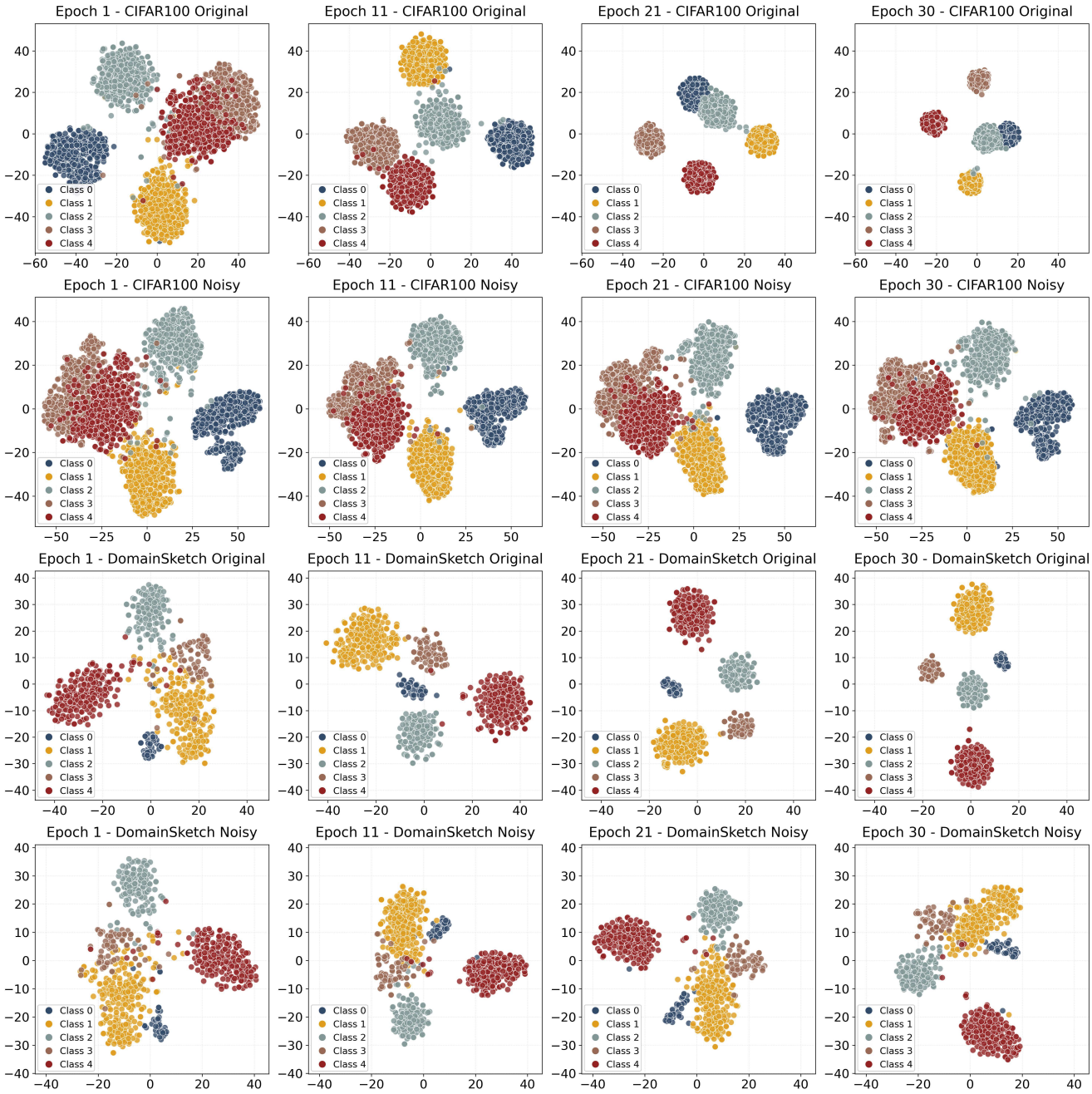


Figure 13. t-SNE trajectories of projected original features (Original) and perturbed features (Noisy) during fine-tuning on DomainNetSketch and CIFAR-100, using ResNet-50/ImageNet-1K ($\gamma = 5\%$).

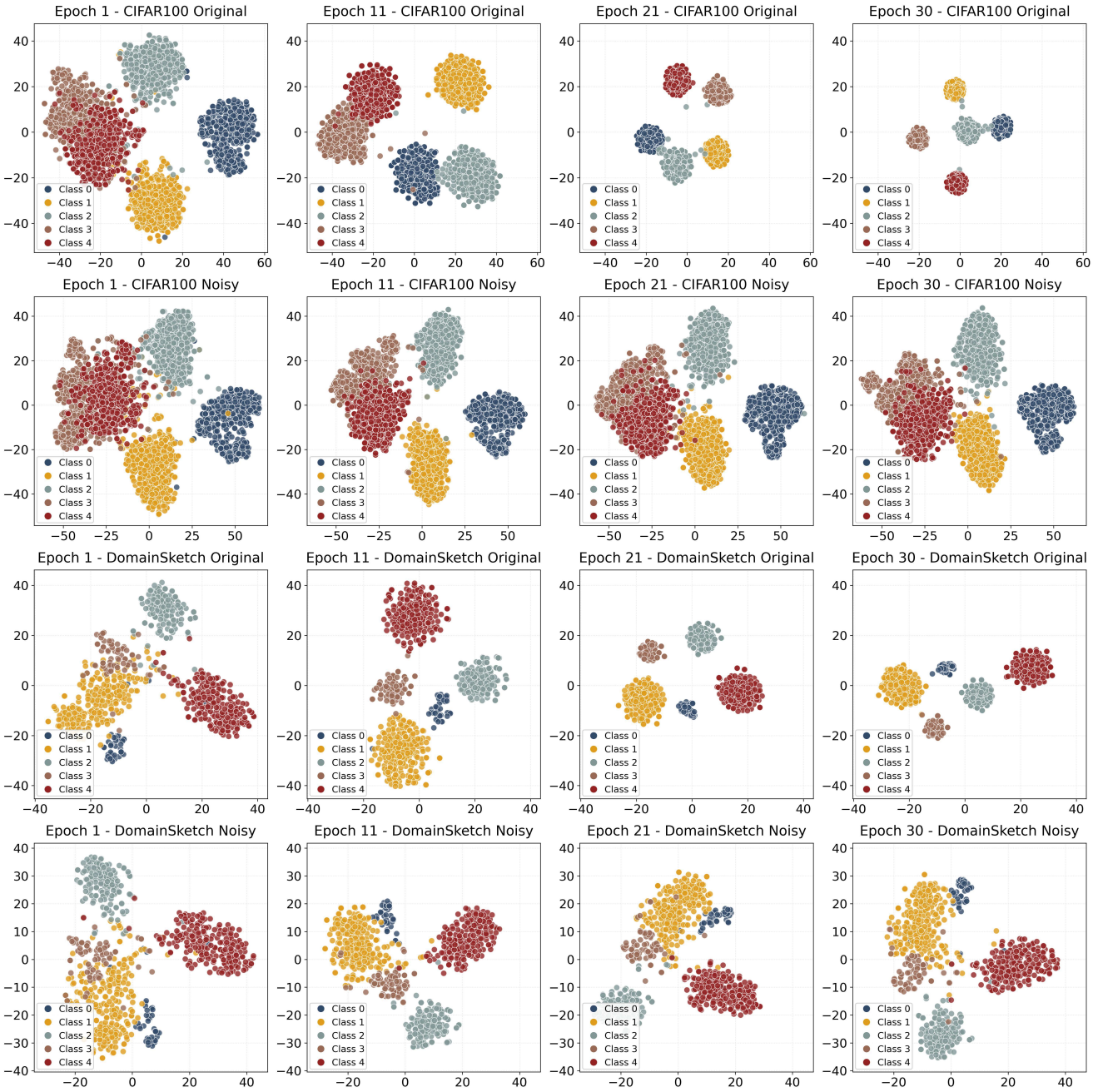


Figure 14. t-SNE trajectories of projected original features (Original) and perturbed features (Noisy) during fine-tuning on DomainNetSketch and CIFAR-100, using ResNet-50/ImageNet-1K ($\gamma = 10\%$).

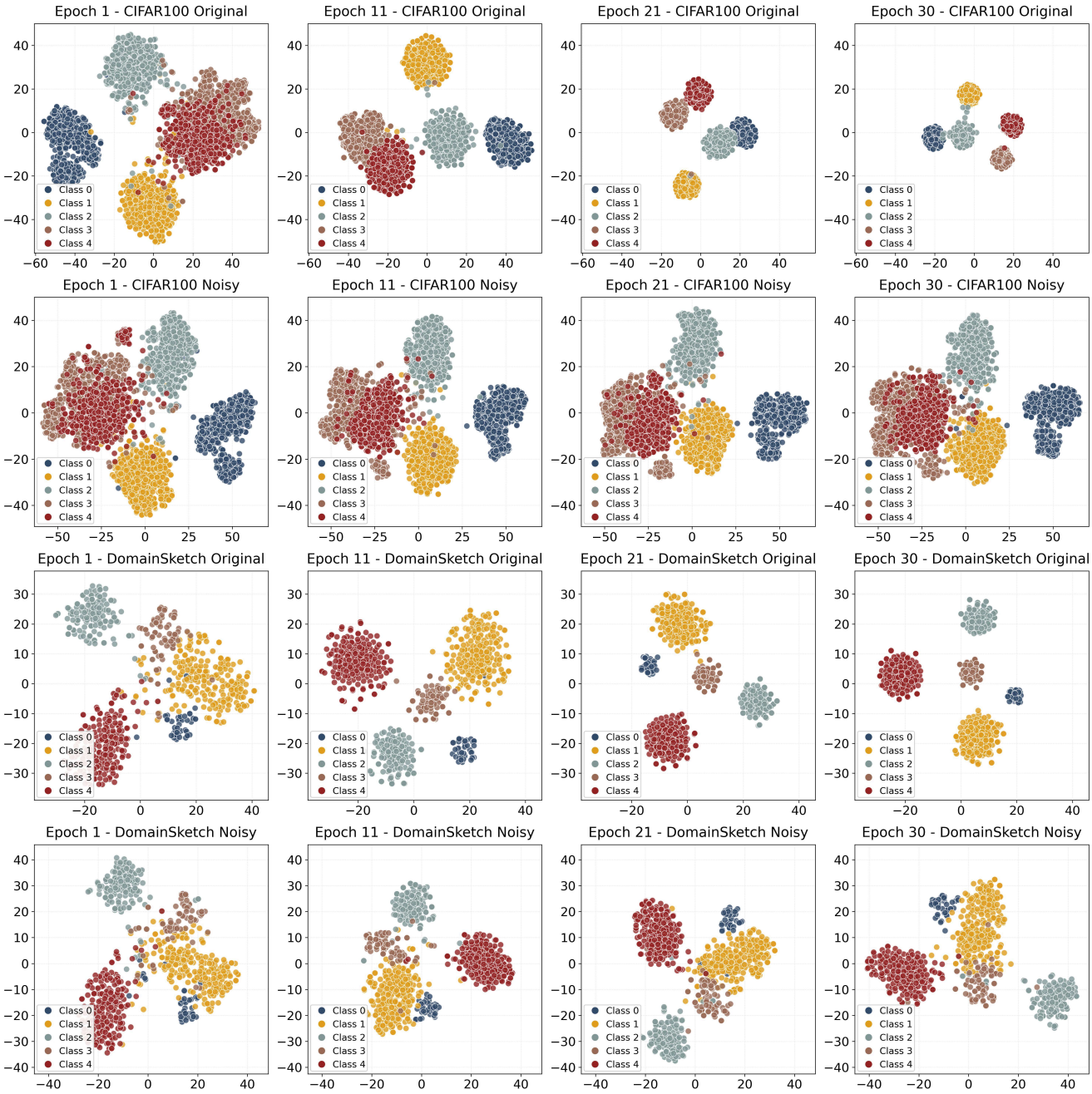


Figure 15. t-SNE trajectories of projected original features (Original) and perturbed features (Noisy) during fine-tuning on DomainNetSketch and CIFAR-100, using ResNet-50/ImageNet-1K ($\gamma = 20\%$).

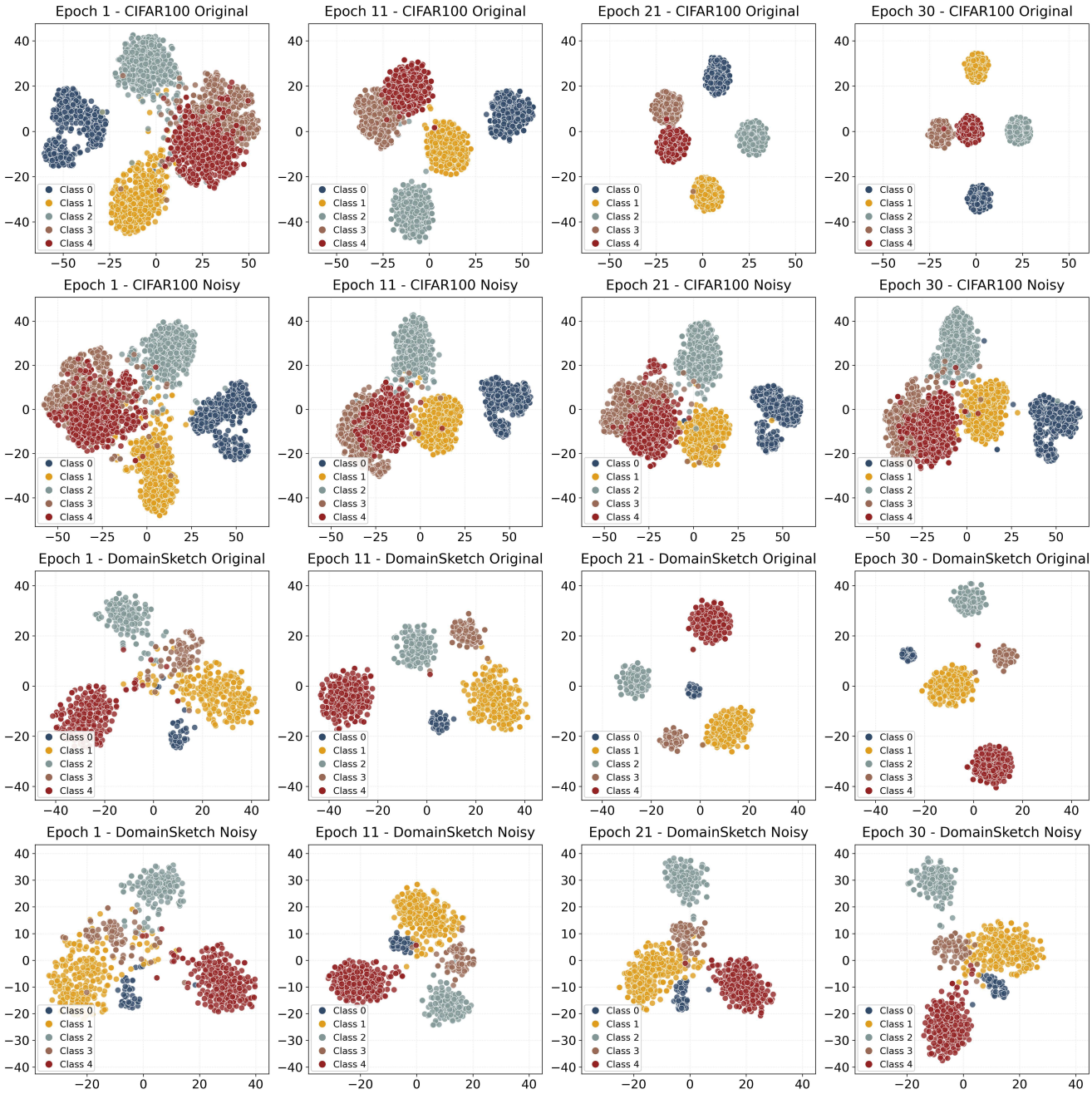


Figure 16. t-SNE trajectories of projected original features (Original) and perturbed features (Noisy) during fine-tuning on DomainNetSketch and CIFAR-100, using ResNet-50/ImageNet-1K ($\gamma = 30\%$).

Table 6. Sensitivity analysis of different loss weights on OOD tasks. Results are reported on ResNet-50 models pretrained on ImageNet-1K and YFCC15M under 0%, 5%, and 20% pretraining noise.

$\mathcal{L}_{\text{FCD}}$	$\mathcal{L}_{\text{FPC}}$	$\mathcal{L}_{\text{VAR}}$	Noise (%)	ImageNet-1K Acc	YFCC15M Acc
0.1	0.01	0.0001	0	0.4367	0.3221
			5	0.4255	0.3087
			20	0.4073	0.2871
0.1	0.05	0.0001	0	0.4391	0.3241
			5	0.4291	0.3141
			20	0.4100	0.2913
0.1	0.03	0.001	0	0.4411	0.3166
			5	0.4345	0.3076
			20	0.4155	0.2879
0.8	0.01	0.0001	0	0.4269	0.3211
			5	0.4241	0.3127
			20	0.4001	0.2900
0.8	0.05	0.0001	0	0.4387	0.3277
			5	0.4319	0.3195
			20	0.4115	0.2956
0.8	0.03	0.001	0	0.4397	0.3181
			5	0.4312	0.3050
			20	0.4122	0.2896
0.4	0.01	0.0001	0	0.4399	0.3179
			5	0.4317	0.3104
			20	0.4133	0.2876
0.4	0.05	0.001	0	0.4386	0.3123
			5	0.4322	0.3055
			20	0.4145	0.2831
0.4	0.03	0.001	0	0.4370	0.3101
			5	0.4299	0.2988
			20	0.4089	0.2865

Table 7. Sensitivity analysis of $\mathcal{L}_{\text{FPC}}$ weights. Results are reported on ResNet-50 models pretrained on ImageNet-1K with 0%, 5%, and 20% pretraining noise, and evaluated on four out-of-domain (OOD) tasks: training on DomainNetSketch (S), and evaluating on DomainNetSketch (S), DomainNetReal (R), DomainNetPainting (P), DomainNetClipart (C) without the training set.

Noise (%)	$\mathcal{L}_{\text{FPC}}$ weight	S→S	S→C	S→P	S→R	Avg
0	0.01	0.5602	0.3507	0.3323	0.4853	0.4321
	0.03	0.5629	0.3516	0.3346	0.4821	0.4328
	0.04	0.5673	0.3512	0.3367	0.4843	0.4349
	0.05	0.5652	0.3523	0.3377	0.4843	0.4349
5	0.01	0.5588	0.3464	0.3207	0.4656	0.4229
	0.03	0.5651	0.3459	0.3273	0.4722	0.4276
	0.04	0.5674	0.3521	0.3262	0.4775	0.4308
	0.05	0.5680	0.3502	0.3337	0.4761	0.4320
20	0.01	0.5463	0.3233	0.3107	0.4460	0.4066
	0.03	0.5474	0.3229	0.3135	0.4471	0.4077
	0.04	0.5495	0.3197	0.3111	0.4525	0.4082
	0.05	0.5496	0.3205	0.3143	0.4517	0.4090

Table 8. Sensitivity analysis of $\mathcal{L}_{\text{VAR}}$ weights. Results are reported on ResNet-50 models pretrained on ImageNet-1K with 0%, 5%, and 20% pretraining noise, and evaluated on four out-of-domain (OOD) tasks: training on DomainNetSketch (S), and evaluating on DomainNetSketch (S), DomainNetReal (R), DomainNetPainting (P), DomainNetClipart (C) without the training set.

Noise (%)	$\mathcal{L}_{\text{VAR}}$ weight	S→S	S→C	S→P	S→R	Avg
0	0.0001	0.5632	0.3504	0.3281	0.4751	0.4292
	0.001	0.5693	0.3507	0.3313	0.4770	0.4321
	0.005	0.5716	0.3469	0.3291	0.4705	0.4295
	0.01	0.5695	0.3492	0.3289	0.4711	0.4297
5	0.0001	0.5599	0.3418	0.3156	0.4581	0.4189
	0.001	0.5666	0.3447	0.3133	0.4573	0.4205
	0.005	0.5713	0.3416	0.3218	0.4570	0.4229
	0.01	0.5671	0.3396	0.3183	0.4537	0.4197
20	0.0001	0.5439	0.3214	0.3056	0.4381	0.4023
	0.001	0.5539	0.3218	0.3027	0.4300	0.4021
	0.005	0.5521	0.3157	0.3005	0.4261	0.3986
	0.01	0.5450	0.3125	0.2902	0.4181	0.3915

Table 9. Sensitivity analysis of $\mathcal{L}_{\text{FCD}}$ weights. Results are reported on ResNet-50 models pretrained on ImageNet-1K with 0%, 5%, and 20% pretraining noise, and evaluated on four out-of-domain (OOD) tasks: training on DomainNetSketch (S), and evaluating on DomainNetSketch (S), DomainNetReal (R), DomainNetPainting (P), DomainNetClipart (C) without the training set.

Noise (%)	$\mathcal{L}_{\text{FCD}}$ weight	S→S	S→C	S→P	S→R	Avg
0	0.1	0.5501	0.3458	0.3163	0.4665	0.4197
	0.4	0.5506	0.3422	0.3186	0.4642	0.4189
	0.8	0.5491	0.3468	0.3170	0.4654	0.4196
	1.0	0.5534	0.3494	0.3150	0.4649	0.4207
5	0.1	0.5488	0.3393	0.3099	0.4510	0.4123
	0.4	0.5479	0.3440	0.3131	0.4509	0.4140
	0.8	0.5518	0.3481	0.3058	0.4546	0.4151
	1.0	0.5510	0.3426	0.3112	0.4586	0.4159
20	0.1	0.5383	0.3176	0.2949	0.4340	0.3962
	0.4	0.5388	0.3175	0.3024	0.4368	0.3989
	0.8	0.5394	0.3185	0.2957	0.4373	0.3977
	1.0	0.5368	0.3199	0.3019	0.4325	0.3978

References

[1] Hao Chen, Jindong Wang, Ankit Shah, Ran Tao, Hongxin Wei, Xing Xie, Masashi Sugiyama, and Bhiksha Raj. Understanding and mitigating the label noise in pre-training on downstream tasks. In *The Twelfth International Conference on Learning Representations*, 2024. 1

[2] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014. 1

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 1

[4] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004. 1

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 1

[6] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1

[7] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *European conference on computer vision*, pages 491–507. Springer, 2020. 1

[8] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 1

[9] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009. 1

[10] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 1

[11] Ziquan Liu, Yi Xu, Yuanhong Xu, Qi Qian, Hao Li, Xiangyang Ji, Antoni Chan, and Rong Jin. Improved fine-tuning by better leveraging pre-training data. *Advances in Neural Information Processing Systems*, 35:32568–32581, 2022. 1

[12] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008. 1

[13] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 1

[14] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019. 1

[15] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 1

[16] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021. 1

[17] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022. 1

[18] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 1

[19] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020. 1