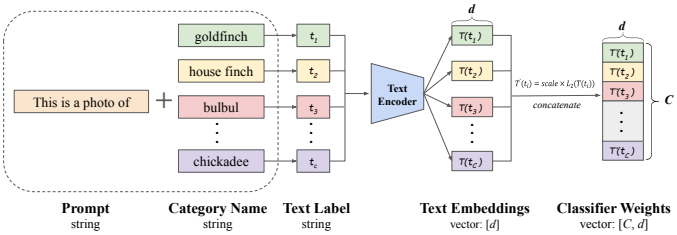


Fig. 1: The pipeline for initializing the classifier with text embeddings. $[d]$: Text embedding dimension. $[C, d]$: Classifier weights dimension.

TABLE I: Training details. LR: Learning rate. T: Temperature.

	CIFAR-10	CIFAR-100	WebVision	Animal-10N	Food-101N
Backbone	CLIP model ViT-L/14@224px				
Resolution	224×224				
th_o	-	-	0.5	-	0.01
th	0.7	0.7	0.7	0.5	0.7
T	1	1	2	2	2
τ	0.95	0.95	0.95	0.95	0.95
λ_u	0.5	0.5	0.5	0.5	0.5
Batch size	Clean set 32, noisy set 16				
Warm up epochs	3	3	3	3	3
Total epochs	20	100	20	20	20
Optimizer	SGD, momentum 0.9, weight decay 5e-4				
Initial LR	5e-4	1e-4	5e-4	5e-4	1e-4
LR decay strategy	a cosine strategy				

APPENDIX

In this section, we first provide details on classifier initialization and training. We then evaluate the effectiveness of OOD detection and clean/noisy sample detection, along with an analysis of the backbone's impact. Next, we conduct a sensitivity analysis of key hyperparameters to assess their influence on model performance. Finally, we discuss the limitations of our work.

A. Classifier Initialization

Inspired by [1], we initialize the classifier using text embeddings, as shown in Fig. 1. For a dataset with C classes, it contains C corresponding text labels. Each text label can be mapped to a feature vector of length d. The classifier's parameters are represented as a matrix of size $[C, d]$, which aligns with the feature dimension. Initializing the classifier with text features effectively utilizes prior knowledge from the texts, enhancing the model's learning performance and accelerating convergence.

B. Detailed Implementations

Our experiments are implemented on two GeForce RTX3090 GPU and PyTorch 1.11.0. We utilize the pretrained CLIP model ViT-L/14@224px [2]. Depending on the noise types present in datasets, a comprehensive three-stage process is implemented for WebVision and Food-101N, while only stage II and III are executed for Animal-10N, CIFAR-10 and CIFAR-100. Due to limitations in our computing resources, the batch sizes for clean and noisy samples are set inconsistently. For Animal-10N, a lower th is used due to its low noise rate. The detailed hyperparameter settings are shown in Tab I.

C. The Effect of OOD Detection

We present examples of ID and OOD images detected from the WebVision dataset using vision-language similarity, as shown in Fig. 2 and Fig. 3. The dataset mainly encompasses categories such as fish, birds and reptiles. It is evident that ID and OOD data exhibit distinct characteristics, and our method effectively separates them. The detection threshold for this experiment is set to 0.5. We usually set a low detection threshold to reduce the risk of misclassifying ID samples as OOD and to prevent the loss of valuable data.

Fig. 2: ID example images from WebVision detected using vision-language similarity distribution (detection threshold: 0.5).

D. The Effect of Clean/Noisy Sample Detection

In this section, we evaluate the efficacy of our method in detecting clean samples from both quantitative and qualitative perspectives. The quantitative analysis is conducted on synthetic noise datasets, CIFAR-10 and CIFAR-100, while the qualitative analysis is performed on real-world noisy dataset, WebVision.

Quantitative results. We test the classification accuracy for both clean and noisy samples, as well as the recall and precision for noisy samples, on CIFAR-10 and CIFAR-100 datasets with different types and levels of synthetic label noise. The results are shown in Tab II. We uniformly set the detection threshold to 0.7 across all scenarios. It is observed that the recall rate for noisy samples is nearly 100%, indicating that the clean dataset we screened contains virtually no noisy labels. At low noise ratios, noise detection precision is low because the detection threshold of 0.7 is too high, causing clean samples to be misclassified as noisy. This impact is minimal as the misclassified clean samples can also be learned in Stage III. In practice, we usually set a high detection threshold to improve

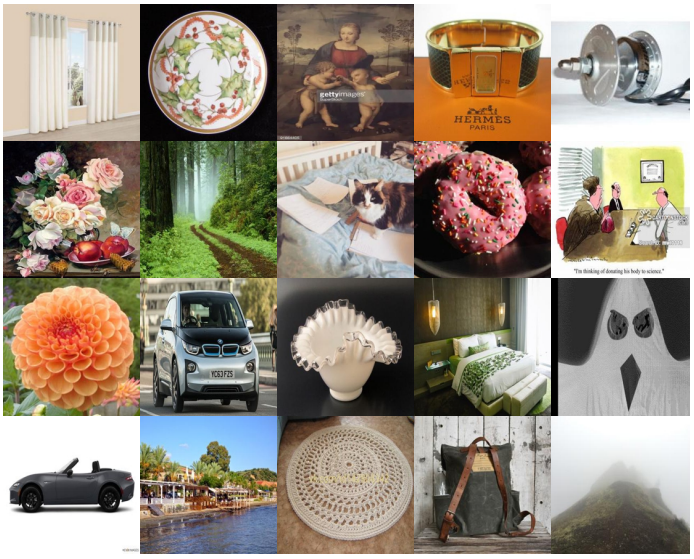


Fig. 3: OOD example images from WebVision detected using vision-language similarity distribution (detection threshold: 0.5).

TABLE II: Clean and noisy samples detection performance on CIFAR-10/100 dataset(detection threshold: 0.7). Acc: Detection Accuracy for Clean and Noisy Samples; Noisy Recall: Recall for Noisy Samples; Noisy Precision: Precision for Noisy Samples.

Datasets	Metric/Noise ratio	Sym.				Asym.
		20%	50%	80%	90%	40%
CIFAR-10	Acc	96.3	97.4	98.7	99.0	96.4
	Noisy Recall	99.9	99.8	99.8	99.9	99.0
	Noisy Precision	82.8	94.6	98.3	98.9	85.3
CIFAR-100	Acc	82.4	88.9	95.4	97.5	-
	Noisy Recall	99.9	99.9	99.9	99.9	-
	Noisy Precision	53.0	81.8	94.6	97.4	-

the precision for clean samples. Although the high threshold may mislabel some clean samples as noisy, these samples are reused in later stages, preserving their valuable information.

Qualitative results. We select four images labeled as "indigo bundling" from the WebVision dataset and compute their corresponding scores. The effects of data augmentations and the associated model predictions are shown in Fig. 4.

We categorize the outcomes into four primary scenarios as follows: For simple clean sample depicted in Fig. 4 (a), the model demonstrates high confidence in its predictions, which are consistent across various data augmentation effects and align with the given labels. In contrast, for challenging clean sample shown in Fig. 4 (b), the model's predictions vary based on the results of data augmentation, showing low confidence. Simple noisy sample, as illustrated in Fig. 4 (c), receive high-confidence predictions from the model, but these predictions often differ from the labels. Lastly, for hard noisy sample presented in Fig. 4 (d), the model's confidence is notably low, and the majority of its predictions do not correspond with the labels.

TABLE III: Ablation experiment results on WebVision for backbone effect.

Method	Backbone	WebVision		ImageNet	
DivideMix [3]	Inception-ResNetV2	77.32	91.64	75.20	90.84
	CLIP-ViT	83.92	95.92	80.96	94.92
UNICON [4]	Inception-ResNetV2	77.60	93.44	75.29	93.72
	CLIP-ViT	88.72	96.32	85.92	95.48
DISC [5]	Inception-ResNetV2	80.28	92.28	77.44	92.28
	CLIP-ViT	87.96	98.00	85.96	98.12
Ours	CLIP-ViT	88.84	98.40	88.12	98.84

E. The Influence of Backbone

To comprehensively validate the effectiveness of our method, we reproduce SOTA methods and replace the previously pretrained Inception-ResNetV2 with the pretrained ViT model from CLIP. The results are presented in Tab III.

It is evident that the features of CLIP exhibit strong robustness, contributing to performance improvements across various methods. Despite this, our method still achieves the best performance. While UNICON's top-1 accuracy on WebVision is comparable to ours, it relies on a dual-branch network with twice the number of model parameters. Moreover, UNICON's generalization performance on ImageNet is significantly lower than ours (85.96% vs. 88.12%), indicating its relatively weaker robustness. These experiments suggest that the advantages of our method stem from two key factors: the robust feature representations provided by the large foundational model and the improved handling of OOD samples and fine-grained noise.

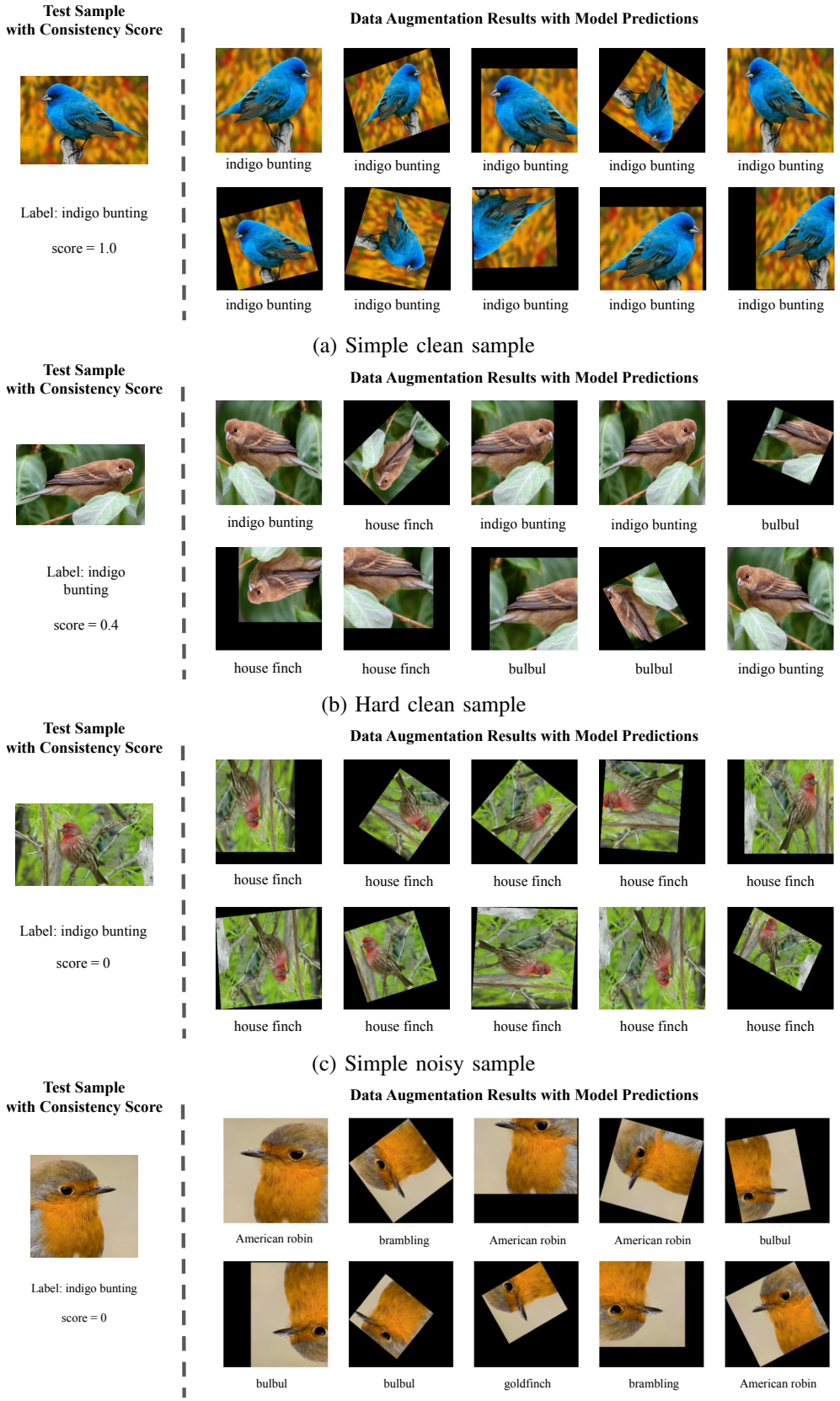
F. Sensitivity Study of Hyperparameters

Our method contains four main hyperparameters. th_o (in Sec. 3.B) and th (in Eq. 3) are thresholds that help detect OOD data and clean data, respectively. τ (in Eq. 6) is used to select high-confidence pseudo labels for noisy samples during semi-supervised training, while the weight of these noisy samples in the loss function is controlled by λ_u (in Eq. 7). We conduct sensitivity analysis for these parameters on real world datasets (WebVision and ILSVRC12). The results are shown in Fig. 5.

For OOD detection, a low th_o may fail to filter out OOD samples, while a high th_o may mistakenly remove useful ID data. The threshold th_o is robust around 0.25 and 0.5 within [0, 1]. Extremely low or high values degrade model performance.

The threshold th is used to detect clean samples. A higher th increases the precision of clean samples detection by making classification stricter. It is robust over the broad range of 0.3 to 0.7. However, when th is set to 0.9, accuracy drops sharply. In this experiment, τ is fixed at 0.95. When both th and τ are high, fewer clean and noisy samples are used for training, reducing learnable information and degrading performance.

The threshold τ balances the number of noisy samples used in training and the quality of pseudo-labels. A smaller value may introduce incorrect pseudo-labels, while a larger one may exclude too many useful samples. We evaluate τ for four values ranging from 0.9 to 1. Fig. 5 shows that our choice 0.95 is a reasonable option.



(a) Simple clean sample

(b) Hard clean sample

(c) Simple noisy sample

(d) Hard noisy sample

Fig. 4: Visualization of consistency score. Left: Test image with its label and consistency score. Right: Data augmentation effects on the test sample and corresponding model predictions.

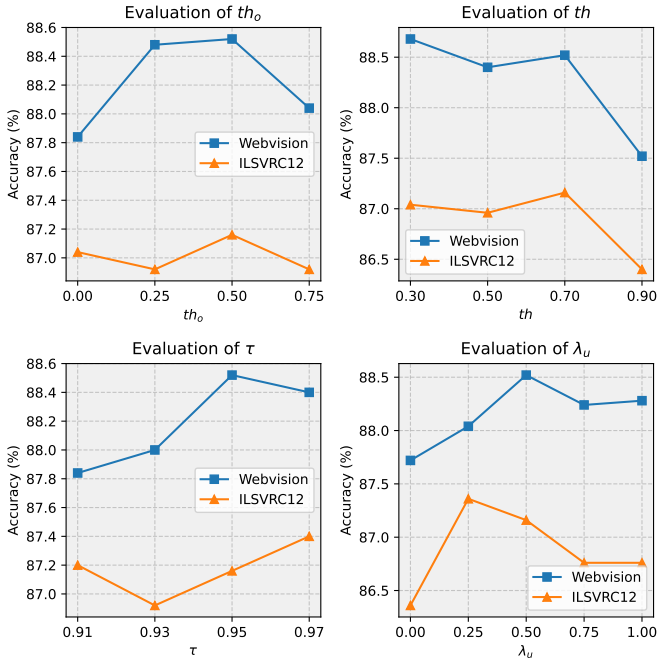


Fig. 5: Sensitivity study results for th_o, th, τ, and λ_u on WebVision and ILSVRC12 datasets. We report the top-1 accuracy.

The weight λ_u adjusts the contribution of noisy samples to training. A small λ_u underutilizes them, while a large one overemphasizes them, both harming performance. λ_u are also robust around values 0.25 and 0.5 among the interval [0, 1].

G. Limitations.

Our work still has certain limitations, including: 1) Our method only uses text labels. Richer textual information such as category description can further improve the performance. 2) The reliance on pretrained models could limit its applicability in domains with insufficient pretraining data. 3) Several hyperparameters in use, while providing flexibility, complicates deployment in new environments. Future work will explore leveraging category descriptions or image captions generated by open-source large models, as well as merging Stage II and Stage III, to mitigate these limitations.

REFERENCES

[1] Mitchell Wortsman et al., "Robust fine-tuning of zero-shot models," in *CVPR*, 2022, pp. 7959–7971.

[2] Alec Radford et al., "Learning transferable visual models from natural language supervision," in *ICML*. PMLR, 2021, pp. 8748–8763.

[3] Junnan Li, Richard Socher, and Steven CH Hoi, "Dividemix: Learning with noisy labels as semi-supervised learning," *arXiv preprint arXiv:2002.07394*, 2020.

[4] Nazmul Karim, Mamshad Nayeem Rizve, Nazanin Rahnavard, Ajmal Mian, and Mubarak Shah, "Unicon: Combating label noise through uniform selection and contrastive learning," in *CVPR*, 2022, pp. 9676–9686.

[5] Yifan Li, Hu Han, Shiguang Shan, and Xilin Chen, "Disc: Learning from noisy labels via dynamic instance-specific selection and correction," in *CVPR*, 2023, pp. 24070–24079.