

TECHNICAL APPENDICES AND SUPPLEMENTARY MATERIAL

A DICTIONARY OF SYMBOLS

Table 8: Dictionary of Symbols Used in Problem Statement (Sec. 2) and Approach (Sec. 3)

Symbol	Description
\tilde{D}	Noisy training dataset, $\tilde{D} = \{(x_i, \tilde{y}_i)\}_{i=1}^N$
D^*	Ideal clean dataset with distributional labels, $D^* = \{(x_i, y_i^*)\}$
D_{ref}	Manually verified small clean reference set
D_{ref}^*	Augmented reference set derived from D_{ref}
\hat{D}	Curated dataset with refined soft labels
$x_i \in \mathbb{R}^d$	Input sample in d -dimensional space
$\tilde{y}_i \in \{0, 1\}^C$	Noisy one-hot label for sample x_i
$y_i^* \in [0, 1]^C$	Ground-truth soft label (class distribution), $\sum_{c=1}^C y_{ic}^* = 1$
C	Number of classes
$f_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^h$	Feature extractor / encoder with parameters θ
$g_\phi : \mathbb{R}^h \rightarrow \mathbb{R}^C$	Classification head with parameters ϕ
$\hat{\theta}$	Trained model parameters from empirical risk minimization
θ^*	Optimal model parameters minimizing true risk
$\mathcal{L}(\cdot, \cdot)$	Loss function (e.g., cross-entropy)
$\lambda \ f\ $	Regularization term (e.g., weight decay)
$k(x_i, x_j)$	Similarity kernel (e.g., cosine similarity) between samples x_i and x_j
$\hat{y}(x)$	Refined soft label of x computed via voting from reference samples
$Z(x)$	Normalization constant to ensure $\hat{y}(x)$ is a valid probability distribution
τ	Cosine similarity threshold for voting pool inclusion
k	Number of neighbors selected for diverse reference voting
$D_{\text{vote}}(x)$	Set of reference samples with similarity $\geq \tau$ to x
$D_{\text{vote}}^*(x)$	k -diverse subset of $D_{\text{vote}}(x)$ selected via max-diversity
$t \in (0, 1]$	Temperature parameter for sharpening predicted label distribution
$\text{IF}(s_i, s_{\text{ref}})$	Influence of sample s_i on s_{ref}
M_i	Subset of D_{ref} with same label as \tilde{y}_i
δ_{IF}	Threshold for influence to include sample in D_{ref}^*
$\hat{y}^*(x)$	Sharpened label distribution: $\hat{y}^*(x) = \hat{y}(x)^{1/t} / \sum_c \hat{y}_c(x)^{1/t}$

B COMPUTATIONAL COST ANALYSIS

Training Efficiency. One of the common concerns when introducing a multi-phase training framework is the potential computational overhead. In this section, we provide a detailed breakdown of the time cost of our method TRAINREF, and compare it with the top-performing baselines under the same hardware setting—specifically, a single NVIDIA GeForce RTX 4090 GPU.

Phase-wise Training Time. As shown in Table 9, TRAINREF comprises three phases: (1) a self-supervised pretraining phase, (2) an influence-based reference augmentation phase, and (3) a reference-guided co-evolution phase.

In **Phase I**, we apply Masked Image Modeling (MIM) using BEiT_{v2} to learn a robust and generalizable embedding space. The tokenization mechanism in BEiT_{v2} enables efficient training, with each MIM epoch taking only 3 minutes. We pretrain the model for 300 epochs in this stage.

In **Phase II**, we apply influence functions to augment the small trusted reference set, identifying clean samples from the noisy dataset. The model is then fine-tuned on this augmented reference

set to enhance its classification capability. This stage requires 5 fine-tuning epochs, each taking approximately 18 minutes.

In **Phase III**, we iteratively co-evolve the model and the dataset through reference-guided curation and distributional supervision. Specifically, the model refines its predictions using neighborhood voting from the reference set, while the curated dataset is simultaneously updated to reflect these refined soft labels. This iterative process ensures that both the embedding space and label quality improve progressively. The finetuning process involves 10 epochs (when $N = 2$), each taking around 18 minutes. During this stage, standard data augmentation techniques such as MixUp are applied. Thanks to the high-quality initialization from Phase I, only a small number of finetuning epochs are sufficient to achieve strong performance.

Despite incorporating a self-supervised pretraining stage, the overall runtime of TRAINREF remains comparable to the fastest baselines, demonstrating its practical efficiency.

Overall Runtime. As summarized in Table 9, the total training time of TRAINREF is approximately 1470 minutes, which is only marginally higher than DISC (1400 minutes), the most efficient baseline among state-of-the-art methods. Despite including a self-supervised pretraining stage, our approach remains competitive in terms of wall-clock time due to (i) the efficiency of BEiT-based MIM and (ii) the reduced number of fine-tuning epochs required.

Figure 4 further illustrates the per-epoch training time across various baselines. Notably, the runtime of TRAINREF per epoch during finetuning is comparable to that of LSL and CC. These results collectively show that TRAINREF achieves a favorable trade-off between computational cost and performance.

Table 9: Training Time Comparison on WebVision (RTX 4090)

Method	Time per Epoch (min)	Training Epochs	Total Time (min)
CC	23 / epoch	—	—
DISC	14 / epoch	100	1400
LSL	22 / epoch	100	2200
Ours	3 (MIM), 18 (FT)	300 + 15	1470

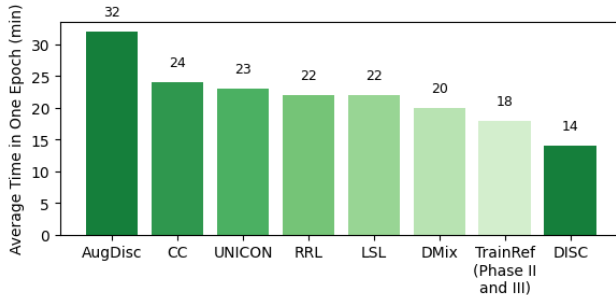


Figure 4: Training time per epoch (in minutes) across different methods. TRAINREF has a comparable finetuning cost to LSL and CC, and an efficient pretraining stage with BEiT.

C IMPLEMENTATION DETAILS

C.1 MODEL ARCHITECTURE AND PRETRAINING SETUP

In Phase I of TRAINREF, we adopt the BEiT_{v2} pipeline to perform self-supervised masked image modeling (MIM). The encoder is a Vision Transformer (ViT) trained from scratch on each target dataset. The model consists of 12 transformer blocks, each with 12 attention heads and a hidden dimension of 768. The patch size is set to 16×16 , and input images are resized to 224×224 . We use a tokenizer pretrained on ImageNet-1K, which yields 8,192 discrete visual tokens. This setup enables efficient and semantically rich representation learning, which is essential for robust downstream curation.

For MIM training, we use the AdamW optimizer with a weight decay of 0.05 and cosine annealing learning rate schedule initialized at 1×10^{-3} . A warm-up phase is applied over the first 10,000

iterations. To prevent overfitting, we use stochastic depth regularization with a drop path rate of 0.1, and stabilize optimization through layer-wise learning rate decay. Training is conducted for 200 epochs on CIFAR-100, CIFAR-80N, Animal10N, and WebVision.

C.2 INFLUENCE-BASED REFERENCE AUGMENTATION

After pretraining, we perform linear probing to prepare for influence function analysis. A randomly initialized linear classification head is attached to the frozen encoder and trained for 15 epochs on the clean reference set. Model parameters are saved every 5 epochs to support multi-checkpoint influence estimation.

We compute the influence score of each training sample relative to reference samples using gradient similarity. Samples whose normalized influence scores exceed the threshold $\delta_{\text{IF}} = 0.8$ are selected for inclusion in the augmented reference set. By default, we initialize the reference set with 10 clean samples per class. The size and quality of this set are further analyzed in our ablation studies.

C.3 FINETUNING AND ITERATIVE CO-EVOLUTION

In Phase II, we fine-tune the model on the augmented reference set to specialize the embedding space. We use the Adam optimizer with cosine decay, a learning rate of 1×10^{-4} , and train for 5 epochs. RandAugment is applied with parameters ($n = 2, m = 10$) to enhance generalization, and MixUp regularization is incorporated using an interpolation coefficient of $\alpha = 0.4$.

Phase III involves two rounds of iterative co-evolution between the model and the dataset. Each iteration lasts for 5 epochs and follows the same optimization and augmentation settings as in Phase II. In each round, refined label distributions are generated via reference-guided voting, and the model is retrained on the newly curated dataset. This procedure ensures that both the embedding function and the pseudo-labels are progressively improved.

C.4 REPRODUCIBILITY

All experiments are implemented in PyTorch and conducted on a single NVIDIA RTX 4090 GPU. Unless otherwise specified, we use a batch size of 128. Detailed training logs, configuration files, and checkpoints will be made publicly available in the project repository.

This three-phase design enables TRAINREF to efficiently extract semantically aligned embeddings, construct high-quality reference sets, and iteratively refine soft labels, ultimately yielding a robust model trained under extreme label noise.

D EXTENDED RESULTS ON CONFIDENCE RELIABILITY

To complement the main results in Section 4.2, we provide extended evaluations on additional CIFAR-100 noise regimes (symmetric 20%, 80%, asymmetric 40%) in Table 10, 11 and 12 and on two real-world datasets (WebVision, Animal-10N) in Table 13. We also report multiple calibration metrics, including ECE, AdaECE (Mukhoti et al., 2020a), $\text{ECE}_{\text{debias}}$, and $\text{ECE}_{\text{sweep}}$ (Roelofs et al., 2022).

Table 10: Detailed results on CIFAR-100 with symmetric 20% noise.

Method	Test Acc (%)	ECE (\downarrow)	AdaECE (\downarrow)	$\text{ECE}_{\text{debias}}$ (\downarrow)	$\text{ECE}_{\text{sweep}}$ (\downarrow)
CE	51.76	0.0880	0.0879	0.0878	0.0880
Focal Loss	52.16	0.1199	0.1198	0.1197	0.1201
Ada Focal Loss	51.69	0.0923	0.0913	0.0921	0.0918
Dual Focal Loss	47.32	0.1476	0.1476	0.1474	0.1476
CE+TS	51.76	0.0137	0.0138	0.0130	0.0147
CE+PTS	51.76	0.0263	0.0280	0.0261	0.0261
CE+Spline	51.76	0.0242	0.0280	0.0240	0.0240
CE+MnM	51.76	0.0177	0.0126	0.0168	0.0153
DISC	78.75 \pm 0.13	0.118 \pm 0.011	0.114 \pm 0.016	0.117 \pm 0.009	0.118 \pm 0.013
DISC+TS	78.75 \pm 0.13	0.043 \pm 0.005	0.045 \pm 0.010	0.041 \pm 0.011	0.051 \pm 0.013
L2B	79.67 \pm 0.14	0.103 \pm 0.013	0.112 \pm 0.009	0.108 \pm 0.021	0.117 \pm 0.016
L2B+TS	79.67 \pm 0.14	0.042 \pm 0.012	0.043 \pm 0.011	0.043 \pm 0.009	0.045 \pm 0.012
Ours	85.44\pm0.21	0.048\pm0.009	0.047\pm0.008	0.044\pm0.009	0.052\pm0.014
Ours+TS	85.44\pm0.21	0.015 \pm 0.009	0.016 \pm 0.006	0.012\pm0.005	0.016 \pm 0.006

Discussion. These results confirm that TrainRef consistently outperforms state-of-the-art train-time, post-hoc, and denoising methods in both synthetic and real-world noise scenarios. Its superior

Table 11: Detailed results on CIFAR-100 with symmetric 80% noise.

Method	Test Acc (%)	ECE (\downarrow)	AdaECE (\downarrow)	ECE _{debias} (\downarrow)	ECE _{sweep} (\downarrow)
CE	16.38	0.0946	0.0946	0.0945	0.0945
Focal Loss	16.26	0.1055	0.1055	0.1054	0.1055
Ada Focal Loss	16.68	0.1050	0.1049	0.1048	0.1050
Dual Focal Loss	16.95	0.1057	0.1055	0.1054	0.1057
CE+TS	16.38	0.0116	0.0097	0.0069	0.0078
CE+PTS	16.38	0.0120	0.0135	0.0097	0.0109
CE+Spline	16.38	0.0240	0.0286	0.0239	0.0257
CE+MnM	16.38	0.0134	0.0085	0.0069	0.0068
DISC	57.61 \pm 0.29	0.120 \pm 0.013	0.147 \pm 0.016	0.133 \pm 0.005	0.154 \pm 0.015
DISC+TS	57.61 \pm 0.29	0.061 \pm 0.007	0.053 \pm 0.012	0.065 \pm 0.008	0.053 \pm 0.013
L2B	69.66 \pm 0.19	0.133 \pm 0.009	0.152 \pm 0.022	0.171 \pm 0.017	0.121 \pm 0.008
L2B+TS	69.66 \pm 0.19	0.057 \pm 0.015	0.061 \pm 0.017	0.055 \pm 0.011	0.059 \pm 0.007
Ours	77.85\pm0.35	0.082\pm0.013	0.086\pm0.011	0.080\pm0.007	0.088\pm0.010
Ours+TS	77.85\pm0.35	0.011\pm0.005	0.014 \pm 0.009	0.013 \pm 0.007	0.009 \pm 0.005

Table 12: Detailed results on CIFAR-100 with asymmetric 40% noise.

Method	Test Acc (%)	ECE (\downarrow)	AdaECE (\downarrow)	ECE _{debias} (\downarrow)	ECE _{sweep} (\downarrow)
CE	41.85	0.0231	0.0242	0.0228	0.0227
Focal Loss	38.35	0.0316	0.0319	0.0313	0.0320
Ada Focal Loss	38.71	0.0163	0.0173	0.0160	0.0151
Dual Focal Loss	32.79	0.0540	0.0556	0.0536	0.0532
CE+TS	41.85	0.0253	0.0258	0.0254	0.0260
CE+PTS	41.85	0.0165	0.0156	0.0162	0.0166
CE+Spline	41.85	0.0177	0.0188	0.0155	0.0183
CE+MnM	41.85	0.0235	0.0245	0.0268	0.0276
DISC	76.50 \pm 0.15	0.140 \pm 0.017	0.135 \pm 0.012	0.127 \pm 0.023	0.123 \pm 0.015
DISC+TS	76.50 \pm 0.15	0.066 \pm 0.007	0.061 \pm 0.009	0.059 \pm 0.013	0.057 \pm 0.009
L2B	78.22 \pm 0.14	0.134 \pm 0.009	0.121 \pm 0.011	0.126 \pm 0.009	0.142 \pm 0.011
L2B+TS	78.22 \pm 0.14	0.067 \pm 0.007	0.058 \pm 0.008	0.061 \pm 0.009	0.071 \pm 0.015
Ours	79.67\pm0.22	0.071\pm0.011	0.084\pm0.012	0.076\pm0.009	0.077\pm0.013
Ours+TS	79.67\pm0.22	0.015\pm0.005	0.021 \pm 0.007	0.014\pm0.006	0.017\pm0.005

performance stems from two principles: (i) robust anchoring via a small clean reference set, which avoids error amplification, and (ii) distributional relabeling, which preserves uncertainty while improving both accuracy and calibration.

E ADDITIONAL EXPERIMENTAL RESULTS ON CIFAR-80N

To further assess the robustness of TrainRef under realistic noisy-label conditions, we conduct experiments on the CIFAR-80N benchmark. Following the protocol of (Yao et al., 2021), CIFAR-80N is constructed by treating the last 20 classes of CIFAR-100 as out-of-distribution (OOD), while the remaining 80 classes are considered in-distribution. This setting introduces open-set label noise by mixing semantically unrelated classes, which challenges a model’s ability to generalize under both closed-set and open-set noise.

We inject both symmetric and asymmetric label noise on the in-distribution subset, following the setup of (Sheng et al., 2024). Specifically, symmetric noise is applied at $\rho \in \{20\%, 80\%\}$ and asymmetric noise is applied at $\rho = 40\%$. These configurations allow us to evaluate model robustness under varying degrees of noise severity.

As shown in Table 14, TrainRef achieves substantial performance gains over previous state-of-the-art methods. In the Sym. 20% setting, TrainRef improves accuracy by **12.74%** over the best prior method. Under the severe Sym. 80% noise, TrainRef surpasses the closest baseline by **32.29%**. In the Asym. 40% case, which involves structured noise aligned with semantic class relationships, TrainRef achieves an improvement of **19.52%**.

These gains highlight the effectiveness of TrainRef’s unified framework in handling both closed-set and open-set noise. Notably, TrainRef does not discard OOD samples outright. Instead, it leverages reference-guided distributional labeling to assign soft targets to OOD samples based on semantic similarity. This design allows OOD instances to contribute positively to representation learning, rather than being treated as outliers.

These results reinforce the generalization ability of TrainRef in practical noisy-label scenarios, where label corruption often involves both ambiguity and distribution shift. Additional qualitative examples of TrainRef’s curation process can be found in (TrainRef, 2025).

Table 13: Accuracy and calibration on WebVision and Animal-10N. Lower calibration errors are better.

Method	WebVision				Animals-10N			
	Test Acc	ECE	AdaECE	ECE _{debias}	Test Acc	ECE	AdaECE	ECE _{sweep}
CE	63.23	0.1306	0.1306	0.1287	80.21	0.1659	0.1656	0.1659
DISC	80.17	0.1021	0.1021	0.1008	87.03	0.0865	0.0865	0.0876
Ours	82.33	0.0835	0.0823	0.0819	90.85	0.0289	0.0282	0.0298
CE+TS	63.23	0.0277	0.0312	0.0264	80.21	0.1306	0.1298	0.1305
DISC+TS	80.17	0.0337	0.0374	0.0323	87.03	0.0312	0.0306	0.0350
Ours+TS	82.33	0.0226	0.0265	0.0213	90.85	0.0254	0.0221	0.0253

Table 14: Test accuracy (%) on CIFAR-80N under varying noise levels. TrainRef achieves consistent improvements across both mild and severe noise settings in open-set scenarios.

Method	CIFAR-80N		
	Sym. 20%	Sym. 80%	Asym. 40%
Standard	29.37	4.20	22.25
Co-teaching (Han et al., 2018)	60.38	16.59	42.42
Co-teaching+ (Yu et al., 2019)	53.97	12.29	43.01
JoCoR (Wei et al., 2020)	59.99	12.85	39.37
Jo-SRC (Yao et al., 2021)	65.83	29.76	53.03
SELC (Lu & He, 2022)	57.51	22.79	47.50
DivideMix (Li et al., 2020)	57.47	21.18	37.47
Co-LDL (Sun et al., 2021)	58.81	24.22	50.69
UNICON (Karim et al., 2022)	54.50	36.75	51.50
NCE (Li et al., 2022)	58.53	39.34	56.40
SOP (Liu et al., 2022)	60.17	34.05	53.34
SPRL (Shi et al., 2023)	47.90	22.25	40.86
AGCE (Zhou et al., 2023)	60.24	25.39	44.06
DISC (Li et al., 2023)	50.33	38.23	47.63
SED (Sheng et al., 2024)	69.10	42.57	60.87
TrainRef (Ours)	81.84	74.86	80.39

F ADDITIONAL ABLATION STUDY

F.1 ABLATION ON INFLUENCE-BASED REFERENCE AUGMENTATION

To evaluate the effectiveness of influence-based reference set augmentation, we conduct a comparative study against several alternative strategies for reference construction and data utilization. This experiment is performed on CIFAR-100 under three distinct label noise conditions: symmetric noise at 20% and 80%, and instance-dependent noise at 40%.

We compare the following configurations:

- **KNN Embedding Voting:** Clean sample selection using k -nearest neighbor consistency in the embedding space, without reference set expansion or direct interaction with noisy labels.
- **Full Dataset Fine-tuning:** Standard fine-tuning on the entire noisy training set without any filtering.
- **Initial Reference Set Fine-tuning:** Model is fine-tuned only on the initial manually specified reference set (set to 500 samples).
- **First Augmented Reference Set Fine-tuning:** Model is trained using the reference set expanded via influence score-based selection.

Note that both the KNN-based method and the Initial Reference Set approach do not interact with noisy labels during training, and thus their performance remains constant across different noise configurations.

As shown in Table 15, fine-tuning on the influence-augmented reference set yields substantial gains across all noise settings. Compared to full-dataset training, the improvement exceeds 6% under symmetric 20% noise, 59% under symmetric 80% noise, and 22% under instance-dependent noise. These results underscore the importance of influence-guided augmentation in filtering out noisy examples and expanding the clean set with high precision.

Table 15: Ablation study on influence-based reference augmentation. Performance (accuracy in %) is reported under various label noise settings on CIFAR-100.

Method	CIFAR-100		
	Sym. 20%	Sym. 80%	Inst. 40%
KNN Embedding Voting	–	51.70 \pm 1.64	–
Full Dataset Fine-tuning	66.04 \pm 0.28	13.17 \pm 1.20	54.83 \pm 0.85
Initial Reference Set Fine-tuning	–	64.63 \pm 0.18	–
1st Augmented Ref. Set Fine-tuning	81.24 \pm 0.88	72.91 \pm 0.73	76.81 \pm 1.30

The ablation confirms that influence-based augmentation plays a central role in enabling TrainRef to scale from a minimal trusted set to a robust, curated training set, which in turn leads to substantial improvements in downstream performance.

F.2 EMBEDDING SPACE QUALITY ACROSS PHASES

The design of TRAINREF reflects a progressive strategy to *approximate* an ideal embedding through phase-wise refinement. Our objective is to demonstrate that improved embedding quality is positively correlated with better noise detection and label refinement.

To empirically validate this, we measure the quality of the learned embedding space at each stage of the training pipeline using a non-parametric KNN classifier. Specifically, we compute the top-1 KNN classification accuracy using features extracted from the frozen encoder after each phase. The rationale is that better separation and alignment of class representations in the feature space should yield higher KNN accuracy, making it a suitable proxy for embedding quality.

Table 16: KNN classification accuracy (%) on CIFAR-100 across different phases of TRAINREF. Embedding quality improves consistently as the model progresses through the three-phase framework.

Metric	Phase I (MIM)	Phase II (Ref. Aug)	Phase III (1st Iter)	Phase III (2nd Iter)
KNN Accuracy (%)	52.18	75.18	77.78	79.12

As shown in Table 16, the embedding quality improves substantially from Phase I to Phase III. The initial self-supervised encoder achieves modest KNN accuracy (52.18%), reflecting its general-purpose nature. Fine-tuning on the influence-augmented reference set in Phase II leads to a significant jump (75.18%), and iterative refinement in Phase III further improves separability, reaching 79.12% after the second iteration.

These results empirically support our design rationale: although a perfect embedding space is not assumed, our framework steers the representation space toward that ideal through principled, iterative refinement. We will revise the main text to make this intent more explicit and to avoid any ambiguity regarding our assumptions.

F.3 SENSITIVITY ANALYSIS OF δ_{IF}

Training samples with positive $IF(\mathbf{s}_i, \mathcal{D}_{ref})$ larger than threshold δ_{IF} are used to construct an augmented reference set \mathcal{D}_{ref}^* . In the main experiments we set $\delta_{IF} = 0.8$, and here we study its sensitivity under different values. After constructing \mathcal{D}_{ref}^* , the parameters θ and ϕ are updated jointly, denoted $\hat{\theta}$.

Table 17 reports F1 scores on CIFAR-100 across three noise settings when varying $\delta_{IF} \in \{0.9, 0.8, 0.7\}$. The results show stable performance across different thresholds, confirming the robustness of TrainRef to the choice of δ_{IF} .

F.4 GENERALIZATION TO NON-TRANSFORMER ARCHITECTURES

To assess whether TRAINREF is limited to transformer-based architectures, we investigate its applicability to convolutional neural networks (CNNs), specifically ResNet34.

We note that Phase I of TRAINREF leverages Masked Image Modeling (MIM), which is inherently tailored to transformer-based architectures such as BEiT v2. This is because patch-level masking and reconstruction, core to MIM objectives, are not naturally compatible with the inductive biases

Table 17: Sensitivity analysis of δ_{IF} on CIFAR-100. Results are reported as F1 scores.

CIFAR-100 Setting	$\delta_{IF} = 0.9$	$\delta_{IF} = 0.8$	$\delta_{IF} = 0.7$
Sym-50%	0.871	0.942	0.920
Asym-40%	0.834	0.921	0.907
Inst-40%	0.866	0.934	0.919

of CNNs. However, once the reference-guided soft labels are obtained, the curated dataset is architecture-agnostic and can be used to train alternative backbones.

To explore this, we adopt a hybrid setup where BEiT_{v2} is used solely for Phase I to obtain soft labels, and a ResNet34 is trained from scratch in Phases II and III using the curated dataset. Table 18 summarizes the performance under symmetric and instance-dependent label noise on CIFAR-100.

Table 18: Test accuracy (%) on CIFAR-100 with different architectures. BEiT_{v2} is used for soft-label generation, and ResNet34 is trained from scratch on the curated dataset. Despite underperforming the end-to-end BEiT_{v2} pipeline, the hybrid setup outperforms the strongest ResNet-based baseline (L2B-C2D), demonstrating architecture generalizability.

Method (Backbone)	Sym. 50%	Sym. 80%	Inst. 40%
DISC (Li et al., 2023) (ResNet34)	75.21 \pm 0.15	57.61 \pm 0.29	78.44 \pm 0.19
L2B-C2D (Zhou et al., 2024) (ResNet34)	78.10	69.60	–
TRAINREF (BEiT _{v2} \rightarrow ResNet34)	78.98 \pm 0.11	74.80 \pm 0.17	79.87 \pm 0.13
TRAINREF (BEiT _{v2} end-to-end)	82.07 \pm 0.17	77.85 \pm 0.35	82.33 \pm 0.16

These results show that although using BEiT_{v2} end-to-end yields the strongest performance—likely due to continuity in feature learning from MIM to classification—the hybrid setup still achieves significant gains over state-of-the-art CNN-based baselines. This underscores the robustness and modularity of our reference-based relabeling framework, which can benefit downstream models regardless of architecture.

We conclude that while transformer-based architectures are preferred due to their compatibility with MIM, the relabeling and curation components of TRAINREF are generalizable and transferable to alternative backbones such as CNNs.

F.5 FAIRNESS OF BACKBONE CHOICE

TRAINREF adopts a transformer-based backbone (BEiT_{v2}) for its end-to-end pipeline, whereas many prior baselines are implemented with ResNet-50. To ensure that the performance gains of TRAINREF are not solely attributable to architectural differences, we re-evaluate DISC and L2B under the same transformer backbone. This provides a fair comparison by aligning backbone capacity across methods.

Table 19 reports results on CIFAR-100 (Sym. 20%, Asym. 40%), WebVision, and Animals-10N. Transformer backbones improve both DISC and L2B compared to their ResNet-50 counterparts, but TRAINREF consistently achieves the highest accuracy. This indicates that while backbone choice contributes to performance, the primary gains arise from the proposed reference-based curation framework.

Table 19: Test accuracy (%) of DISC, L2B, and TRAINREF with ResNet-50 and transformer backbones. Results show that TRAINREF’s improvements persist under fair backbone alignment, confirming that the advantage is not due to architectural bias.

Method (Backbone)	CIFAR-100 Sym. 20%	CIFAR-100 Asym. 40%	WebVision	Animals-10N
DISC (ResNet-50)	78.75	76.50	80.28	87.10
DISC (Transformer)	80.31	77.52	80.79	88.45
L2B (ResNet-50)	79.67	78.22	80.56	89.03
L2B (Transformer)	80.91	79.03	81.15	89.92
TRAINREF (Transformer)	85.44	79.67	82.28	90.90

These findings demonstrate that transformer backbones provide benefits across methods, but the consistent superiority of TRAINREF highlights the effectiveness of its reference-based curation strategy rather than architectural advantage alone.

G PERFORMANCE UNDER NOISE-FREE CONDITIONS

To further assess the effectiveness and generalizability of TRAINREF, we report its performance under fully clean training conditions using standard cross-entropy (CE) loss. This experiment serves to answer whether the proposed soft-labeling framework is still beneficial in the absence of label noise.

We evaluate TRAINREF and several strong baselines on CIFAR-100 and CIFAR-80N under noise-free settings. Additionally, we conduct an ablation in which we disable the soft-labeling component of our method and train solely on one-hot targets derived from the clean labels.

Table 20: Test accuracy (%) on CIFAR-100 (noise-free) and CIFAR-80N (close-set noise-free, open-set noise at 20%). TRAINREF achieves state-of-the-art performance in both settings, showing benefits of soft-labeling and robustness under partial open-set corruption.

Method	CIFAR-100 (Clean)	CIFAR-80N (20% Open-Set Noise)
CE (Standard Cross-Entropy)	77.87 \pm 0.17	64.12 \pm 0.16
DISC (Li et al., 2023)	81.23 \pm 0.10	68.88 \pm 0.13
SED (Sheng et al., 2024)	67.48 \pm 0.21	69.80 \pm 0.19
TRAINREF (w/o soft label)	83.77 \pm 0.10	80.19 \pm 0.13
TRAINREF	85.87 \pm 0.15	82.81 \pm 0.20

As shown in Tables 20, TRAINREF achieves 85.87% accuracy on CIFAR-100 and 82.81% on CIFAR-80N under noise-free conditions. These results are only marginally lower than those obtained under symmetric 20% noise (85.44% and 81.84%, respectively), with performance drops of just 0.43% and 0.97%. In contrast, the best baseline (DISC) experiences significantly larger degradations of 2.48% and 8.64%, respectively.

Furthermore, removing the soft-labeling component from TRAINREF leads to noticeable declines in accuracy, even under clean supervision. This supports our claim that rigid one-hot labels may introduce inductive bias or semantic overconfidence, particularly in ambiguous instances, and that learning from distributional supervision remains beneficial.

These findings validate the utility of our approach in both noisy and clean regimes and emphasize the general-purpose benefit of soft label learning.

H LIMITATIONS

While TRAINREF demonstrates strong performance across noisy vision benchmarks, several limitations remain:

- **Generalization to Other Modalities.** Our study is limited to image classification tasks. Although the framework of reference-guided distributional curation is conceptually extensible, adapting it to other modalities such as text and speech requires careful design of influence functions and embedding spaces that may differ substantially from vision tasks.
- **Scalability to Large-Class Problems.** Even though TRAINREF is effective with as little as one clean sample per class, scaling to tasks with tens of thousands of classes (e.g., fine-grained clinical coding) still requires non-trivial human effort to collect a sufficiently diverse reference set. Reducing this dependency on human annotation remains an important direction.
- **Reliance on Reference Anchors.** The success of our method hinges on the availability of a trusted reference set, however small. In domains where no reliable clean data exists, alternative strategies for bootstrapping anchors are necessary.

These limitations highlight opportunities for future work, particularly in extending TRAINREF to broader modalities and reducing its reliance on human effort in extremely large-scale classification settings.

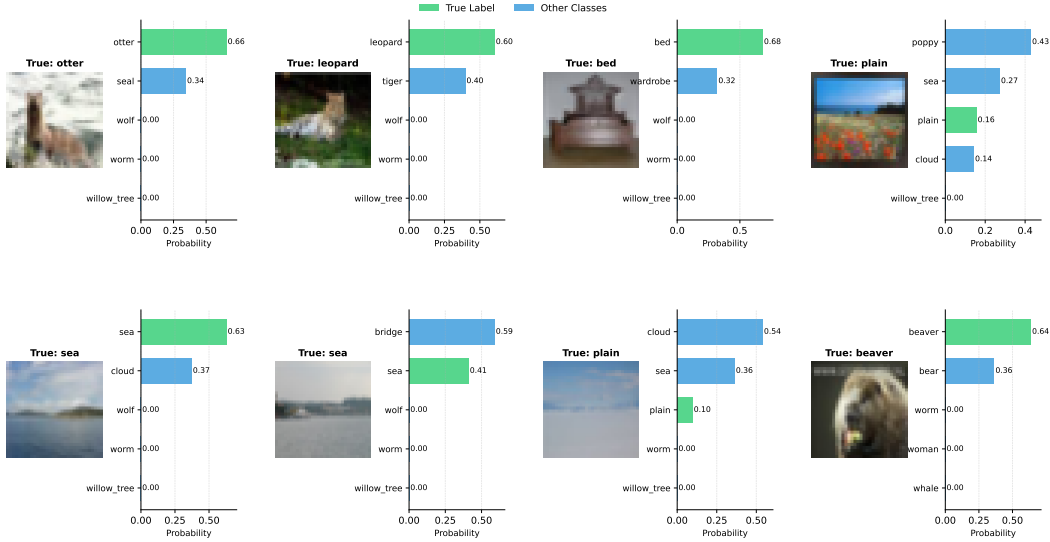


Figure 5: Representative ambiguous CIFAR-100 samples with expert-provided soft labels used to evaluate distributional-noise curation.

I PER-TYPE ANALYSIS OF CATEGORICAL VS. DISTRIBUTIONAL NOISE

TrainRef is designed to address two complementary forms of label misinformation: (i) *categorical noise*, where the ground-truth is one-hot but the observed label is flipped, and (ii) *distributional noise*, where the ground-truth should be a soft class distribution due to inherent ambiguity. To quantify TrainRef’s effectiveness on each type separately, we conduct the following controlled analysis on CIFAR-100.

Subset construction. We embed all CIFAR-100 training images using a pretrained DINOv2 encoder and compute a local neighborhood label distribution for each sample via k -nearest-neighbor voting in embedding space. We then use the entropy of this neighborhood distribution as an ambiguity indicator:

- **Distributional-noise subset (ambiguous).** We select samples with high neighborhood entropy ($H > 1.5$), and randomly sample 50 cases. Three independent experts annotate each case with a soft label distribution. Representative samples and expert-provided soft labels are shown in Figure 5.
- **Categorical-noise subset (unambiguous + injected flips).** We select low-entropy samples ($H < 0.1$) as unambiguous instances, inject 20% symmetric hard-flip noise, and evaluate TrainRef’s ability to identify and remove mislabeled samples.

Evaluation metrics. For the categorical-noise subset, we report the mislabeled fraction before and after curation. For the distributional-noise subset, we measure the KL divergence between TrainRef’s curated soft labels and the expert soft labels.

Results. After TrainRef curation:

- **Categorical noise rate:** 20% \rightarrow 0.32%.
- **Distributional noise (KL to human soft labels):** 1.67 \rightarrow 1.43.

These results indicate that TrainRef removes categorical noise aggressively by filtering or correcting clear label flips, while refining distributional noise more subtly by shifting labels toward calibrated soft distributions rather than discarding them. Importantly, as demonstrated in Table 7, preserving and curating distributional labels is crucial for both accuracy and confidence calibration, even when the absolute reduction in KL is smaller.

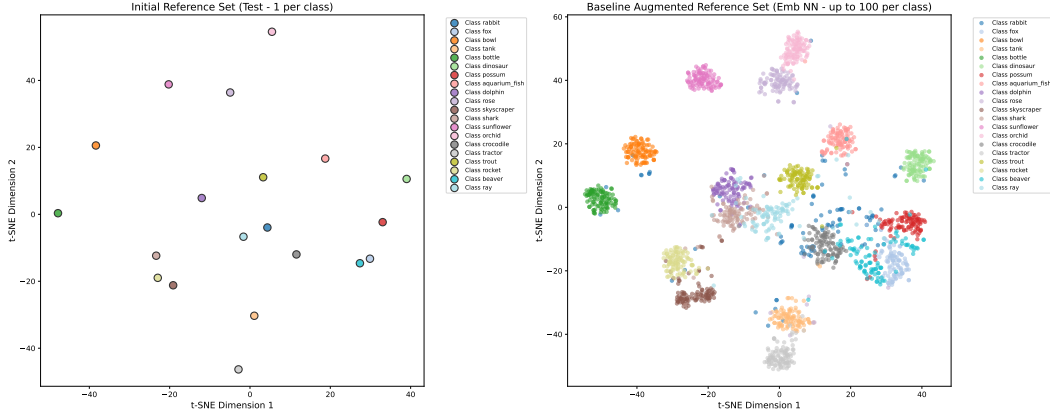


Figure 6: Embedding-NN augmentation expanded from the same initial D_{ref} . The expansion clusters tightly around the seeds, indicating limited diversity gain.

J REFERENCE-SET DIVERSITY: MEASUREMENT, AUGMENTATION BEHAVIOR, AND THRESHOLD SENSITIVITY

TrainRef relies on a small clean reference set D_{ref} and its influence-augmented expansion D_{ref}^* . A key practical concern is whether D_{ref} is sufficiently diverse to represent semantic modes within each class, and whether influence-based augmentation preserves or improves this diversity.

Measuring diversity. Diversity is not characterized by set size alone. We measure *semantic diversity within each class* using the average pairwise cosine similarity of reference embeddings:

$$\text{Sim}_{\text{intra}}(c) = \frac{2}{|D_{\text{ref}}^c|(|D_{\text{ref}}^c| - 1)} \sum_{i < j} \cos(z_i, z_j),$$

where z_i is the DINOv2 embedding of sample i and D_{ref}^c denotes reference samples in class c . We report the mean over classes. Lower $\text{Sim}_{\text{intra}}$ indicates broader coverage of distinct semantic modes (higher diversity).

Why influence augmentation does not collapse diversity. Influence scores are computed via *gradient alignment* with the reference training signal (Sec. 3.1), rather than raw embedding proximity. A candidate is added to D_{ref}^* if it strengthens (or at least does not conflict with) the reference objective. As a result, TrainRef can select label-consistent yet embedding-diverse samples, instead of only near-duplicates of the initial seeds.

Empirical comparison at matched size. To isolate the effect of augmentation strategy from reference size, we compare two expansions with the same number of added samples per class: (i) **Embedding-NN augmentation**, which adds nearest neighbors in embedding space; and (ii) **Influence augmentation (ours)**, which adds samples with high influence scores (Sec. 3.1). Average intra-class cosine similarity (lower = more diverse):

- Embedding-NN augmentation: 0.67
- Influence augmentation (ours): 0.55

Figures 6 and 7 provide qualitative evidence: embedding-NN expansion concentrates around the initial seeds, while influence expansion covers multiple semantic modes per class.

Influence-threshold sensitivity. The influence threshold δ_{IF} primarily controls the *cleanliness* of D_{ref}^* with an indirect cleanliness–diversity trade-off: higher δ_{IF} yields a cleaner but potentially narrower expansion, while lower δ_{IF} admits mildly aligned samples that may increase coverage but risk adding noise. Sensitivity results in Appendix F.3 (Table 17) show TrainRef remains stable across a reasonable range of δ_{IF} , indicating that performance does not hinge on a finely tuned threshold.

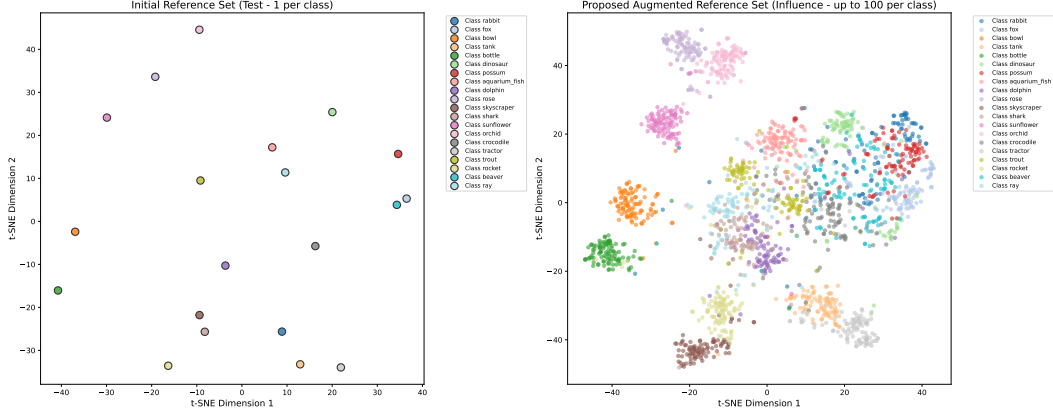


Figure 7: Influence-based augmentation expanded from the same initial D_{ref} . The expansion covers multiple semantic modes per class, increasing reference diversity.

K EFFECT OF INITIAL NOISE AND INFLUENCE THRESHOLD ON PHASE III CONVERGENCE

We provide a theoretical insight into how the initial noise level of the training set (p_0) and the influence threshold (δ_{IF}) used in Phase II affect the convergence speed of Phase III co-evolution.

Setup recap. Let $p_0 := \mathbb{P}[\tilde{y} \neq y^*]$ be the initial label noise rate in the noisy training set \tilde{D} . Phase II augments the clean reference set D_{ref} into D_{ref}^* by selecting training samples whose influence score exceeds δ_{IF} .

Define

$$\alpha_c(\delta_{\text{IF}}) = \mathbb{P}[\text{clean sample added}], \quad \alpha_n(\delta_{\text{IF}}) = \mathbb{P}[\text{noisy sample mistakenly added}],$$

and the noise-to-clean ratio

$$\kappa(\delta_{\text{IF}}) := \frac{\alpha_n(\delta_{\text{IF}})}{\alpha_c(\delta_{\text{IF}})}.$$

Then the resulting noise rate of D_{ref}^* is

$$q(\delta_{\text{IF}}; p_0) = \frac{p_0 \kappa(\delta_{\text{IF}})}{(1 - p_0) + p_0 \kappa(\delta_{\text{IF}})}. \quad (11)$$

Assumption K.1 (i.i.d. soft voting model). For a fixed sample with true label y^* , let $Z_j \in [0, 1]$ denote the soft weight assigned to y^* by the j -th voting neighbor in the C-step. Assume $\{Z_j\}_{j=1}^K$ are i.i.d. with mean $\mu(\delta_{\text{IF}}; p_0) > \frac{1}{2}$.

Assumption K.2 (Clean/noisy neighbor separation). There exists $\beta \in (1/2, 1]$ such that

$$\mu_c(\delta_{\text{IF}}) := \mathbb{E}[Z_j \mid j \text{ clean}] \geq \beta, \quad \mu_n(\delta_{\text{IF}}) := \mathbb{E}[Z_j \mid j \text{ noisy}] \leq 1 - \beta.$$

Key bound on C-step error. Let $\bar{Z} = \frac{1}{K} \sum_{j=1}^K Z_j$ be the average soft support for the true class. By Hoeffding's inequality and Assumptions K.1–K.2,

$$\begin{aligned} \mathbb{P}[\text{C-step wrong}] &= \mathbb{P}[\bar{Z} \leq \tfrac{1}{2}] \leq \exp(-2K(\mu - \tfrac{1}{2})^2) \\ &\leq \exp(-c(1 - 2q(\delta_{\text{IF}}; p_0))^2), \end{aligned} \quad (12)$$

where $c := 2K(\beta - \frac{1}{2})^2$. Thus, a cleaner augmented reference set (smaller q) yields a smaller C-step error.

Assumption K.3 (Co-evolution error contraction). One full co-evolution iteration contracts the classification error:

$$e_{t+1} \leq \rho(p_0, \delta_{\text{IF}}) e_t, \quad \rho(p_0, \delta_{\text{IF}}) = \exp(-c(1 - 2q(\delta_{\text{IF}}; p_0))^2) \in (0, 1).$$

Theorem K.4 (Iteration complexity of Phase III). *Under Assumptions K.1 and K.3, if $q(\delta_{\text{IF}}; p_0) < \frac{1}{2}$, then Phase III converges linearly:*

$$e_t \leq \rho(p_0, \delta_{\text{IF}})^t e_0.$$

To achieve $e_t \leq \varepsilon$, it suffices to take

$$T \geq \frac{\log(e_0/\varepsilon)}{-\log \rho(p_0, \delta_{\text{IF}})} = \frac{\log(e_0/\varepsilon)}{c(1 - 2q(\delta_{\text{IF}}; p_0))^2}. \quad (13)$$

Setting $e_0 \approx p_0$ and substituting equation 11 yields

$$T \geq \frac{\log(p_0/\varepsilon)}{c} \cdot \frac{((1 - p_0) + p_0\kappa(\delta_{\text{IF}}))^2}{(1 - p_0(1 + \kappa(\delta_{\text{IF}})))^2}. \quad (14)$$

Interpretation. Equation equation 14 makes the dependence explicit:

- **Effect of initial noise p_0 .** Larger p_0 increases the required iterations through both the $\log(p_0/\varepsilon)$ term and by enlarging $q(\delta_{\text{IF}}; p_0)$, which weakens the contraction factor.
- **Effect of influence threshold δ_{IF} .** Increasing δ_{IF} makes Phase II selection more stringent, decreasing $\kappa(\delta_{\text{IF}})$ and thus $q(\delta_{\text{IF}}; p_0)$. This strengthens contraction and reduces T . Conversely, an overly low threshold may admit more noisy references, increasing q and slowing convergence.

Example. Suppose $p_0 = 0.5$, $\delta_{\text{IF}} = 0.8$, and $\kappa(\delta_{\text{IF}}) \approx 1/20$ (i.e., clean samples are $\sim 20\times$ more likely to be selected than noisy ones). To reach $\varepsilon = 0.2$, equation 14 gives

$$T \gtrsim \frac{\log(5/2)}{c} \cdot \frac{(1 + \kappa)^2}{(1 - \kappa)^2} \approx \frac{1.12}{c},$$

suggesting that only ~ 2 iterations are sufficient when $c \approx 1$. Empirically, we observe that 3 Phase III iterations are enough for convergence across most noise settings, consistent with the bound.

L USE OF LARGE LANGUAGE MODELS

In preparing this manuscript, we employed large language models (LLMs) solely as auxiliary tools for language refinement. Their usage was limited to polishing expressions, checking grammar, and improving readability. No parts of the technical content, experimental design, analysis, or results were generated by LLMs. All scientific contributions, methods, and evaluations presented in this paper were conceived, implemented, and validated entirely by the authors.