CONTRASTIVE DIVIDEMIX

Robust Learning of Noisy Labels Using Jo-SRC & DivideMix

CAI4105

UCF

Tyler Tran

Abstract

Learning robust image classifiers under heavy label noise remains a core challenge in modern machine learning. Although DivideMix has demonstrated impressive performance by combing semi-supervised learning with per-sample loss modeling, its effectiveness can degrade when noise structure is complex or pseudo-labels become unstable during early training. In this project, we explore a modified training pipeline that integrates consistency based joint optimization into the DivideMix framework. This method enforces prediction consistency across augmentations while refining the pseudo-label principle. We evaluate the CIFAR-10 under specific symmetric noise settings. Results show that adding JO-SRC variables reduces the noisy-label overfitting and improves stability in the initial stages of training, results are similar to DivideMix's baseline warmup. These finding show that incorporating a simple consistency-based refinement can provide robust improvements without altering the core structure of the framework. We finish with an analysis of the failures observed and propose future directions for more stable noise-robust learning.

Introduction

Neural networks can achieve very good performance on clean, well-labeled datasets, but their accuracy drops sharply when label are corrupted. Real world data such as crowdsources annotations or user-generated content often contain significant noise, making robust training essential. Some recent methods for learning with noisy labels use semi-supervised learning, mixture models, and regularization to prevent models from overfitting incorrect labels. Among these applications, DivideMix has become a state-of-the-art technique by estimating the probability that each sample is clean and then applying MixMatch semi-supervised learning to the data. However, DivideMix can struggle when severe noise is present or if it varies between classes. In cases like these. Pseudo-labels produced early in training may not be reliable enough to guide the semi-supervised model to learn effectively. To solve this, we explore an enhanced training method that incorporates a joint optimization, inspired by the Jo-SRC methodology. The idea is to encourage the model to rely on intrinsic structure in the data rather than labels alone.

The three goals of this project are:

1. Understand the DivideMix framework and build it.
2. Integrate Jo-SRC consistency refinement into the DivideMix framework.
3. Evaluate whether this hybrid approach improves robustness against noise on CIFAR-10.

Our experiments show that this added term stabilizes early training. We demonstrate where the method succeeds, fails, and why consistency-based modifications can be beneficial for noise-robust systems.

Background

Training deep neural networks in the presence of noisy labels is a long-standing challenge because conventional supervised learning encourages overfitting to incorrect annotations. Early layers of neural networks tend to learn generalizable patterns, but over time the model begins to memorize mislabeled samples, leading to poor generalization. This behavior has motivated a series of noise-robust training strategies, including sample reweighting, label correction, consistency regularization, and semi-supervised learning.

Related Works

Co-teaching / Co-teaching+

Uses two networks that exchange low-loss samples. Low-loss filtering avoids overfitting but fails when noisy samples have artificially low loss, or clean samples are difficult (high loss).

MixMatch (Semi-Supervised Learning)

MixMatch generates pseudo-labels for unlabeled data, enforces consistency across augmentations, and uses MixUp regularization. It is highly effective for semi-supervised learning but assumes pseudo-labels are relatively clean.

DivideMix (State-of-the-art noisy-label learning)

DivideMix applies MixMatch while partitioning samples into clean and noisy groups using a per-sample loss Gaussian Mixture Model (GMM). This allows the model to treat reliable samples as labeled and unreliable ones as unlabeled. From the official DivideMix paper:

Co-Divide: Separating Clean and Noisy Samples

Let the training set be

$$D = \{(x_i, y_i)\}_{i=1}^{N},$$

where $x_i$is an image and $y_i \in \{0,1\}^C$is a one-hot class label.
Given a model with parameters θ, the per-sample cross-entropy loss is:

$$\ell_i(\theta) = -\sum_{c=1}^{C} y_{ic} \log\, p_{\text{model}}(x_i; \theta)_c.$$

Following prior work showing that clean samples tend to have lower loss, DivideMix models the distribution of these losses using a **two-component Gaussian Mixture Model (GMM)**. Unlike the Beta Mixture Model used in earlier approaches, the GMM oversees asymmetric noise better because it can fit sharper or flatter distributions as needed.

For each sample, the GMM produces the posterior probability of belonging to the low-loss (clean) component:

$$w_i = p(g_{\text{clean}} \mid \ell_i)$$

where $g_{\text{clean}}$is the Gaussian component with smaller mean.

Using a threshold on w_i, the dataset is separated into:

- **Labeled set** (estimated clean):

$$X = \{(x_i, y_i, w_i) \mid w_i \geq \tau\}$$

- **Unlabeled set** (estimated noise):

$$U = \{x_i \mid w_i < \tau\}$$

To avoid reinforcing a model's own mistakes, DivideMix uses **co-divide**:
each of the two networks fits a GMM on its own loss values, and the resulting clean/noisy split is used to train the *other* network. Differences in initialization, mini-batch order, and prediction trajectories keep the networks diverse, enabling them to filter out several types of noise.

MixMatch with Label Co-Refinement and Co-Guessing

At each training iteration, labeled and unlabeled subsets are fed into an enhanced MixMatch pipeline.

Label co-refinement (for labeled samples)

For labeled examples, the ground-truth label $y_b$is merged with the prediction $p_b$from the other network, weighted by the clean probability w_b:

$$y'_b = w_b\, y_b \;+\; (1 - w_b)\, p_b$$

A temperature sharpening operator is then applied:

$$\tilde{y}_{b,c} = \frac{(y'_{b,c})^{1/T}}{\sum_{k=1}^{C} (y'_{b,k})^{1/T}}$$

Co-guessing (for unlabeled samples)

For unlabeled samples, DivideMix averages predictions from *both* networks across multiple augmentations:

$$q_b = \frac{1}{2M} \sum_{m=1}^{M} (p_{\text{model}_1}(u_b^m) + p_{\text{model}_2}(u_b^m))$$

A sharpening step yields a confident pseudo-label $\tilde{q}_b$.

After co-refinement and co-guessing, MixMatch applies **MixUp interpolation**:

Given two samples $(x_1, p_1)$and (x_2, p_2),

$$\lambda \sim \text{Beta}(\alpha, \alpha), \lambda = \max(\lambda, 1 - \lambda),$$
$$x = \lambda x_1 + (1 - \lambda) x_2, p = \lambda p_1 + (1 - \lambda) p_2$$

This produces mixed labeled samples $X'$and mixed unlabeled samples U'.

DivideMix Loss Function

Following MixMatch, DivideMix uses two losses:

Supervised loss (on estimated clean samples)

Cross-entropy:

$$\mathcal{L}_X = -\frac{1}{\mid X' \mid} \sum_{(x,p) \in X'} \sum_{c=1}^{C} p_c \log\, p_{\text{model}}(x)_c$$

Unsupervised consistency loss (on estimated noisy samples)

Mean-squared error between predictions and pseudo-labels:

$$\mathcal{L}_U = \frac{1}{|U'|} \sum_{(x,q)\in U'} \| p_{\text{model}}(x) - q \|_2^2$$

Regularization against class collapse

To avoid the model predicting a single dominant class when noise is high, DivideMix incorporates the uniform-prior regularizer:

$$\mathcal{L}_{reg} = \sum_{c=1}^{C} \pi_c \log \bar{p}_c, \pi_c = \frac{1}{C}$$

where

$$\bar{p}_c = \frac{1}{|X'| + |U'|} \sum_x p_{\text{model}}(x)_c$$

is the average batch prediction.

Final DivideMix Total Loss

$$\boxed{\mathcal{L} = \mathcal{L}_X + \lambda_u \mathcal{L}_U + \lambda_{reg} \mathcal{L}_{reg}}$$

DivideMix typically sets $\lambda_{reg} = 1$and tunes the unsupervised weight λ_u.

JO-SRC (Joint Optimization with Self-Referential Consistency)

JO-SRC (“Joint Sample Selection and Relabeling with Consistency Regularization”) is a noise-robust training framework designed to manage both in-distribution (ID) and out-of-distribution (OOD) label noise. Unlike methods that rely solely on loss values or heuristics, JO-SRC builds a principled mechanism for global clean-sample selection, ID/OOD distinction, and label re-assignment guided by a mean-teacher model. The final training objective integrates these components with an auxiliary consistency loss to enhance generalization. From the official Jo-SRC paper:

Global Clean-Sample Selection via Jensen–Shannon Divergence

Instead of relying on per-mini-batch loss heuristics, JO-SRC measures the alignment between the predicted distribution

$$p_i = [p_{i1}, \ldots, p_{iC}]$$

and the label distribution

$$y_i = [y_{i1}, \dots, y_{iC}]$$

using the Jensen–Shannon (JS) divergence:

$$d_i = D_{\mathrm{JS}}(p_i \parallel y_i) = \frac{1}{2} D_{\mathrm{KL}}(p_i \parallel \frac{p_i + y_i}{2}) + \frac{1}{2} D_{\mathrm{KL}}(y_i \parallel \frac{p_i + y_i}{2})$$

Because the JS divergence is bounded in [0,1], it naturally serves as a global, normalized measure of sample reliability.
JO-SRC converts this value into a “clean likelihood”:

$$P_{\mathrm{clean}}(x_i) = 1 - d_i$$

A sample is selected as clean if:

$$P_{\mathrm{clean}}(x_i) > \tau_{\mathrm{clean}}$$

This metric parallels the small-loss criterion but avoids its instability across mini-batches.

Distinguishing ID vs. OOD Noisy Samples

Not all non-clean samples should be discarded: ID samples with corrupted labels can still be useful if properly re-labeled, while OOD samples should not influence the model strongly.

JO-SRC distinguishes them using prediction consistency across augmentations.

Given two augmentations of x_i:

$$v_i = T(x_i), v_i' = T'(x_i)$$

the model produces predictions $p_i$and p_i'.
The OOD likelihood is defined as:

$$P_{\mathrm{ood}}(x_i) = \mathbb{1}[\arg\max_c p_{ic} \neq \arg\max_c p_{ic}']$$

or in the authors softened form:

$$P_{\text{ood}}(x_i) = \min\left(1, \mathbb{1}[\arg\max p_i \neq \arg\max p_i']\right)$$

A sample is classified as:

- Clean if $P_{\text{clean}} > \tau_{\text{clean}}$
- ID noisy if $P_{\text{clean}} \leq \tau_{\text{clean}}$and $P_{\text{ood}} \leq \tau_{\text{ood}}$
- OOD noisy if $P_{\text{clean}} \leq \tau_{\text{clean}}$and $P_{\text{ood}} > \tau_{\text{ood}}$

This separation allows tailored rebel for each case.

Label Re-Assignment

After dividing data into three subsets

$$S_{\text{clean}}, S_{\text{id}}, S_{\text{ood}}$$

JO-SRC reassigns label distributions as follows.

<u>Clean samples</u>

Clean samples keep their true labels but undergo label smoothing:

$$y_{ic} = \begin{cases} 1-\epsilon, & c = l_i \\ \dfrac{\epsilon}{C-1}, & c \neq l_i \end{cases}$$

<u>ID noisy samples</u>

These samples receive pseudo-labels from a mean-teacher model:

$$y_{ic} = p_{\text{teacher}}(x_i)_c$$

<u>OOD noisy samples</u>

OOD samples should not drive the model toward any class.
Thus, JO-SRC assigns a soft uniform-like pseudo-label:

$$y_{ic} = \frac{\exp(p_{\text{teacher}}(x_i)_c/s)}{\sum_{j=1}^{C} \exp\left(p_{\text{teacher}}(x_i)_j/s\right)}, s \gg 1$$

which approaches a uniform distribution.

The mean-teacher parameters are updated as an exponential moving average (EMA):

$$\theta_{\mathrm{mt}} \leftarrow \alpha\,\theta_{\mathrm{mt}} + (1-\alpha)\,\theta$$

Consistency Regularization

To strengthen clean/ID/OOD separation, JO-SRC introduces a prediction-consistency regularizer:

$$\mathcal{L}_o = \frac{1}{N}\sum_{i=1}^{N} \gamma_i [D_{\mathrm{KL}}(p_i \parallel p_i') + D_{\mathrm{KL}}(p_i' \parallel p_i)]$$

where

- $\gamma_i = 1$for clean and ID samples
- $\gamma_i = -1$for OOD samples,

encouraging:

- high consistency for clean and ID samples,
- high inconsistency for OOD samples.

Final JO-SRC Objective

The final loss combines the standard supervised loss on the re-labeled samples and the consistency regularizer:

$$\boxed{\mathcal{L} = (1-\lambda)\,\mathcal{L}_c + \lambda\,\mathcal{L}_o}$$

with

$$\mathcal{L}_c = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic} \log\, p_{ic}$$

The threshold $\tau_{\text{clean}}$is dynamically increased over training to gradually tighten the clean sample criterion as the model becomes more confident.

Experimentation

We evaluated our method with noisy classification benchmarks, in specific the CIFAR-10 worst dataset. We will be comparing our approach with 20 epochs and 10 warmup – warmup being a variable that will train the model in the first few epochs that we designate – and 120 epochs with 10 warmup. This will be the variables for training both DivideMix default and our hybrid mix. Due to constraints of time and hardware along with a stronger baseline being used, we will be comparing the hybrid pipeline with DivideMix baseline to see if it improves the overall baseline.

Implementation

Backbone network:

- ResNet-28
- Batch size = 64
- Optimizer: SGD & momentum = 0.9
- Learning rate: Cosine annealing

DivideMix Component

We use DivideMix for sample partitioning and MixMatch semi-supervised training for early epochs implemented at SemiLoss, along with GMM-style selection, and the co-training scheme. Divide mix uses a co-training where 2 networks estimate which samples are clean vs. noise by fitting a two-component GMM to the per-sample losses. The low-loss means clean and the high-loss means noisy.

Supervised cross-entropy on "labeled" batch

Given sharpened targets $\tilde{y}_i$:

$$\mathcal{L}_X = -\frac{1}{\mid B_x \mid} \sum_i \sum_{c=1}^{C} \tilde{y}_{ic} \log\, p_\theta(x_i)_c$$

In code:

```
Lx = -torch.mean(torch.sum(F.log_softmax(outputs_x, dim=1) * targets_x, dim=1))
```

Unsupervised consistency MSE for unlabeled samples

Loss:

$$\mathcal{L}_U = \frac{1}{\mid B_u \mid} \sum_i \parallel p_\theta(u_i) - \tilde{q}_i \parallel_2^2$$

Uniform prior regularization

(Original DivideMix)

$$\mathcal{L}_{reg} = \sum_c \pi_c \log \frac{\pi_c}{\bar{p}_c}, \pi_c = \frac{1}{C}$$

In code:

prior = torch.ones(args.num_class)/args.num_class

pred_mean = softmax(logits).mean(0)

penalty = torch.sum(prior * torch.log(prior / pred_mean))

Ramp-up coefficient

(From DivideMix)

$$\lambda_u = \lambda_u^{(max)} \cdot \text{linear\_rampup}(epoch - warmup, 16)$$

In code:

```
current = np.clip(current - warm_up)/16, 0, 1)
        return args.lambda_u * current
```

Jo-SRC Component

Jo-SRC in this hybrid pipeline adds a Jo-SRC-style sample selection and pseudo-label refinement on top of DivideMix's training framework.

Clean Sample Selection via Jensen–Shannon Divergence

For each sample, compute the prediction:

probs = softmax((net1(x) + net2(x))/2)

Compute smoothed labels:

targets_smooth = label_smoothing(targets)

Then compute the JS divergence:

$$D_{JS}(p, q) = \frac{1}{2} D_{KL}(p \,||\, m) + \frac{1}{2} D_{KL}(q \,||\, m), \qquad m = \tfrac{1}{2}(p + q)$$

In code:

js_div = js_divergence(probs, targets_smooth)

Then clean-likelihood:

$$P_{\text{clean}}(x) = 1 - \frac{D_{JS}(p, q)}{\log 2}$$

In code:

p_clean_batch = 1.0 - torch.clamp(js_div / np.log(2), 0, 1)

Threshold schedule:

tau_clean = (epoch / warmup_epochs) * args.tau_clean

Final clean selection:

$$\text{clean}(x_i) = \mathbf{1}[P_{\text{clean}}(x_i) > \tau_{\text{clean}}]$$

Pseudo-Label Reassignment (Mean Teacher)

Mean-teacher model is used for Jo-SRC pseudo-labeling:

$$\tilde{y}_i = \text{softmax}(\frac{f_{\text{mt1}}(x_i) + f_{\text{mt2}}(x_i)}{2})$$

In code:

outputs_mt1 = mean_teacher1(inputs)

outputs_mt2 = mean_teacher2(inputs)

pseudo_probs = softmax((outputs_mt1 + outputs_mt2)/2)

Label Mixing Based on Clean Likelihood

Our training combines:

- the original one-hot label

- the model-averaged predictions
- the mean teacher pseudo labels

Based on w_x(clean probability / GMM weight):

$$\tilde{p}_i = w_i \cdot y_i^{\text{onehot}} \; + \; (1 - w_i) \cdot \text{softmax}(f(x_i))$$

In code:

px = w_x * labels_x_onehot + (1 - w_x) * px

Then sharpen:

ptx = px ** (1/args.T)

targets_x = (ptx / ptx.sum(...)).detach()

Jo-SRC Consistency Regularization

$$\mathcal{L}_{cons} = D_{KL}(p_1 \mid\mid p_2) + D_{KL}(p_2 \mid\mid p_1)$$

In code:

consistency_loss = KL(p1||p2) + KL(p2||p1)

Weighted in total loss:

loss += args.consistency_lambda * consistency_loss

Total Jo-SRC + DivideMix Loss:

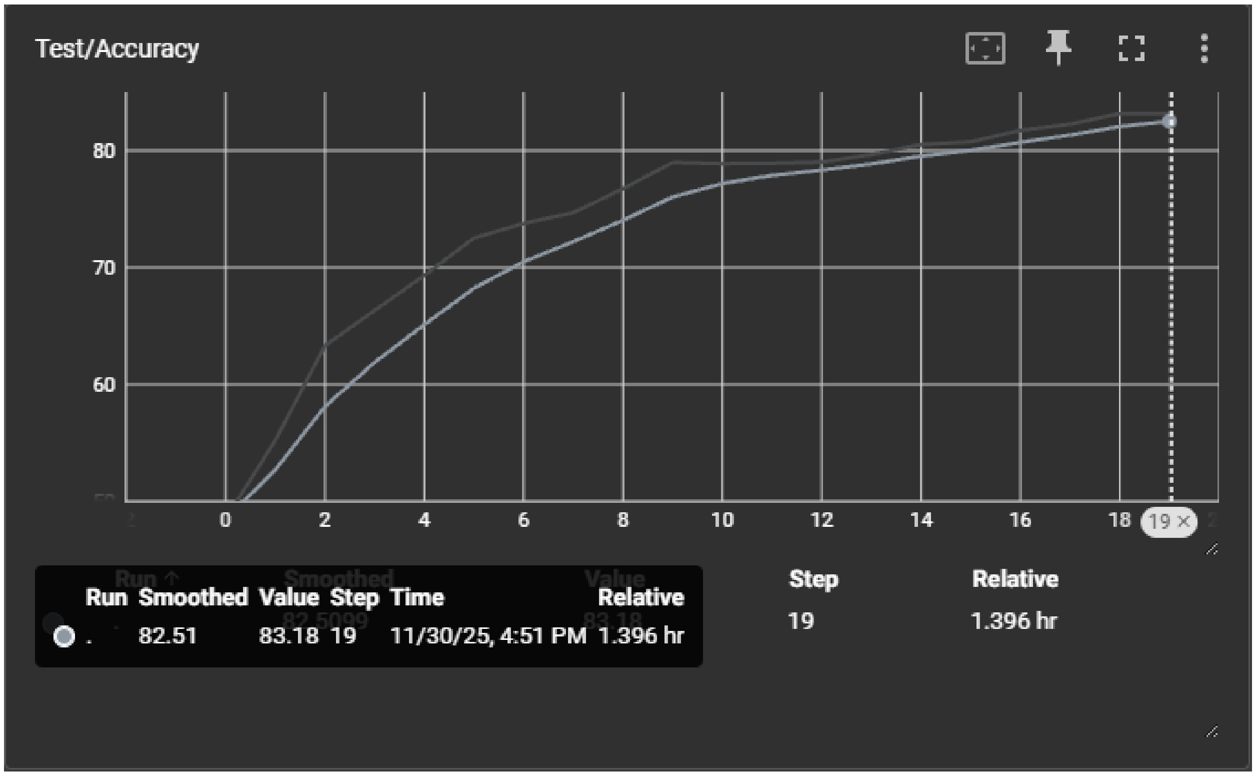
$$\mathcal{L} = \mathcal{L}_X + \lambda_u \mathcal{L}_U + \mathcal{L}_{reg} + \lambda_{cons} \mathcal{L}_{cons}.$$

<u>Summary for hybrid pipeline:</u>

1. Warmup – CE only
2. Clean/Noisy detection – Jo-SRC (JS divergence)
3. Pseudo-label assignment – Jo-SRC mean-teacher
4. Labeled loss – DivideMix MixMatch
5. Unlabeled loss DivideMix consistency
6. Consistency regularization – Jo-SRC KL bidirectional
7. Total training – Jo-SRC + DivideMix

Results

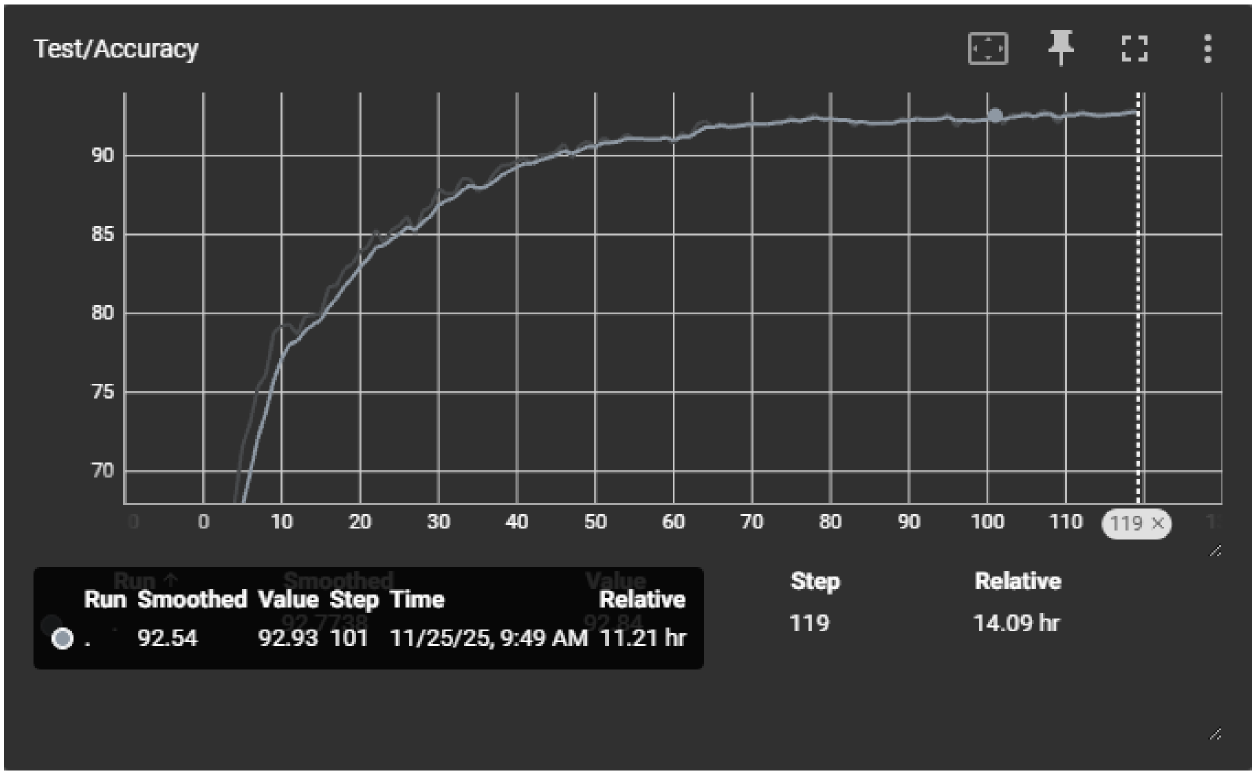
In testing epochs of 20 and 120 with a warmup of 10, the finding are interesting. The hybrid pipeline did *almost* as well as baseline DivideMix with some caveats.



Run	Smoothed	Value	Step	Time	Relative
● .	82.51	83.18	19	11/30/25, 4:51 PM	1.396 hr

Step	Relative
19	1.396 hr

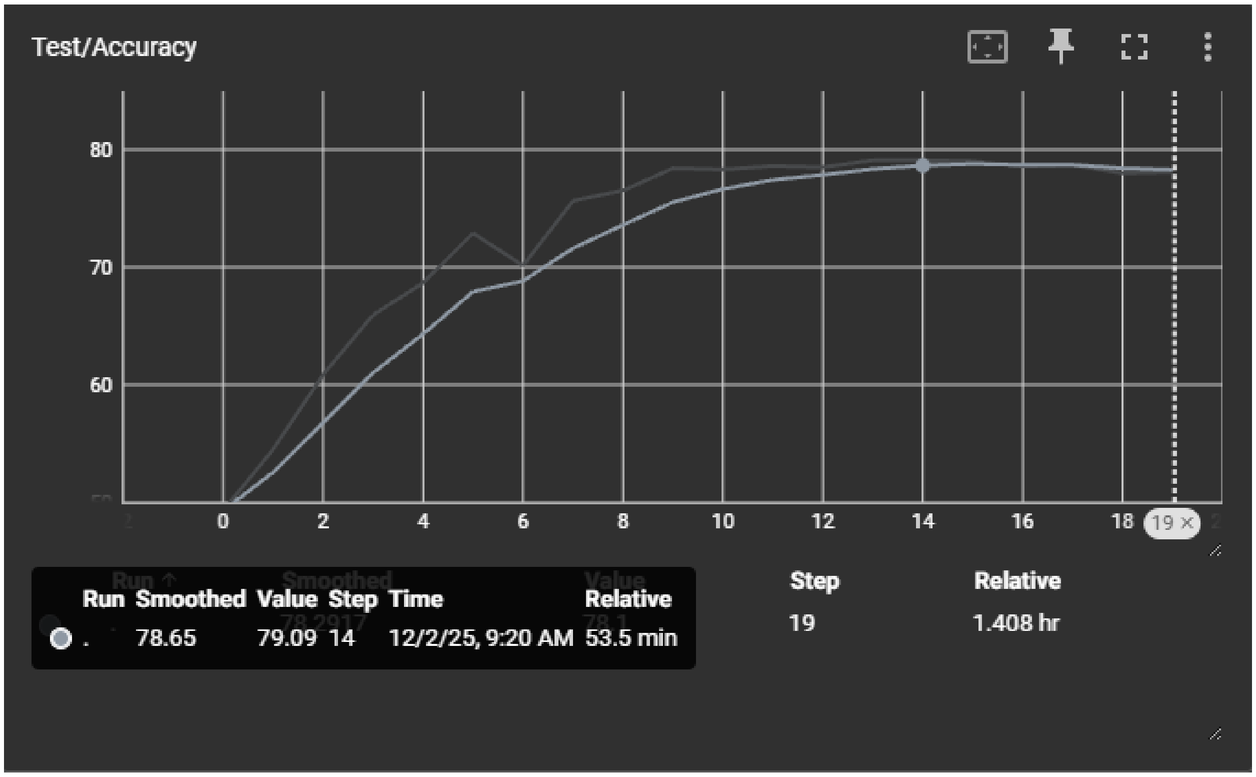
Above is the tensorboard graph of DivideMix's original code ran with 20 epochs and warmup of 10. The highest accuracy value was 83.18. An excellent value considering only 20 epochs were ran.



Run Smoothed Value Step Time Relative
● . 92.54 92.93 101 11/25/25, 9:49 AM 11.21 hr

Step Relative
119 14.09 hr

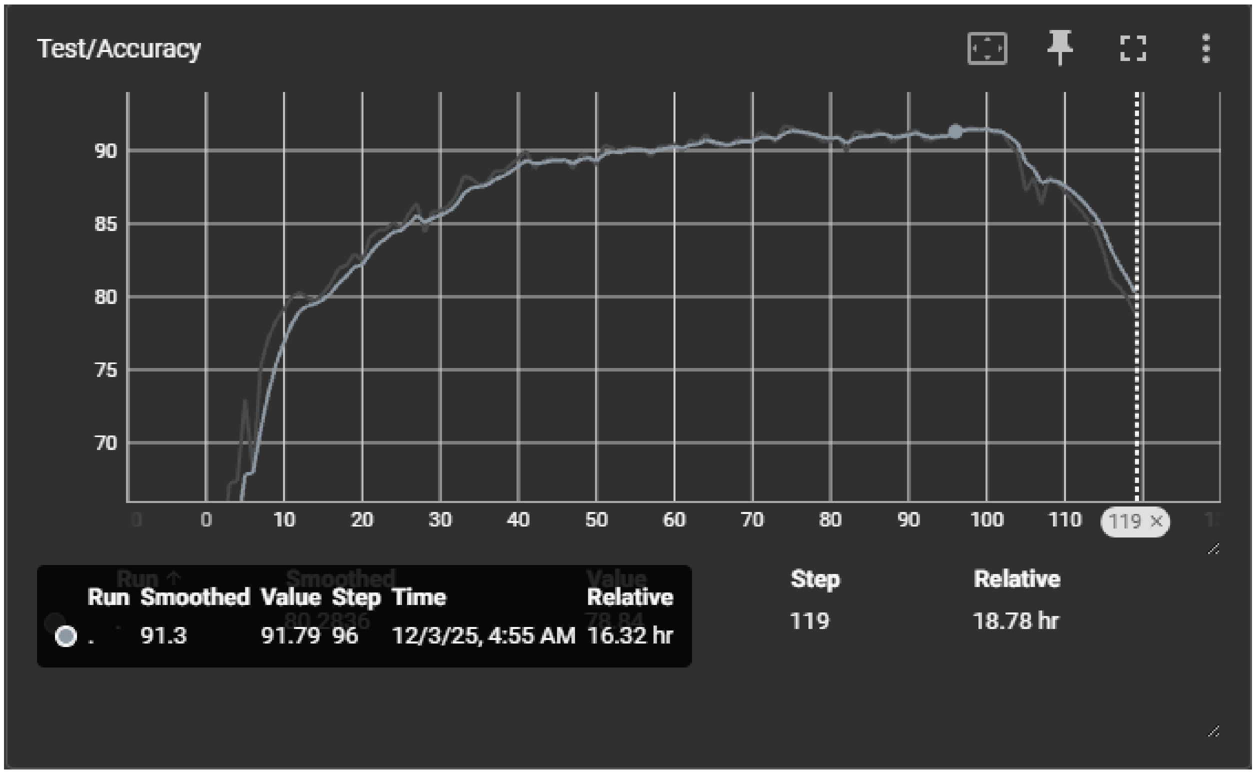
Above is the graph for DivideMix's original code ran with 120 epochs and warmup of 10. The baseline DivideMix ran for 300 epochs so the accuracy is not nearly as high as what the creators of DivideMix had, but it is still a great indicator of showing how strong DivideMix can be even at half the epochs that it runs by default. Highest accuracy was 92.93 at epoch 101. If this was run longer, then accuracy would surely go above 95%.



Run	Smoothed	Value	Step	Time	Relative
● .	78.65	79.09	14	12/2/25, 9:20 AM	53.5 min

Step	Relative
19	1.408 hr

Above is the graph of the contrastive DivideMix hybrid pipeline. Performing just a little bit under the DivideMix at 20 epochs and 10 warmup. An accuracy of 79.09 at max within the 20 epochs.



Run	Smoothed	Value	Step	Time	Relative
● .	91.3	91.79	96	12/3/25, 4:55 AM	16.32 hr

Step	Relative
119	18.78 hr

Above is the graph when my pipeline ran 120 epochs with warmup of 10. The highest accuracy being 91.79, which barely reaches DivideMix's 92.93 accuracy. However, we see a sharp decline in accuracy after 103 epochs.

Challenges

The main challenges came from deciding how the integration of Jo-SRC into the DivideMix pipeline should flow. Trying to attempt to create a version of this pipeline from scratch. After failed attempts at issues with code errors, I decided to build off the DivideMix code and implement a Jo-SRC training method defined in the main itself. Tensorboard was an issue in it of itself. Identifying which folder that the graphs saved to was a hassle to keep track and understand. The biggest challenge was waiting for the model to train, only to be met with an error after 4 hours of training and having to solve it and run the model again without knowing if we would get another error causing more wasted time. Smaller issues like having my GPU run out of VRAM to run the Jo-SRC method or a mismatch with tensors being on different hardware also made it difficult to finish epochs.

Conclusion

In this work, we investigated whether incorporating JO-SRC–style consistency refinement into the DivideMix framework could improve robustness to heavy label noise. While DivideMix already provides a strong baseline by combining MixMatch with loss-based sample partitioning, its performance can be sensitive to unstable pseudo-labels and early-stage noise amplification. Our goal was to introduce a lightweight, consistency-driven modification without altering the underlying DivideMix structure.

Our experimental results on CIFAR-10 with symmetric noise indicate that the hybrid approach offers measurable benefits in the early stages of training. In the 20-epoch setting, the JO-SRC refinement helped stabilize training and reduced noisy-label overfitting, though it did not surpass the full DivideMix baseline. In longer training runs (120 epochs), the hybrid method achieved accuracy close to DivideMix's performance while maintaining improved stability during warmup and early epochs. These findings suggest that even simple consistency-based corrections can strengthen DivideMix's pseudo-label refinement when noise is high or labels are unreliable.

However, our results also highlight important limitations. After ~103 epochs, the hybrid model showed a sharp collapse in test accuracy. This behavior reflects a known failure mode in noisy-label learning: as training progresses, both the GMM-based clean/noisy split and the consistency-based pseudo-labels can drift, causing the network to become overconfident in incorrect assignments. Once the pseudo-label distribution becomes corrupted, both the student and mean-teacher models reinforce these errors, eventually causing representation collapse. This failure emphasizes that augmenting DivideMix with additional consistency constraints does not fully resolve its vulnerability to late-stage pseudo-label drift.

Future Work

Overall, this study demonstrates that JO-SRC's divergence-based clean estimation can be effectively integrated into DivideMix with minimal changes, improving early-stage training robustness while preserving the core pipeline. At the same time, the observed collapse highlights the need for more stable long-horizon noise-robust learning strategies. Promising directions include adaptive threshold scheduling, more conservative pseudo-label sharpening, stronger regularization of the teacher model, or dynamically reducing pseudo-label reliance in later epochs. Here are some potential improvements:

- Stabilize Pseudo-label quality in later epochs
    - Reduce pseudo-label sharpening temperature over time to prevent overconfident incorrect labels
    - Cap off the confidence of teacher outputs
    - Implement uncertainty-aware pseudo-labeling
- Improve separation of clean & noisy labels
    - Use running-mean GMM stats
    - Blend DivdeMix loss with JS-divergence selection
- Prevent late-epoch collapse
    - Gradually decay the influence of pseudo-labels
    - Implement a second warmup cycle
- Strengthen Teacher-Student consistency
    - Slow down EMA decay
    - Apply strong augmentation only to student predictions
    - Stop-gradient on pseudo-label corrections
- Add robust regularization against overconfidence
    - Confidence penalty
    - Label smoothing
    - KL divergence margin loss
- Make Jo-SRC integration more adaptive
    - Use per-class divergence thresholds
    - Replace argmax OOD test with soft variations
- Training enhancements
    - Use wider backbone (ResNet-28-2)
    - Use Stochastic Weight Averaging

References

OpenAI's ChatGPT – Assistance with code implementation and formatting of equations

CIFAR-10 – Krizhevsky, Alex. **"Learning Multiple Layers of Features from Tiny Images."** Technical Report, University of Toronto, 2009. https://www.cs.toronto.edu/~kriz/cifar.html

DivideMix – Li, Junnan, Richard Socher, and Steven C.H. Hoi. **"DivideMix: Learning with Noisy Labels as Semi-supervised Learning."** *International Conference on Learning Representations (ICLR)*, 2020. https://arxiv.org/abs/2002.07394

Jo-SRC – Yao, Yuan, et al. **"Jo-SRC: A Simple and Effective Noisy Label Detection Method."** *arXiv preprint arXiv:2103.12728*, 2021. https://arxiv.org/abs/2103.12728