-Supplementary Material-

Robust Dataset Condensation using Supervised Contrastive Learning

A. Formation and Evaluation of the Golden Set

Figure 4 presents a graph depicting the training process of RDC within IDM when condensing 10 images per class from the CIFAR-10 dataset with 40% asymmetric noise, 40% symmetric noise, and 40% real-world noise (CIFAR-10N-worse dataset), using a ResNet18 backbone model. The figure tracks the progress of the metrics related to the formation of the golden set from the end of the warm-up phase at epoch 5 to the final training epoch at epoch 150.

To provide a detailed analysis, the following describes the key metrics observed in the 40% asymmetric noise dataset, located at the top line.

The first graph displays the precision and recall values of the clean set $\mathcal{C}$. The precision increases and eventually stabilizes around 0.82, while the recall continuously rises and converges near 0.9. This trend suggests that the clean set extraction process via Gaussian Mixture Model (GMM) effectively identifies clean images, contributing to the formation of a well-curated golden set.

The second graph illustrates the size dynamics of the clean set $\mathcal{C}$ and the unclean set $\mathcal{U}$ classified by GMM. The clean set steadily increases and stabilizes at approximately 32,000 samples, whereas the unclean set gradually decreases and converges around 18,000 samples. This observation indicates that a sufficient number of clean images are successfully extracted throughout the training process.

The third graph presents the number of images in the unclean set $\mathcal{U}$ that undergo relabeling due to their confidence exceeding the threshold c (0.95), along with the accuracy of the relabeled set $\mathcal{R}$. The relabeling accuracy stabilizes around 0.82, while the number of relabeled images progressively increases as training advances, ultimately reaching approximately 12,000 samples.

The fourth graph depicts the evolution of the golden set $\mathcal{G}$'s size and the leftovers, which is the unlabeled set $\mathcal{U} \setminus \mathcal{R}$. The golden set consistently expands, eventually reaching around 43,000 samples, while the unlabeled set declines and stabilizes at approximately 7,000 samples.

The middle and bottom rows of the figure which correspond to the 40% symmetric noise dataset and the CIFAR-10N-worse dataset, follow a structure similar to top row, providing a comparative analysis of how the golden set evolves under different types of noise. These findings collectively demonstrate that the golden set achieves a sufficient size while effectively filtering out noisy samples, highlighting the efficacy of the proposed noise-filtering and sample selection strategy.

B. Implementation Details

B.1. RDC Settings

B.1.1. Detailed Explanation of DivideMix Loss Utilized in RDC

The DivideMix [20] loss function consists of three main components: $\mathcal{L}_x$, $\mathcal{L}_u$, and $\mathcal{L}_{\text{reg}}$, each contributing to training stability and robustness in the presence of noisy labels.

$\mathcal{L}_x$ represents the MixUp loss, which is computed using both labeled and unlabeled data with predicted pseudo-labels due to the MixUp-based cross-entropy loss formulation. Pseudo-labels are generated by applying a sharpening function to the model's predicted logits. Since MixUp is applied to both clean and noisy samples, the model learns from interpolated inputs and labels, effectively leveraging information from both sets. This enhances generalization and robustness by encouraging the model to learn smoother decision boundaries and mitigating the impact of label noise.

$\mathcal{L}_u$ is the unsupervised consistency loss, which enforces consistency between the model's predictions on an image and its augmented counterpart. The model first generates pseudo-labels from its predicted logits, and the same process is applied to the augmented version of the image. The consistency loss is then computed as the squared difference between the pseudo-labels of the original and augmented images. This encourages the model to make stable predictions under different augmentations, thereby improving its resistance to label noise. The weight of this loss term is controlled by a hyperparameter λ_u.

$\mathcal{L}_{\text{reg}}$ is a regularization term that prevents the model from becoming overconfident in its predictions. It is based on KL divergence between the average model prediction and a uniform prior distribution over all classes. By aligning the model's output distribution with this prior, the regularization term discourages the model from making overly confident predictions on uncertain samples. The contribution of this term to overall loss is controlled by a scaling factor λ_r.

By combining these three components, the DivideMix loss effectively balances supervised learning on clean samples, consistency regularization for noisy data, and prior-based regularization to enhance robustness. The overall loss function is formulated as:

$$\mathcal{L}_{\text{DivideMix}} = \mathcal{L}_x + \lambda_u \mathcal{L}_u + \lambda_r \mathcal{L}_{\text{reg}} \tag{18}$$

B.1.2. Parameter Search for λ in Golden MixUp Contrast

We conducted a grid search to determine the optimal MixUp parameter for use in golden MixUp contrast. As shown in Table 7, we performed experiments on the CIFAR-

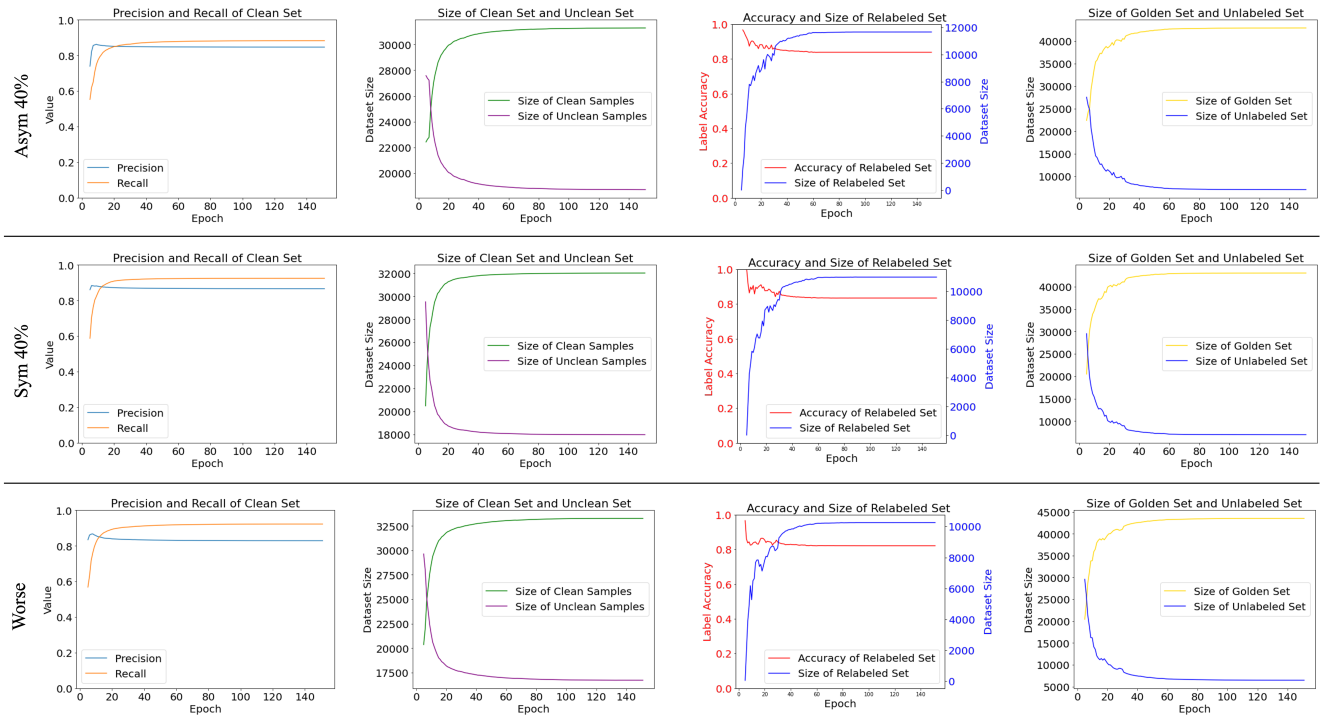

Figure 4. Progression of metrics related to the formation of the golden set when condensing the noisy CIFAR-10 dataset (40% asymmetric noise, 40% symmetric noise, CIFAR-10N-worse) with IPC 10 using ResNet18, employing RDC and IDM.

λ	Accuracy
0.25	54.17
0.50	55.13
0.75	**55.55**

Table 7. Grid search for the MixUp parameter λ in Golden MixUp Contrast on noisy CIFAR-10 (40% asymmetric noise, IPC 50) using ResNet18.

10 dataset with 40% asymmetric noise in an IPC 50 setting, evaluating λ values of 0.25, 0.5, and 0.75. The experimental results showed that when $\lambda = 0.75$, the classification accuracy of the synthetic set reached 55.55%, the highest among all tested values. Based on this finding, we set $\lambda = 0.75$ for all subsequent experiments.

B.1.3. Parameters for RDC

In all experiments, the MixUp parameter λ was set to 0.75, determined through a parameter search to optimize performance. The loss function incorporated a temperature scaling factor τ of 0.07, which is used in golden MixUp contrast. During training, a confidence threshold c (0.95) was used to filter high-confidence predictions. The model temperature value used for predicting pseudo-labels from the unlabeled set was applied with the value of 0.5, same as the value used in DivideMix. The value of λ_u, which controls the contribution of the unsupervised loss, was set to 0 for CIFAR-10, CIFAR-10N rand1, and datasets with 20% and 40% asymmetric noise, as well as those with 20% symmetric noise. For CIFAR-10 datasets with higher noise levels, including 40% symmetric noise and CIFAR-10N worse, λ_u was set to 25. For CIFAR-100, λ_u was set to 0 for the standard dataset. For datasets with 20% and 40% asymmetric noise, as well as 20% symmetric noise, λ_u was set to 25. For datasets with 40% symmetric noise and CIFAR-100N-noisy, λ_u was further increased to 125 to enforce stronger regularization.

B.2. Dataset Cleaning via DivideMix

We proposed a two-stage approach as the baseline for RDC, where dataset cleaning was first performed using DivideMix, followed by the application of a dataset condensation method. Two models used for DivideMix were both ResNet18, with a learning rate of 0.02. For pseudo-label prediction in the unlabeled set, sharpening was applied with a temperature value of 0.5. The models were trained for a total of 300 epochs. For the unsupervised loss λ_u, same values were used as Section B.1.3.

B.3. Applying RDC on IDM

This section provides the implementation details for reproducing IDM [56], as well as the details of applying RDC to IDM.

B.3.1. IDM reproduction

We performed the reproduction of IDM following its original settings, not only on clean CIFAR-10 and CIFAR-100 datasets but also on datasets containing asymmetric, symmetric, and real-world noise. Additionally, we utilized datasets refined through DivideMix.

Evaluation Setup. The model used for both training and evaluation was ResNet18. Evaluation accuracy was measured as the average of 10 independent evaluations. Each evaluation involved training a newly initialized ResNet18 model from scratch using the synthetic set and then assessing its performance on the test sets of CIFAR-10 and CIFAR-100. Each evaluation consisted of 1,000 training steps.

Model Update Strategy. During training, we initialized the process with four models. Every 30 iterations, one additional model was introduced, and once the total number of models reached 100, the oldest models were discarded to maintain a fixed number. At each iteration, two randomly selected models were trained for a total of 10 updates. The batch size for training was set to 256. The ResNet18 model used for training original dataset was trained with learning rate 0.01 using the SGD optimizer with momentum set to 0.9 and weight decay set to 0.0005. The entire training process lasted for 20,000 iterations.

Synthetic Set Update Strategy. During condensation, the synthetic set was updated based on not only the distribution loss, but also the accuracy-weighted cross-entropy loss ($Acc_\phi \mathcal{L}_{CE}(\mathcal{S})$). Specifically, when the number of images per class (IPC) was 50, a scaling factor of 0.1 was applied to the accuracy-weighted cross-entropy loss, while for IPC 1 and 10, the scaling factor was set to 0.5. For synthetic set optimization, we used the SGD optimizer with learning rate 0.2 and momentum value 0.5.

Data Augmentation Strategy. To enhance training diversity, various augmentation strategies were applied. Color transformations included random adjustments to brightness, saturation, and contrast. Spatial transformations involved random cropping, flipping, scaling, and rotation to introduce variability in image geometry. Additionally, structural modifications were implemented using random cutout, which occludes parts of the image to encourage the model to focus on more generalizable features.

B.3.2. Applying RDC to IDM

To apply RDC to IDM, we trained a single model using semi-supervised learning to properly learn clean representations from the original dataset, rather than training 100 models for short periods and frequently replacing them. Therefore, we set up a single model from the beginning and trained it throughout the entire process. Also, to apply GMM based on the loss values of every images per epoch, we converted the update count from iterations to epochs and trained RDC for a total of 150 epochs. Some model parameter settings were adjusted based on the parameters used in DivideMix. Specifically, the learning rate was set to 0.02, and the learning rate was reduced by a factor of 0.1 at every 50 epochs.

Additionally, IDM is a type of distribution matching technique that aligns the mean embeddings of the original set and the synthetic set for each class. However, if the original dataset contains noise, the mean embeddings of the original set may also be corrupted. To address this, we performed distribution matching using the mean embeddings of the golden set $\mathcal{G}$, which were extracted as described in Section 4.1.1.

B.4. Applying RDC on Acc-DD

This section presents the implementation details for reproducing Acc-DD [52], along with the specifics of applying RDC to Acc-DD.

B.4.1. Acc-DD reproduction

Pretraing Models. For pretraining in Acc-DD, we used the CIFAR-10 dataset along with datasets containing asymmetric, symmetric, and real-world noise. Additionally, we incorporated datasets that were cleaned using DivideMix to improve data quality. The model used for pretraining was ResNet18, and the number of pretraining epochs was set to 2. The augmentation strategy applied during pretraining included a combination of color transformation, cropping, cutout, flipping, scaling, and rotation.

Evaluation Setup. The evaluation was conducted using ResNet18 for both training and testing. Accuracy was averaged over 5 independent runs, where each run involved training a newly initialized ResNet18 from scratch on the synthetic CIFAR-10 dataset. The trained model was then evaluated on the CIFAR-10 test set. Each evaluation consisted of 1,000 training steps.

Model Update Strategy. The model was trained using stochastic gradient descent (SGD) with a learning rate of 0.01, momentum of 0.9, and weight decay of 0.0005. The learning rate for the model remained fixed throughout training. The condensation process ran for 500 iterations, with each iteration containing 100 inner-loop updates. Each batch of real images contained 64 samples. The dataset was iteratively refined over 2000 real samples per iteration, with one training epoch per network update. To ensure reproducibility, the entire process was repeated five times.

Algorithm 1: Robust Dataset Condensation (RDC)

Input: $\mathcal{T}$: original noisy dataset. f_θ: base model trained on $\mathcal{T}$. T_{warmup}: model warm-up epoch.
Initialize: Train base model f_θ on $\mathcal{T}$ for T_{warmup}.
for iteration $\leftarrow T_{\text{warmup}}$ *to* max_iteration **do**
  Update θ_{RDC} by Eq. (16)
  Define the golden set $\mathcal{G}$ by Eq. (10)
  Make MixUp augmented synthetic set $\mathcal{S}_{\text{MixUp}}$ using $\mathcal{S}$ and $\mathcal{G}$ by Eq. (11)
  Calculate $\mathcal{L}_{\text{GMC}}$ using $\mathcal{S}_{\text{MixUp}}$ by Eq. (14)
  Update $\mathcal{S}$ by Eq. (17)
Output: Robust synthetic dataset $\mathcal{S}_{\text{RDC}}$

Synthetic Set Update Strategy. The synthetic dataset was optimized separately with a learning rate of 0.01 and a momentum of 0.5 to allow for gradual refinement. A multi-scale condensation strategy was applied, where the factor parameter was set to 2, meaning that images were divided into smaller patches and reconstructed to retain different levels of detail. The synthetic batch size was set to a maximum of 128. Additionally, the synthetic dataset was refined through a gradient-based matching strategy, ensuring that the gradients of synthetic and real images were aligned. Mean squared error (MSE) was used as the loss metric for this matching process.

Data Augmentation Strategy. Differentiable data augmentation was applied using color transformation, cropping, cutout, flipping, scaling, and rotation to improve generalization.

B.4.2. Applying RDC to Acc-DD

Since Acc-DD matches the gradient values of a model trained on the original dataset in the early epochs with the gradient of a model trained on the synthetic set, training a single model for an extended period results in diminishing gradient magnitudes in later stages, making effective condensation infeasible. To address this, instead of training a single model throughout the entire process, we applied RDC for a limited number of initial epochs and immediately retrained a new model to maintain effective gradient matching. Given that the pretraining phase consisted of 2 epochs, we trained the model on the original dataset for 3 epochs during the condensation process and applied RDC for 2 epochs. This process was repeated every 5 epochs.

C. Algorithm of Robust Dataset Condensation

Algorithm 1 describes the full pipeline of Robust Dataset Condensation (RDC).

Method	IDM			IDM + RDC		
CIFAR-10 \Metrics	HLR	IOR	LRS	HLR	IOR	LRS
Clean	50.27	22.82	-0.1565	48.09	25.00	–0.1565
Asymm. 40%	28.2	11.08	–0.1564	17.46	21.82	3.607
Symm. 40%	23.62	22.81	0.1272	18.56	22.81	3.407
Real. 40%	29.00	18.17	–0.1547	20.84	26.33	6.464

Table 8. Evaluation of RDC using the **HLR, IOR, and LRS metrics** from the Dd-Ranking benchmark on CIFAR-10 under various noise types.

D. Evaluation with Dd-Ranking Benchmark

Table 8 presents the evaluation of RDC using three metrics from the Dd-Ranking benchmark [22]: hard label recovery (HLR), improvement over random (IOR), and label-robust score (LRS). Applying RDC consistently outperforms the baseline across all noise settings, demonstrating its robustness and effectiveness. Lower HLR, higher IOR, and higher LRS indicate better performance. For LRS, the weighting parameter λ is set to 0.5.

E. RDC with Soft Labels

Let the number of classes be c, and the dataset be defined as $\mathcal{T} = \{(x_1, y_1), \ldots, (x_{|\mathcal{T}|}, y_{|\mathcal{T}|})\}$, where each label $y_i \in \mathbb{R}^c$ is a soft label represented as a probability vector $y_i = [y_i^{(1)}, y_i^{(2)}, \ldots, y_i^{(c)}]$ where $\sum_{j=1}^{c} y_i^{(j)} = 1$. Here, $y_i^{(c)}$ denotes the probability that sample x_i belongs to class c.

Since every sample contains probability values over all classes, explicitly defining positive and negative pairs is not meaningful. Instead, we compute similarity between the anchor and all other samples, and weight the contribution based on the product of the class probabilities from the soft labels.

Let the set of all samples excluding the anchor (x_i, y_i) be: $A = \{(x_a, y_a) \in \mathcal{T} \mid x_a \neq x_i\}$. Then, the supervised contrastive loss with soft labels is defined as:

$$\mathcal{L}_{\text{SupCon}}(\mathcal{T}, A) = \sum_{x_i \in \mathcal{T}} \sum_{c=1}^{C} \left(-y_i^{(c)} \sum_{x_a \in A} y_a^{(c)} \cdot \log \frac{\exp\left(f_\theta(x_i) \cdot f_\theta(x_a)/\tau\right)}{\sum_{x_a \in A} \exp\left(f_\theta(x_i) \cdot f_\theta(x_a)/\tau\right)} \right) \tag{19}$$

This formulation removes the need to explicitly define positive sets. Instead, similarity is computed for all samples in the denominator, weighted by the soft label probabilities. As all samples are used in the denominator, normalization by the size of the positive set (e.g., dividing by $|P|$) is considered unnecessary and therefore omitted.

F. Applying RDC to MTT and DATM

In Table 9, we evaluate applying RDC to MTT [2] and DATM [11], two representative trajectory matching-based condensation methods. We synthesize 10 images per class on noisy CIFAR-10 using ResNet18. Both MTT and DATM

CIFAR-10 (10 Img/Cls)	Clean	Asymm. 40%	Symm. 40%	Real. 40%
Random	22.28	19.53	18.36	20.19
MTT	38.26	27.42	35.11	36.39
MTT + Two-stage	37.40	32.36	36.53	36.71
MTT + RDC (Ours)	**38.83**	**33.61**	**40.02**	**40.11**
DATM	**42.74**	24.24	23.75	25.66
DATM + Two-stage	40.60	22.08	38.86	35.64
DATM + RDC (Ours)	41.59	**34.54**	**41.21**	**39.07**
Whole Dataset	95.37	58.81	64.79	67.36

Table 9. Robustness comparison among different methods using **MTT and DATM as the base model** on CIFAR-10 under various noise types.

are highly sensitive to label noise, and although the two-stage approach provides some performance recovery, it remains insufficient across all noise settings.

In contrast, integrating RDC into both MTT and DATM leads to substantial performance recovery. In the symmetric and real noise settings, the performance closely approaches that of the clean-data baseline. In the asymmetric noise setting, RDC yields significant improvements over the original methods. These results demonstrate the adaptability of RDC to trajectory matching methods. In particular, they highlight its effectiveness even when applied to soft-label methods such as DATM.

G. Latency of RDC over Two-Stage Approach

To demonstrate the resource efficiency of the two-stage approach, we compared the time consumption of RDC with a two-stage approach that performs data cleaning prior to applying the base method. This experiment was conducted using a single A5000 GPU, with IDM employed as the base condensation method. The dataset used was CIFAR10N-worse, which contains approximately 40% real noise, and condensation was performed with 10 images per class. DivideMix was utilized as the data cleaning method.

The two-stage approach required 18.83 hours in total, with the data cleaning process taking 6.78 hours and the condensation process taking 12.05 hours. In contrast, RDC completed the entire process in 14.03 hours, demonstrating a time efficiency improvement of approximately 5 hours. This result highlights not only the time efficiency of RDC but also its effectiveness in achieving performance comparable to or even superior to that of condensation conducted on a fully clean dataset, thereby validating its efficiency and practicality.

H. Visualization of RDC

We can verify the effectiveness of RDC through its visualization. The following figures present the overall results in four scenarios: (Fig 5) condensing clean CIFAR-10 using Acc-DD, (Fig 6) applying Acc-DD on CIFAR-10 with 40% asymmetric noise, (Fig 7) using the two-stage approach of DivideMix for data cleaning followed by condensing CIFAR-10 with 40% asymmetric noise using Acc-DD, and (Fig 8) applying RDC to Acc-DD on CIFAR-10 with 40% asymmetric noise. In all scenarios, 10 images per class were condensed. A detailed explanation can be found in Section 5.5 in the main paper.

Figure 5. Visualization of Condensed Images of CIFAR-10, IPC 10 using Acc-DD

Figure 6. Visualization of Condensed Images of CIFAR-10 with 40% asymmetric noise, IPC 10 using Acc-DD

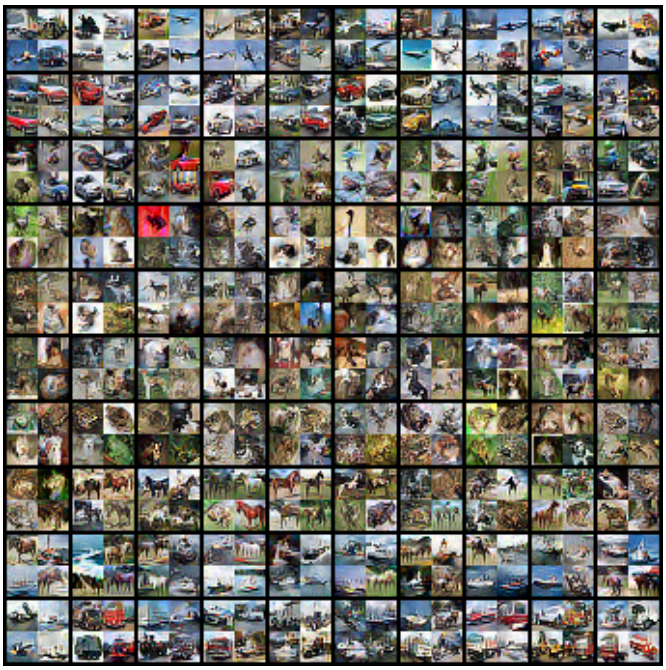

Figure 7. Visualization of Condensed Images of CIFAR-10 with 40% asymmetric noise, IPC 10 using data cleaning and Acc-DD

Figure 8. Visualization of Condensed Images of CIFAR-10 with 40% asymmetric noise, IPC 10 using RDC and Acc-DD