Supplementary Material
Enhancing Dataset Distillation via Non-Critical Region Refinement

Minh-Tuan Tran[1], Trung Le[1], Xuan-May Le[2], Thanh-Toan Do[1], Dinh Phung[1]
[1]Monash University, [2]The University of Melbourne
{tuan.tran7,trunglm,toan.do,dinh.phung}@monash.edu
xuanmay.le@student.unimelb.edu.au

A. Training Details

For all experiments conducted in this study, we fixed the following hyperparameters:

- Scale factor for batch normalization $\alpha_{bn} = 10$ (as follows to [2])
- CAM matrix's upper threshold $\epsilon = 0.5$
- Embedding radius $r = 0.4$
- Scale factors for distance-based representation $\alpha_{dbr} = 1$ and $\alpha_{lr} = 1$

These values were selected through careful tuning based on prior work and were maintained consistently across all experiments to ensure comparability and minimize the impact of hyperparameter variability. By fixing these hyperparameters, we aim to provide a fair and reproducible assessment of our methods across different datasets and experimental conditions.

A.1. Non-Critical Region Refinement Phase.

During this phase, we utilized the following settings:

- Optimizer: Adam
- Learning rate: 0.05
- Betas: $\beta_1 = 0.5$, $\beta_2 = 0.9$
- Batch size: 100
- Iterations: 2000

This phase is designed to refine the synthetic data, particularly in non-critical regions of the feature space, enhancing the model's ability to generalize for subsequent tasks.

A.2. Knowledge Transfer and Post-Evaluation Phase.

In this phase, we applied the following parameters:

- Optimizer: AdamW (incorporating weight decay for better generalization)
- Learning rate: 1e-3
- Batch size: 100
- Training epochs: 300
- Learning rate scheduler: Smoothing LR

To augment the synthetic data and improve robustness, we employed the following data augmentation techniques:

- Two RandAugment transformations
- RandomResizeCrop
- RandomHorizontalFlip

These augmentations, as detailed in [1] and [2], help prevent overfitting by introducing variability into the data, thereby enhancing the model's generalization capabilities.

All of these settings were consistently applied across all datasets to ensure fair comparison and reliable evaluation of the model's performance under varying experimental conditions.

B. Parameter Sensitivity Analysis

B.1. Threshold ϵ.

In this section, we examine the impact of different threshold values (ϵ) on model performance across two datasets, ImageNette and CIFAR100, for both IPC 10 and IPC 50 classification tasks. The results presented in Table 1 show that a threshold of 0.5 consistently delivers the best performance across all settings. Specifically, for ImageNette (IPC 10), the accuracy reaches 66.2%, and for ImageNette (IPC 50), it peaks at 85.6%. Similarly, for CIFAR100 (IPC 10), the highest accuracy is 62.7%, while CIFAR100 (IPC 50) achieves 67.1%. This optimal performance at $\epsilon = 0.5$ can be attributed to a balanced trade-off, where this threshold maintains an effective level of value retention and updates without excessive loss or staleness.

Thresold ϵ	0.1	0.3	0.4	0.5	0.6	0.7	0.9
ImageNette (IPC 10)	63.15	65.35	65.59	**66.2**	65.87	65.4	65.35
ImageNette (IPC 50)	82.89	84.66	85.01	**85.6**	85.08	84.7	84.31
CIFAR100 (IPC 10)	60.14	61.56	62.28	**62.7**	61.29	61.69	61.14
CIFAR100 (IPC 50)	66.58	65.5	66.46	**67.1**	66.26	66.7	66.15

Table 1. Comparison of model performance across different threshold values (ϵ) on the ImageNette and CIFAR100 datasets for IPC 10 and IPC 50 classification tasks. The best performance is achieved at $\epsilon = 0.5$ for both datasets and tasks.

B.2. Radius r.

In this section, we investigate the impact of different radius values on model performance across the ImageNette and CIFAR10 datasets for both IPC 10 and IPC 50 classification tasks. The results in Table 2 reveal that a radius of 0.4 consistently provides the best performance across all settings. Specifically, for ImageNette (IPC 10), the accuracy peaks at 66.2%, and for ImageNette (IPC 50), it reaches 85.6%. Similarly, for CIFAR10 (IPC 10), the highest accuracy is 62.7%, while CIFAR10 (IPC 50) achieves 67.1%. This suggests that a radius of 0.4 strikes an optimal balance, yielding the highest accuracy without causing overfitting or excessive loss.

radius	0.1	0.2	0.3	0.4	0.5	0.6	0.7
ImageNette (IPC 10)	64.29	65.24	65.24	**66.2**	64.92	65.6	64.91
ImageNette (IPC 50)	84.17	83.41	84.21	**85.6**	84.14	85.24	85.05
CIFAR10 (IPC 10)	61.02	61.21	61.97	**62.7**	61.18	61.16	59.56
CIFAR10 (IPC 50)	65	65.1	66.02	**67.1**	65.74	66.57	65.74

Table 2. Comparison of model performance across different radius values on the ImageNette and CIFAR10 datasets for IPC 10 and IPC 50 classification tasks. A radius of 0.4 consistently yields the best performance across all settings.

B.3. Scale Factor α_{dbr} and α_{lr}.

In this section, we analyze the effect of two hyperparameters, α_{dbr} and α_{lr}, on model performance across different configurations. The results, as shown in Table 3, indicate that the best performance is achieved when $\alpha_{dbr} = 1$ and $\alpha_{lr} = 1$, yielding an accuracy of **66.2**%. This combination outperforms other configurations, particularly for higher values of α_{lr}, where the accuracy tends to decrease. These findings suggest that the balance between these two parameters plays a critical role in optimizing the model's performance.

α_{lr} α_{dbr}	0.2	0.5	1	2	5
0.2	64.2	64.5	64.4	64.4	62.7
0.5	63.7	65.0	65.6	65.0	62.6
1	63.8	65.8	**66.2**	65..9	63.2
2	64.7	66.1	65.6	64.3	62.8
5	64.5	63.6	63.8	63.9	63.0

Table 3. Performance analysis of the model with varying values of α_{dbr} and α_{lr}. The highest accuracy of **66.2**% is achieved when $\alpha_{dbr} = 1$ and $\alpha_{lr} = 1$.

B.4. Ablation Study on L_{bn}:

Inspired by your recommendations, we conducted additional experiments in the Table 4. The results show that: (1) With or without L_{bn}, our method outperforms RDED; (2) L_{bn} is important and helps improve model performance.

	CIFAR10	CIFAR100	ImageNette	ImageNet1k
RDED	37.1 ± 0.3	42.6 ± 0.2	61.4 ± 0.4	42.0 ± 0.1
NRR-DD Without L_{bn}	68.3 ± 0.4	60.1 ± 0.2	65.1 ± 0.5	44.8 ± 0.2
NRR-DD With L_{bn}	72.2 ± 0.4	62.7 ± 0.2	66.2 ± 0.6	46.1 ± 0.2

Table 4. Ablation Study for L_{bn} with 10 IPC.

C. Choosing Lowest Confident Score or Highest Confident Score.

To further highlight the benefits of using the Lowest Confident Score (LCS) over the Highest Confident Score (HCS) for selecting initial patches, we compare two experimental scenarios: (1) training the combined images of these patches directly (referred to as Direct) to evaluate the raw image quality, and (2) refining the images through our Non-Critical Refinement (Referred to as Refining) to assess the improved versions. The results, presented in Table 5, clearly indicate that the Lowest Confident Score consistently yields better performance than the Highest Confident Score across all evaluated settings. These observations can be attributed to two main factors:

- **Hard-to-Learn Samples:** The Lowest Confident Score tends to identify harder-to-learn, more challenging samples that are often neglected by simpler approaches. These samples provide critical information that enhances the model's ability to generalize, which leads to improved performance across different datasets. Additionally, by selecting these harder samples, the model still focus on instance-specific features, which is the most important features in large-scale dataset distillation.

- **Optimization Flexibility with CE Loss:** When using the teacher's classification loss (CE Loss) on synthetic data, the Lowest Confident Score offers more flexibility for parameter updates compared to the Highest Confident Score. This is because patches with

	ConvNet			Resnet18		
	CIFAR10	CIFAR100	ImageNet1k	CIFAR10	CIFAR100	ImageNet1k
Highest-Score (Direct)	50.2 ± 0.3	48.1 ± 0.3	20.4 ± 0.1	37.1 ± 0.3	42.6 ± 0.2	42.0 ± 0.1
Lowest-Score (Direct)	**51.3 ± 0.4**	**50.2 ± 0.3**	**21.2 ± 0.3**	**50.4 ± 0.5**	**51.3 ± 0.2**	**43.2 ± 0.2**
Highest-Score (Refining)	64.8± 0.4	53.1 ± 0.3	24.1 ± 0.3	63.7 ± 0.2	57.1 ± 0.3	49.9 ± 1.1
Lowest-Score (Refining)	**66.7 ± 0.4**	**55.7 ± 0.2**	**25.6 ± 0.2**	**65.1 ± 0.3**	**58.3 ± 0.2**	**51.3 ± 1.0**

Table 5. Comparison of the performance using Lowest Confident Score (LCS) versus Highest Confident Score (HCS) for selecting initial patches. The results show that LCS outperforms HCS in all scenarios, both in direct training and after applying Non-Critical Refinement (Refining), highlighting the advantages of LCS in improving model performance.

> the Highest Confident Score typically have a CE Loss close to zero in many cases, indicating that these samples are already well-classified and do not require significant adjustments. In contrast, the Lowest Confident Score corresponds to harder-to-learn patches, which are more likely to lead to meaningful updates during training. By focusing on these difficult samples, the model has more opportunity for optimization, thus facilitating better overall performance.

Thus, the results demonstrate that using the Lowest Confident Score for selecting patches not only improves the model's performance but also makes the training process more effective. By focusing on hard-to-learn samples, LCS enables the model to update parameters more significantly, yielding better performance both in direct training and after refinement. The effectiveness of LCS in this context underscores its importance as a selection criterion for initial patches.

D. Further Discussion

Balancing instance-level vs class-level features: Thank you for your comment. We've added a discussion in the revised paper, explaining the effect of instance- and class-level information by using different value of ϵ in Eq. 4 (higher ϵ emphasizes lower instance-level and higher class-level information). The table below shows optimal performance at $\epsilon = 0.5$, balancing both levels. Qualitative images with different ϵ values are also included for illustration.

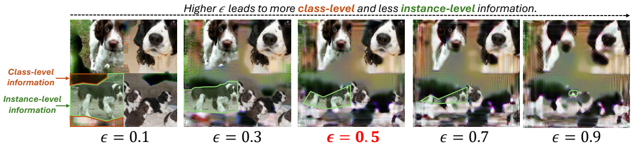
Higher ϵ leads to more class-level and less instance-level information.
Class-level information
Instance-level information
$\epsilon = 0.1$
$\epsilon = 0.3$
$\epsilon = 0.5$
$\epsilon = 0.7$
$\epsilon = 0.9$

Figure 1. Visualization with various value of ϵ.

Thresold ϵ	0.1	0.3	0.4	**0.5**	0.6	0.7	0.9
CIFAR100 (IPC 10)	60.14	61.56	62.28	**62.7**	61.29	61.69	61.14
CIFAR100 (IPC 50)	66.58	65.5	66.46	**67.1**	66.26	66.7	66.15

Table 6. Accuracies with various value of ϵ.

Efficiency test: The experiment in Table 7 shows our method has lower time cost and similar memory to SRe2L. However, due to image refinement, both time and peak memory are higher than RDED.

	Resnet18			MobileNet-V2		
Architecture	SRe2L	RDED	Our	SRe2L	RDED	Our
Time Cost (ms)	2113.23	39.89	520.65	3783.16	64.97	989.46
Peak Memory (GB)	9.14	1.57	9.14	12.93	2.35	12.93

Table 7. Efficiency test in generating 100 images in ImageNet1k.

Neural architecture search (NAS): Due to time constraints, we follow the setting in the DM paper and run experiment in Table 8. Results show that our synthetic data achieves better performance than previous methods.

	Random	DSA	DM	RDED	Our	Whole dataset
Performance (%)	84.0	82.6	84.3	84.6	84.9	85.9
Correlation	-0.04	0.68	0.76	0.78	0.80	1.00
Time cost (min)	142.6	142.6	142.6	142.6	142.6	3580.2

Table 8. NAS on CIFAR-10 (50 IPC) searching for 720 ConvNets.

MMT's Initialization: Table 9 shows results with MMT-initialized images. Our method only yields slight improvements, as these images already reside in high-confidence regions and contain class-general information. The near-zero L_C leads to minimal refinement, and their limited fine-grained details keep performance relatively low.
Real Data Initialization: On the other hand, real data initialization performs well, achieving SOTA results, though slightly lower than our method. This is due to the strong instance-specific information in real images, with our refining method adding class-general features.

	MMT (Baseline)	MMT's Init	Real Image	RDED's Init	NRR-DD's Init
CIFAR100 (10 IPC)	40.1 ± 0.4	40.8 ± 0.6	53.5 ± 0.3	53.8 ± 0.3	55.7 ± 0.2
CIFAR100 (50 IPC)	47.7 ± 0.2	48.0 ± 0.3	59.5 ± 0.1	59.6 ± 0.1	61.1 ± 0.1

Table 9. Comparing various initializations.

References

[1] Peng Sun, Bei Shi, Daiwei Yu, and Tao Lin. On the diversity and realism of distilled dataset: An efficient dataset distillation paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9390–9399, 2024. 1

[2] Zeyuan Yin, Eric Xing, and Zhiqiang Shen. Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective. *Advances in Neural Information Processing Systems*, 36, 2024. 1