# Emphasizing Discriminative Features for Dataset Distillation in Complex Scenarios

## Supplementary Material

We organize our supplementary material as follows:

**Algorithm of EDF:**

- Appendix 9: Pseudo code of EDF with detailed explanation.

**Experimental Settings:**

- Appendix 10.1: Training recipe.
- Appendix 10.2: Evaluation recipe.
- Appendix 10.3: Computing resources required for different settings.

**Additional Experimental Results and Findings:**

- Appendix 11.1: Cross-architecture evaluation.
- Appendix 11.2: Results of distilled datasets without knowledge-distillation-based evaluation.
- Appendix 11.3: Distorted synthetic images under excessive enhancement factors.
- Appendix 11.4: The changing trend of discriminative areas in EDF distillation.
- Appendix 11.5: The impact of using different models to extract activation maps.

**Comp-DD Benchmark**

- Appendix 12.1: Subset details of the Comp-DD benchmark.
- Appendix 12.3: Hyper-parameters of the Comp-DD benchmark.
- Appendix 12.2: More clarifications on the complexity metrics.

**Visualization**

- Appendix 13: Visualization of EDF distilled images.

**Related Work**

- Appendix 14: More related work of dataset distillation.

## 9. Algorithm of EDF

Algorithm 1 provides a pseudo-code of EDF. Lines 1-7 specify inputs of the EDF, including a trajectory-matching algorithm $\mathcal{A}$, the model for Grad-CAM $\mathcal{G}$, the frequency of activation map update $K$, the supervision dropout ratio $\alpha$, the enhancement factor $\beta$, the activation map processing function $\mathcal{F}$, and the number of distillation iterations $T$.

Lines 12-14 describe the Common Pattern Dropout module. After we obtain the trajectory matching losses from $\mathcal{A}$, we sort them in ascending order to get ordered losses. Then, the smallest $\alpha|L|$ elements are dropped as they introduce non-discriminative common patterns.

Lines 15-19 describe the Discriminative Area Enhancement module. For every $K$ iterations, we update activation maps of synthetic images. The gradients of synthetic images are then processed by the function $\mathcal{F}$ (see Equation 4 for the computation). Finally, synthetic images are updated biasedly towards discriminative areas.

**Algorithm 1** Emphasizing Discriminative Features

```
Input: $D_{real}$: The real dataset
Input: $D_{syn}$: The synthetic dataset
Input: $\mathcal{A}$: A trajectory-matching based algorithm
Input: $\mathcal{G}$: Grad-CAM model
Input: $K$: Activation maps update frequency
Input: $\alpha$: Threshold of supervision dropout
Input: $T$: Total distillation steps
Input: $\beta$: Enhancement factor
Input: $\mathcal{F}$: Activation map processing function
Input: $r$: Learning rate of synthetic dataset

1:  for $t$ in $0 \ldots T-1$ do
2:      $L \leftarrow \mathcal{A}(D_{syn}, D_{real})$    ▷ Compute the array of trajectory matching losses
3:      $L' \leftarrow Sort(L)$    ▷ Sort $L$ to get ordered losses
4:      $L_{edf} \leftarrow \sum_{i=\alpha|L|}^{|L|} L'_i$    ▷ Dropout low-loss supervision
5:      if $t \bmod K = 0$ then
6:          $M \leftarrow \mathcal{G}(D_{syn})$    ▷ Update activation maps of current $S$
7:      end if
8:      $(\nabla D_{syn})_{EDF} \leftarrow \nabla D_{syn} \circ \mathcal{F}(M, \beta)$    ▷ Process synthetic image gradients
9:      $D_{syn} \leftarrow D_{syn} - r \cdot (\nabla D_{syn})_{EDF}$    ▷ Biased update towards discriminative areas
10: end for
11: Return $D_{syn}$
```

## 10. Experimental Settings

### 10.1. Training Details

We follow previous trajectory matching works [8, 13, 25] to train expert trajectories for one hundred epochs. Hyper-parameters are directly adopted without modification. For distillation, we implement EDF based on DATM [13] and PAD [25], which simultaneously distills soft labels along with images.

We use torch-cam [11] for Grad-CAM implementation. Hyper-parameters are listed in Table 9.

### 10.2. Evaluation Details

To achieve a fair comparison, when comparing EDF with DD methods, we only adopt the set of differentiable augmentations commonly used in previous studies [1, 68, 69] to train a surrogate model on distilled data and labels.

When comparing EDF with DD+KD methods, we follow their evaluation methods, which we detail the steps as follows:

1. Train a teacher model on the real dataset and freeze it afterward.
2. Train a student model on the distilled dataset by **minimizing the KL-Divergence loss** between the output of the student model and the output of the teacher model on the same batch from distilled data.
3. Validate the student model on the test set and obtain test accuracy.

For implementation, please refer to the official repo of SRe2L and RDED.

### 10.3. Computing Resources

Experiments on IPC 1/10 can be run with 4x Nvidia-A100 80GB GPUs, and experiments on IPC 50 can be run with 8x Nvidia-A100 80GB GPUs. The GPU memory demand is primarily dictated by the volume of synthetic data per batch and the total training iterations the augmentation model undergoes with that data. When IPC becomes large, GPU usage can be optimized by either adopting techniques like TESLA [6] or by scaling down the number of training iterations (”syn_steps”) or shrinking the synthetic data batch size (”batch_syn”).

## 11. Additional experiment results and findings

### 11.1. Cross-architecture Evaluation

Generalizability on different model architectures is one key property of a well-distilled dataset. To show that EDF can generalize well on different models, we evaluate synthetic images under IPC 10 and 50 of the ImageSquawk subset, on three other standard models, AlexNet [20], VGG11 [45], and ResNet18 [14]. As shown in Table 14, our distilled datasets outperform random selection and two baseline methods on both IPC10 and IPC50. Compared with IPC10, distilled images under IPC50 can achieve better performance on unseen neural networks. This suggests that EDF's distillation results have decent generalizability across different architectures, especially when the compressing ratio is smaller which allows distilled datasets to accommodate more discriminative information.

### 11.2. Eval. without Knowledge Distillation

Starting from [54], representative dataset distillation (DD) methods [1, 51, 69, 71] establish a general workflow as follows: 1) *Distillation*: At this stage, information from the real dataset is fully accessible to the DD algorithm to train synthetic data. 2) *Evaluation*: After the distilled dataset is obtained, the evaluation is performed by training a randomly initialized model on the distilled data. Specifically, in the context of classification, the objective is to minimize cross-entropy loss. Recently, some new methods [48, 64, 73, 74] introduced teacher knowledge into the student model by applying knowledge distillation. Although it helps improve performances to a large extent, it may not be able to reflect the effectiveness of dataset distillation accurately.

To this end, we remove the knowledge distillation from Eval. w/ Knowledge Distillation (SRe2L and RDED) methods but keep soft labels to ensure a fair comparison, Specifically, we train a classification model on the synthetic images by only minimizing the cross-entropy loss between student output and soft labels. As shown in Table 11, without knowledge distillation, EDF outperforms SRe2L and RDED in 8 out of 9 settings. Our advantage is more pronounced, especially when IPC is smaller, underscoring the superior efficacy of EDF on smaller compressing ratios.

### 11.3. Distorted Images of Large Enhancement Factor

In Figure 6, we show results of using excessively large enhancement factors as mentioned in Section 4.3. The distributions of these distilled images are distorted, with many pixels containing only blurred information. This occurs because excessively increasing the gradients in discriminative areas can lead to large updates between iterations, resulting in the divergence of the pixel distribution. Therefore, the enhancement of discrimination areas is not the stronger the better. It is important to maintain the enhancement factor within a reasonable range.

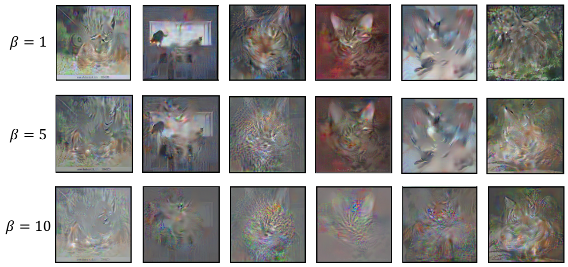


Figure 6. Distorted image distributions due to excessively large enhancement factors (= 10)

https://github.com/VILA-Lab/SRe2L/tree/main/SRe2L
https://github.com/LINs-lab/RDED

| Modules |  | CPD | DAE |  | TM |  |  |  |
|---|---|---|---|---|---|---|---|---|
| Hyper-parameters |  | $\alpha$ | $\beta$ | $K$ | $T$ | batch_syn | lr_pixel | syn_steps |
| ImageNette | 1 | 0 | 1 | 50 | 10000 | 1000 | 10000 | 40 |
|  | 10 | 0.25 | 1 | 100 |  | 250 | 1000 | 40 |
|  | 50 | 0.375 | 1 | 200 |  | 100 | 100 | 80 |
| ImageWoof | 1 | 0 | 1 | 50 | 10000 | 1000 | 10000 | 40 |
|  | 10 | 0.25 | 1 | 100 |  | 250 | 1000 | 40 |
|  | 50 | 0.375 | 2 | 200 |  | 100 | 100 | 80 |
| ImageMeow | 1 | 0 | 1 | 50 | 10000 | 1000 | 10000 | 40 |
|  | 10 | 0.25 | 1 | 100 |  | 250 | 1000 | 40 |
|  | 50 | 0.375 | 2 | 200 |  | 200 | 100 | 40 |
| ImageYellow | 1 | 0 | 1 | 50 | 10000 | 1000 | 10000 | 40 |
|  | 10 | 0.25 | 1 | 100 |  | 250 | 1000 | 40 |
|  | 50 | 0.375 | 1 | 200 |  | 200 | 100 | 40 |
| ImageFruit | 1 | 0 | 1 | 50 | 10000 | 1000 | 10000 | 40 |
|  | 10 | 0.25 | 1 | 100 |  | 250 | 1000 | 40 |
|  | 50 | 0.375 | 1 | 200 |  | 200 | 100 | 40 |
| ImageSquawk | 1 | 0 | 1 | 50 | 10000 | 1000 | 10000 | 40 |
|  | 10 | 0.25 | 1 | 100 |  | 250 | 1000 | 40 |
|  | 50 | 0.375 | 2 | 200 |  | 100 | 100 | 80 |

Table 9. Hyper-parameters of experiments on ImageNet nette, woof, meow, fruit, yellow, squawk subsets.

| Method | ConvNetD5 | ResNet18 | VGG11 | AlexNet |
|---|---|---|---|---|
| Random | 41.8 | 40.9 | 43.2 | 35.7 |
| FTD | 62.8 | 49.8 | 50.5 | 47.6 |
| DATM | 65.1 | **52.4** | 51.2 | **49.6** |
| **EDF** | **68.2** | 50.8 | **53.2** | 48.2 |

(a) ImageYellow, IPC10

| Method | ConvNetD5 | ResNet18 | VGG11 | AlexNet |
|---|---|---|---|---|
| Random | 29.6 | 31.4 | 30.8 | 25.7 |
| FTD | 58.4 | 55.6 | 57.6 | 52.3 |
| DATM | 61.8 | 62.8 | **65.6** | 63.5 |
| **EDF** | **65.4** | **63.6** | 64.8 | **69.2** |

(b) ImageSquawk, IPC50

Table 10. Cross-architecture evaluation on ResNet18, VGG11, and AlexNet. ConvNetD5 is the distillation architecture. Distilled datasets under IPC10 and IPC50 outperform random selection, FTD, and DATM, showing good generalizability.

### 11.4. Changing Trends of Discriminative Areas

Figure 5b demonstrates that EDF effectively expands the discriminative areas (high-activation regions) on several image samples at a fixed distillation iteration. In Figure 7a, we show the changing trend of discriminative areas of 5 different classes across 10000 iterations (sampled every 500 iterations). Despite the fluctuation, these areas expand as the distillation proceeds. This further confirms that one key factor in the success of EDF is that it successfully increases discriminative features in synthetic images and turns them into more informative data samples.

### 11.5. Impact of different Grad-CAM Models

In our experiments, we use ResNe18 as the Grad-CAM model to extract activation maps. However, the choice of Grad-CAM model does not have a significant impact on the performance, as long as it has been trained on the target dataset. As shown in Table 12, differences between performances of different Grad-CAM models are within 0.5, demonstrating that our discriminative area enhancement module doesn't depend on the choice of Grad-CAM model.

### 11.6. Latency of Grad-CAM

We use torchcam [11] implementation of various Grad-CAM methods to extract activation maps. In Table 13, we show the latencies of extracting activation maps of various IPCs. Notably, these latencies are all below one second, demonstrating a high inference speed. In our experiments, the maximum number of extractions for one distillation is 200 (on IPC1). Thus, the total time used for activation map extraction is at most two minutes, which is neglectable compared with the latency of the full distillation (several hours). In conclusion, our use of Grad-CAM activation maps to provide guidance doesn't reduce the efficiency of the backbone.

## 12. Comp-DD Benchmark

### 12.1. Subset Details

The corresponding class labels for each subset are listed as follows:

- **Bird-Hard:** n01537544, n01592084, n01824575, n01558993, n01534433, n01843065, n01530575, n01560419, n01601694, n01532829

| Dataset | ImageNette | | | ImageWoof | | | ImageSquawk | | |
|---|---|---|---|---|---|---|---|---|---|
| IPC | 1 | 10 | 50 | 1 | 10 | 50 | 1 | 10 | 50 |
| Random | 12.6±1.5 | 44.8±1.3 | 60.4±1.4 | 11.4±1.3 | 20.2±1.2 | 28.2±0.9 | 13.2±1.1 | 29.6±1.5 | 52.8±0.4 |
| DM | 28.2±1.5 | 58.1±1.1 | 65.8±1.1 | 19.6±1.4 | 30.4±1.3 | 36.3±1.4 | 19.7±1.3 | 30.0±1.0 | 61.5±1.2 |
| MTT | 47.7±0.9 | 63.0±1.3 | 69.2±1.0 | 28.6±0.8 | 35.8±1.8 | 42.3±0.9 | 39.4±1.5 | 52.3±1.0 | 65.4±1.2 |
| SRe2L | 18.4±0.8 | 41.0±0.3 | 55.6±0.2 | 16.0±0.2 | 32.2±0.3 | 35.8±0.2 | 22.5±0.5 | 35.6±0.4 | 42.2±0.3 |
| RDED | 28.0±0.5 | 53.6±0.8 | 72.8±0.3 | 19.0±0.3 | 32.6±0.5 | **52.6±0.6** | 33.8±0.5 | 52.2±0.5 | 71.6±0.8 |
| EDF | **52.6±0.5** | **71.0±0.8** | **77.8±0.5** | **30.8±1.0** | **41.8±0.2** | 48.4±0.5 | **41.8±0.5** | **65.4±0.8** | **74.8±1.2** |

Table 11. Performances of SRe2L and RDED without using knowledge distillation during evaluation. EDF outperforms the other two methods in most of settings, and our advantage is more pronounced as IPC gets smaller.

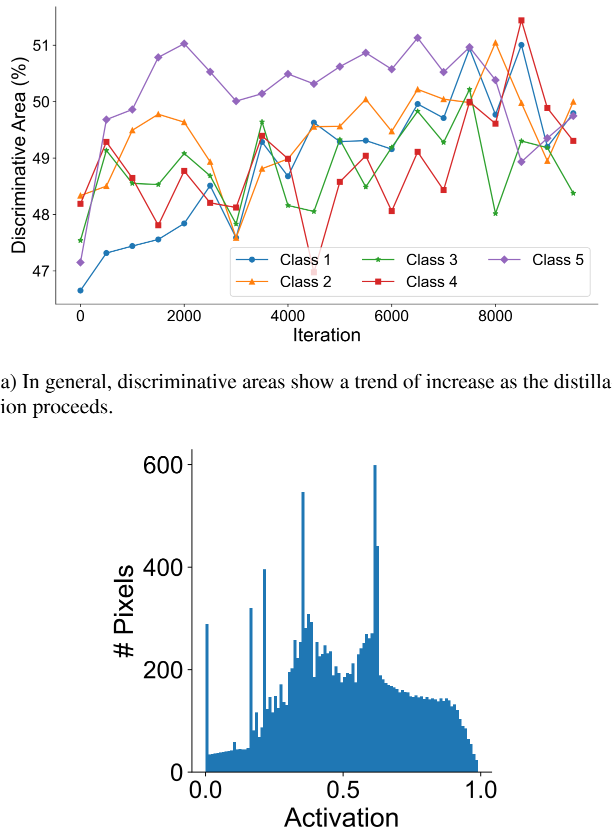
| IPC | Grad-CAM Model | | | |
|---|---|---|---|---|
| | ConvNetD5 | ResNet18 | ResNet50 | VGG11 |
| 1 | 52.3 | **52.6** | 52.5 | 52.5 |
| 10 | **71.2** | 71.0 | 70.8 | 70.7 |
| 50 | 77.4 | **77.8** | 77.6 | 77.6 |

Table 12. Results of using different Grad-CAM models on ImageNette. The choice of model only has minor influence on the performance.

| IPC | 1 | 10 | 50 | 200 | 300 |
|---|---|---|---|---|---|
| Latency (s) | 0.63 | 0.52 | 0.74 | 0.68 | 0.94 |

Table 13. Latency of extracting Grad-CAM activation maps using ResNet18. For each IPC in our experiments, the latency is less than one second.

- **Bird-Easy:** n02007558, n02027492, n01798484, n02033041, n02012849, n02025239, n01818515, n01820546, n02051845, n01608432
- **Dog-Hard:** n02107683, n02107574, n02109525, n02096585, n02085620, n02113712, n02086910, n02093647, n02086079, n02102040
- **Dog-Easy:** n02096294, n02093428, n02105412, n02089973, n02109047, n02109961, n02105056, n02092002, n02114367, n02110627
- **Car-Hard:** n04252077, n03776460, n04335435, n03670208, n03594945, n03445924, n03444034, n04467665, n03977966, n02704792
- **Car-Easy:** n03459775, n03208938, n03930630, n04285008, n03100240, n02814533, n03770679, n04065272, n03777568, n04037443
- **Snake-Hard:** n01693334, n01687978, n01685808, n01682714, n01688243, n01737021, n01751748, n01739381, n01728920, n01728572
- **Snake-Easy:** n01749939, n01735189, n01729977, n01734418, n01742172, n01744401, n01756291, n01755581, n01729322, n01740131
- **Insect-Hard:** n02165456, n02281787, n02280649, n02172182, n02281406, n02165105, n02264363, n02268853, n01770081, n02277742



(a) In general, discriminative areas show a trend of increase as the distillation proceeds.

(b) Most of the pixels have activation around 0.25 to 0.75.

Figure 7. **(a)** The trend of discriminative area change across various distillation iterations. **(b)** Distribution of the activation map of a random image from ImageNet-1K.

- **Insect-Easy:** n02279972, n02233338, n02219486, n02206856, n02174001, n02190166, n02167151, n02231487, n02168699, n02236044
- **Fish-Hard:** n01440764, n02536864, n02514041, n02641379, n01494475, n02643566, n01484850, n02640242, n01698640, n01873310
- **Fish-Easy:** n01496331, n01443537, n01498041,

n02655020, n02526121, n01491361, n02606052, n02607072, n02071294, n02066245

- **Round-Hard:** n04409515, n04254680, n03982430, n04548280, n02799071, n03445777, n03942813, n03134739, n04039381, n09229709
- **Round-Easy:** n02782093, n03379051, n07753275, n04328186, n02794156, n09835506, n02802426, n04540053, n04019541, n04118538
- **Music-Hard:** n02787622, n03495258, n02787622, n03452741, n02676566, n04141076, n02992211, n02672831, n03272010, n03372029
- **Music-Easy:** n03250847, n03854065, n03017168, n03394916, n03721384, n03110669, n04487394, n03838899, n04536866, n04515003

### 12.2. Complexity Metrics

We use the percentage of pixels whose Grad-CAM activation values exceed a predefined fixed threshold to evaluate the complexity of an image. In our settings, the fixed threshold is 0.5. The reasons for fixing the threshold at 0.5 are twofold. Firstly, when selecting subsets, images are static and won't be updated in any form (this is different from EDF's DAE module, which updates synthetic images). Thus, using a fixed threshold is sufficient for determining the high-activation areas.

Secondly, values of a Grad-CAM activation map range from 0 to 1, with higher values corresponding to higher activation. We present the distribution of the activation map of a random image from ImageNet-1K in Figure 13b, where the majority of pixels have activation values between 0.25 and 0.75. Subsequently, if the threshold is too small or too large, the complexity scores of all classes will be close (standard deviation is small), as shown in Figure 12 and 13. This results in no clear distinguishment between easy and hard subsets. Finally, we set 0.5 as the threshold, which is the middle point of the range. Complexity distribution under this threshold is shown in Figure 10.

Our complexity metrics are an early effort to define how complex an image is in the context of dataset distillation. We acknowledge potential biases or disadvantages and encourage future studies to continue the refinement of complex metrics.

### 12.3. Benchmark Hyper-parameters

For the trajectory training, experiment settings are the same as those used for ImageNet-1K and its subsets. For distillation, we provide hyper-parameters of EDF on the Complex DD Benchmark in Table 14. These hyper-parameters can serve as a reference for future works to extend to other subsets of the benchmark.

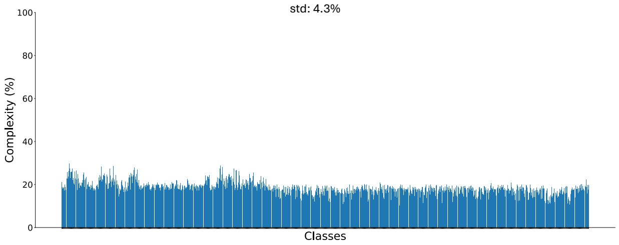


Figure 8. Complexity distribution of all classes from ImageNet-1K under threshold being 0.1. An excessively small threshold will cause the complexity of all classes to become low and difficult to distinguish.

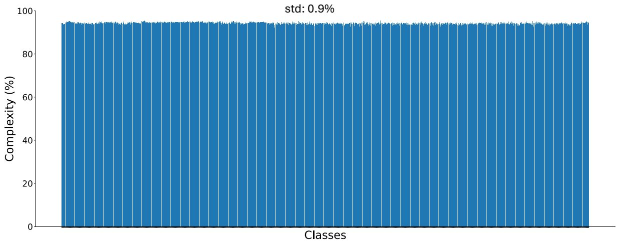


Figure 9. Complexity distribution of all classes from ImageNet-1K under threshold being 0.9. An excessively large threshold will cause the complexity of all classes to become high and difficult to distinguish.

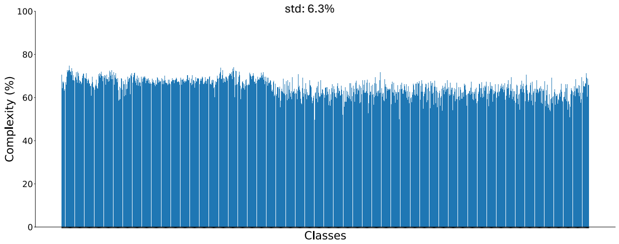


Figure 10. Complexity distribution of all classes from ImageNet-1K under threshold being 0.5. A moderate threshold makes the complexity differences between classes more distinct.

## 13. Visualization of Distilled Images on ImageNet-1K

In Figure 11 to 13, we present a visualization of distilled images of all ImageNet-1K subsets in Table 1.

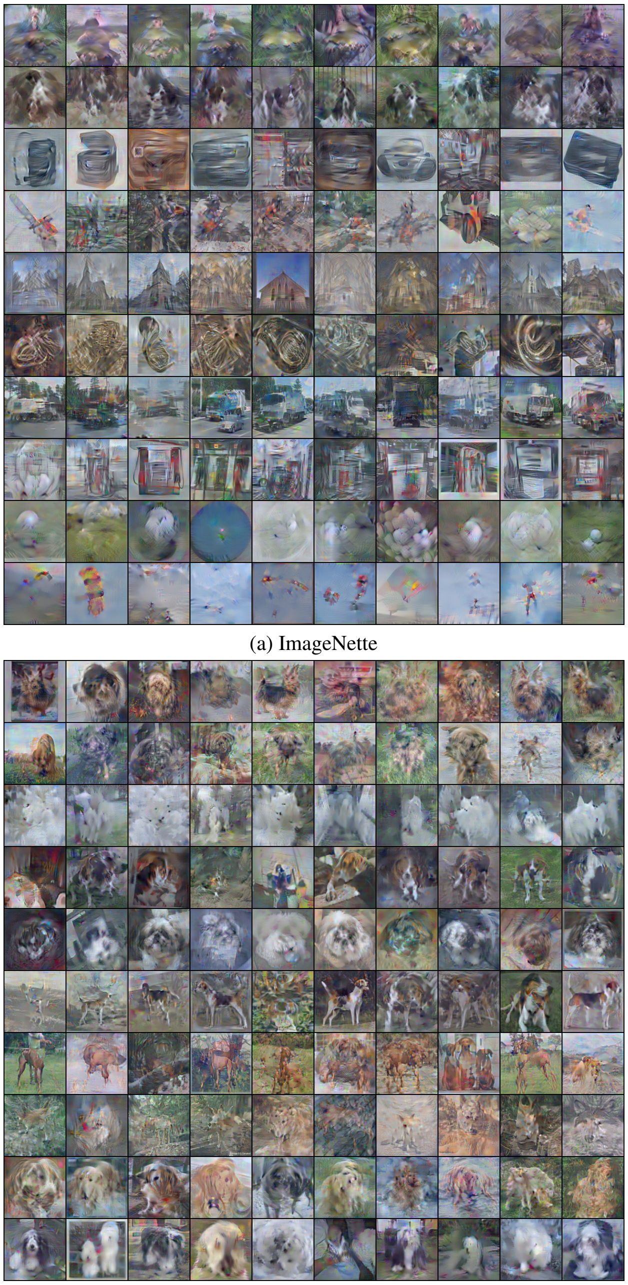
## 14. More Related Work

In Table 15, we present a comprehensive summary of previous dataset distillation methods, categorized by different approaches. There are four main categories of dataset distillation: gradient matching, trajectory matching, distribution matching, and generative model-based methods. Recently, some works [48, 64, 65] add knowledge distillation during

| Modules | | CPD | DAE | | TM | | | |
|---|---|---|---|---|---|---|---|---|
| Hyper-parameters | | $\alpha$ | $\beta$ | $K$ | $T$ | batch_syn | lr_pixel | syn_steps |
| CDD-Bird-Easy | 1 | 0 | 1 | 50 | 10000 | 1000 | 10000 | 40 |
| | 10 | 0.25 | 1 | 100 | | 400 | 1000 | |
| | 50 | 0.375 | 2 | 200 | | 200 | 100 | |
| CDD-Bird-Hard | 1 | 0 | 1 | 50 | 10000 | 1000 | 10000 | 40 |
| | 10 | 0.25 | 1 | 100 | | 400 | 1000 | |
| | 50 | 0.375 | 2 | 200 | | 200 | 100 | |
| CDD-Dog-Easy | 1 | 0 | 1 | 50 | 10000 | 1000 | 10000 | 40 |
| | 10 | 0.25 | 1 | 100 | | 400 | 1000 | |
| | 50 | 0.375 | 2 | 200 | | 200 | 100 | |
| CDD-Dog-Hard | 1 | 0 | 1 | 50 | 10000 | 1000 | 10000 | 40 |
| | 10 | 0.25 | 1 | 100 | | 400 | 1000 | |
| | 50 | 0.375 | 2 | 200 | | 200 | 100 | |
| CDD-Car-Easy | 1 | 0 | 1 | 50 | 10000 | 1000 | 10000 | 40 |
| | 10 | 0.25 | 1 | 100 | | 400 | 1000 | |
| | 50 | 0.375 | 2 | 200 | | 200 | 100 | |
| CDD-Car-Hard | 1 | 0 | 1 | 50 | 10000 | 1000 | 10000 | 40 |
| | 10 | 0.25 | 1 | 100 | | 400 | 1000 | |
| | 50 | 0.375 | 2 | 200 | | 200 | 100 | |

Table 14. Hyper-parameters of EDF on the Complex DD Benchmark.

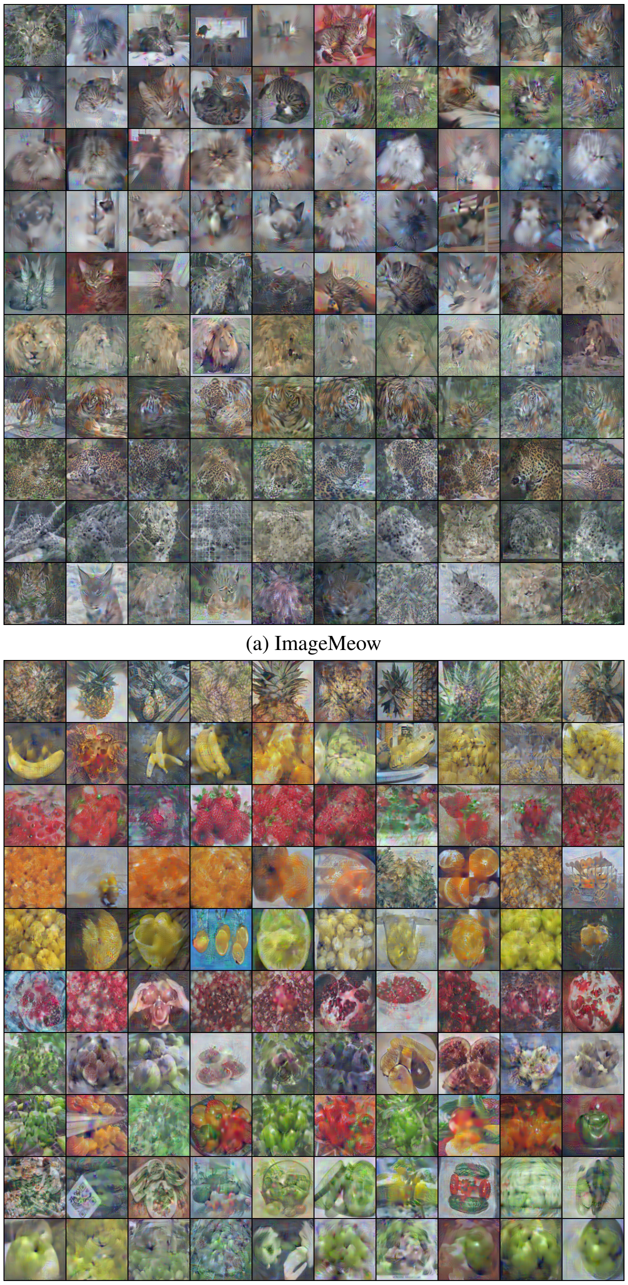
the evaluation stage of dataset distillation [4, 5, 7, 10, 17, 29, 33–36, 40, 50, 55–58, 60, 62, 63, 66].



(a) ImageNette

(b) ImageWoof

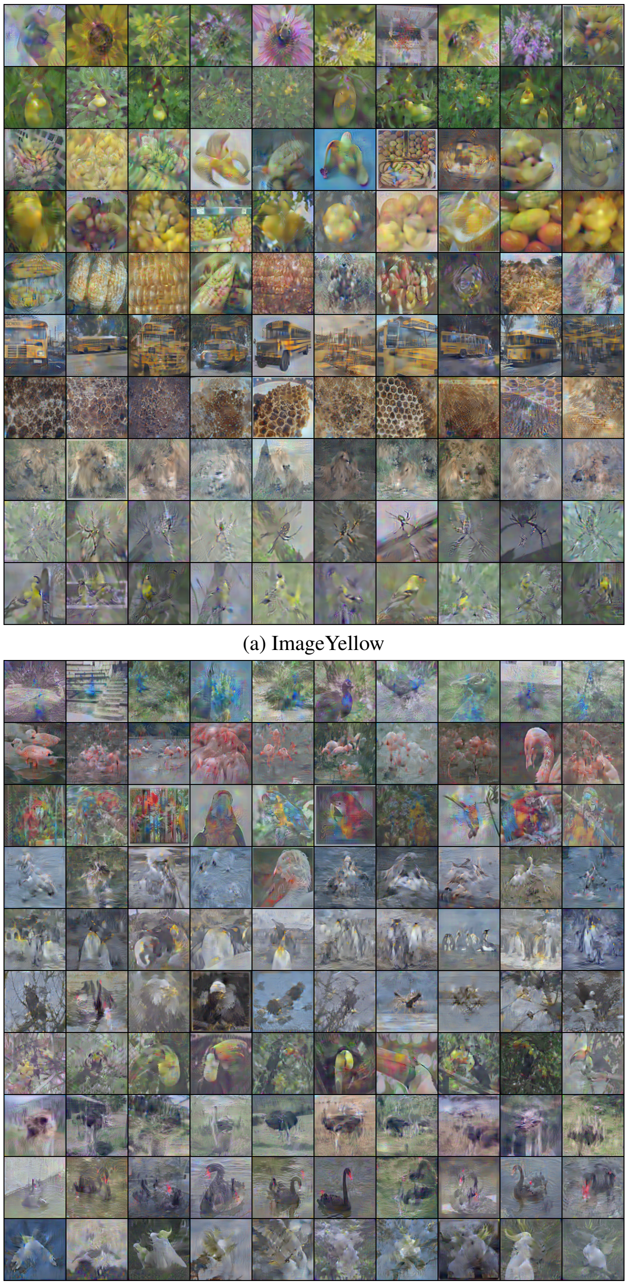
Figure 11



(a) ImageMeow

(b) ImageFruit

Figure 12



(a) ImageYellow

(b) ImageSquawk

Figure 13

| Category | Method |
|---|---|
| Kernel-based | KIP-FC [38] |
| | KIP-ConvNet [39] |
| | FRePo [78] |
| | RFAD [31] |
| | RCIG [32] |
| Gradient-matching | DC [71] |
| | DSA [69] |
| | DCC [22] |
| | LCMat [44] |
| Trajectory-matching | MTT [1] |
| | Tesla [6] |
| | FTD [8] |
| | SeqMatch [9] |
| | DATM [13] |
| | ATT [27] |
| | NSD [61] |
| | PAD [25] |
| | SelMatch [24] |
| Distribution-matching | DM [68] |
| | CAFE [51] |
| | IDM [72] |
| | DREAM [30] |
| | M3D [67] |
| | NCFD [53] |
| Generative model | DiM [52] |
| | GLaD [2] |
| | H-GLaD [77] |
| | LD3M [37] |
| | IT-GAN [70] |
| | D4M [46] |
| | Minimax Diffusion [12] |
| + Knowledge distillation for evaluation | SRe2L [64] |
| | RDED [48] |
| | HeLIO [65] |
| Others | MIM4DD [43] |
| | DQAS [75] |
| | LDD [76] |

Table 15. Summary of previous works on dataset distillation