FocusDC: Real-World Scene Infusion for Robust Dataset Condensation

Youbing Hu[1] Yun Cheng[2] Olga Saukh[3] Firat Ozdemir[2]
Anqi Lu[1] Zhiqiang Cao[1] Min Zhang[4] Zhijun Li[1]
[1]Faculty of Computing, Harbin Institute of Technology
[2]Swiss Data Science Center, Zurich, Switzerland, [3]Graz University of Technology, Austria
[4]Intelligent Computing Research Center, Harbin Institute of Technology (Shenzhen)
{youbing, zhiqiang_cao, luanqi}@stu.hit.edu.cn, {yun.cheng, firat.ozdemir}@sdsc.ethz.ch,
saukh@tugraz.at, {zhangmin2021, lizhijun_os}@hit.edu.cn

Dataset condensation has emerged as a strategy to compress real-world datasets for efficient training. However, it struggles with large-scale and high-resolution datasets, limiting its practicality. This paper introduces a novel resolution-independent dataset distillation method **Focus**ed **D**ataset **C**ondensation (**FocusDC**), which achieves diversity and realism in distilled data by identifying key information patches, thereby ensuring the generalization capability of the distilled dataset across different network architectures. Specifically, FocusDC leverages a pre-trained Vision Transformer (ViT) to extract key image patches, which are then synthesized into a single distilled image. These distilled images, which capture multiple targets, are suitable not only for classification tasks but also for dense tasks such as object detection. To further improve the generalization of the distilled dataset, each synthesized image is augmented with a downsampled view of the original image. Experimental results on the ImageNet-1K dataset demonstrate that, with 100 images per class (IPC), ResNet50 and MobileNet-v2 achieve validation accuracies of 71.0% and 62.6%, respectively, outperforming state-of-the-art methods by 2.8% and 4.7%. Notably, FocusDC is the first method to use distilled datasets for object detection tasks. On the COCO2017 dataset, with an IPC of 50, YOLOv11n and YOLOv11s achieve 24.4% and 32.1% mAP, respectively, further validating the effectiveness of our approach.

1. Introduction

Contemporary deep learning has achieved remarkable success largely due to the exponential growth in model sizes [1–4] and data scales [5–7]. This growth has led to the development of advanced neural networks that achieve groundbreaking performance in tasks like image classification [3], object detection [8], and natural language processing [9]. However, this progress is not without its challenges. The rapid expansion of model complexities and data volumes has led to significantly increased computational costs and time expenses, in particular when training large neural networks on high-resolution and large-scale datasets [10–12]. These challenges significantly hinder the deployment of deep learning models, especially in resource-limited environments [13].

Dataset condensation/distillation [14] has emerged as a promising strategy to address these challenges. The core idea is to compress large, real-world datasets into smaller, more manageable representations that retain essential information while reducing the computational burden of ingesting them. Various methods have been proposed, including coreset selection-based distillation [15–19], which select representative samples from the original dataset; bi-level optimization-based distillation [20–24], which treats dataset distillation as a meta-learning problem involving two nested optimization loops—where the outer loop optimizes the meta-dataset and the inner loop trains a model with the distilled data; and distillation with prior regularization [25–27], which leverages prior knowledge at the feature level to guide the generation of the condensed dataset.

Although traditional solutions have made significant progress in handling small-scale and low-resolution datasets (such as Tiny-ImageNet [28], downscaled ImageNet [29], or subsets of Ima-

Third Conference on Parsimony and Learning (CPAL 2026).

geNet [30]), their high computational cost limits their practical application when scaled to high-resolution and large-scale datasets. To address this issue, SRe^{2}L [31] proposed a decoupled approach for model updates and datasets, which was the first to extend dataset distillation techniques to the scale of ImageNet. Subsequently, several methods [32–35] have been proposed to improve the efficiency of SRe^{2}L and significantly enhance accuracy. For example, SCDD [32] replaces the batch-level statistics used in SRe^{2}L with statistics calculated over the entire distillation dataset. RDED [33] randomly crops a region from the original high-resolution image, selects multiple images with the highest authenticity scores, and merges them into a distilled image. While these methods effectively synthesize high-resolution images, they rely on specific network architectures during the distillation process, limiting the generalization ability of the distilled dataset. Furthermore, the datasets distilled by these methods typically only apply to classification tasks and cannot be directly applied to dense tasks, such as object detection.

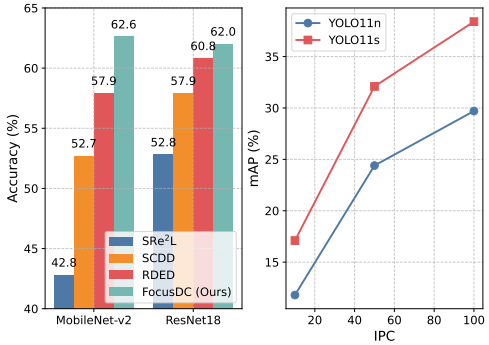


Figure 1: FocusDC performance on classification and detection tasks.

In this paper, we propose a novel dataset distillation method called **Focus**ed **D**ataset **C**ondensation (**FocusDC**), which aims to improve the efficiency and realism of dataset distillation by focusing on key information patches within the data. FocusDC consists of two stages: (i) **information extraction** and (ii) **image reconstruction**. In the information extraction stage, we leverage a pre-trained Vision Transformer (ViT) [3] to guide the selection of key image patches. By using ViT, we can accurately extract key image patches corresponding to foreground objects, thereby enhancing the realism of the distilled dataset and ensuring the relevance of the extracted information. Since these distilled images contain target regions, they are well-suited for downstream dense tasks such as object detection. As shown in Fig. 1, FocusDC demonstrates superior performance at different IPC levels on the COCO validation dataset when using the YOLOv11 model. This is the first work to extend dataset distillation methods to object detection tasks.

In the reconstruction stage, we combine downsampled versions of representative real images with the extracted key image patches to generate distilled images. This process not only preserves the diversity of the dataset but also ensures its realism, providing high-quality training data that enhances the generalization ability of the model. Table 1 highlights the advantages of FocusDC in improving model generalization performance. Finally, an optional dynamic fine-tuning on a small subset of the original dataset can further boost performance and is investigated in Appendix 6.2.1.

Table 1: Generalization performance of ResNet-18 with IPC=10.

Model	Method	Flower102	Food101	CIFAR100
ResNet18	Random	22.4%	57.8%	54.5%
	RDED	67.8%	74.2%	69.3%
	FocusDC	**71.1%**	**77.6%**	**71.3%**

Our contributions can be summarized as follows:

(1) We are the first to integrate ViTs into the image distillation process. By selectively emphasizing critical regions and foreground objects, ViT ensures that the distilled dataset retains crucial contents of the data distribution for model training.

(2) Our method not only preserves the realism and diversity of the images but also enables effective application to downstream dense tasks, such as object detection. By leveraging Attention-guided distillation, we can clearly identify the image regions most critical for model learning. To the best of our knowledge, we are the first work to extend dataset distillation to object detection tasks.

(3) We provide a rigorous evaluation of our approach including multiple ablation studies and show improved model generalization capabilities across different network architectures. Com-

pared to SOTA methods on classification tasks, FocusDC improves the accuracy of ResNet50 and MobileNetV2 at IPC level 50 by 2.8% and 4.7%, respectively. On object detection tasks, FocusDC achieves 24.4% mAP with YOLOv11n, and 32.1% mAP with YOLOv11s at an IPC of 50 on the COCO validation set.

2. Related Work

Data condensation [14] aims to reduce the computational costs of training deep learning models by condensing large datasets into smaller, information-rich subsets. Most previous dataset distillation methods [14, 21, 23, 36–41] focus on small-scale and low-resolution datasets [28–30] and can be classified into several categories: Bi-level optimization methods treat dataset distillation as a meta-learning problem, where an outer loop optimizes the synthetic dataset while an inner loop focuses on model training using distilled data, methods include FRePo [36], DD [14], RFAD [37], KIP [37], and LinBa [42]. Trajectory-matching methods align model training trajectories on the original and distilled datasets over multiple iterations, methods include MTT [38], TESLA [27], and DATM [23]. Distribution-matching methods match the distribution of the distilled dataset with that of the original in a single optimization step, with examples like KFS [40], DM [21], CAFE [39], HaBa [43], and IT-GAN [44]. Gradient-matching methods align gradients of the network trained on original and synthesized data, with examples including DSA [45], IDC [30], DC [41], and DCC [40].

Building on these foundations, recent approaches have extended dataset distillation to large-scale, high-resolution datasets. For example, SRe^{2}L [31] decouples model updates and dataset synthesis through "squeeze", "restore", and "relabel" stages, pioneering the expansion of dataset distillation to ImageNet-scale resolutions. SCDD [32] further improves on SRe^{2}L by replacing batch-level statistics with global dataset statistics, achieving notable performance gains. D3S [35] reframes dataset distillation as a domain shift problem, introducing a scalable algorithm, while RDED [33] generates distilled images by randomly cropping and selecting high-realism image regions. Moreover, some dataset distillation methods [46, 47] leverage diffusion models for distillation.

Although previous methods excel with high-resolution images, they compress the original dataset into a specific architecture [31–33], limiting the generalization of the distilled dataset. In contrast, FocusDC synthesizes datasets using the well-established Attention mechanism, which improves generalization, as shown in Table 1 and Table 4 across different ViT models. Furthermore, by synthesizing images focused on target locations, FocusDC extends its use to dense tasks like object detection, marking the first application of dataset distillation in this domain.

3. METHOD

3.1. Preliminaries

Dataset condensation. Dataset condensation [14] aims to compress information from a large-scale original dataset to a new compact dataset while striving to preserve the utmost degree of the original data informational essence. The resulting compressed dataset denoted as $\mathcal{S}$, should enable a model trained on it to perform comparably to a model trained on the original, full dataset $\mathcal{T}$. Considering a large labeled dataset $\mathcal{S} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_{|\mathcal{S}|}, y_{|\mathcal{S}|})\}$, where $|\mathcal{S}|$ denotes the total number of samples, and each $\boldsymbol{x}_i$ is an image with its corresponding label y_i. The aim is to create a condensed dataset $\mathcal{T} = \{(\tilde{\boldsymbol{x}}_1, \tilde{y}_1), \ldots, (\tilde{\boldsymbol{x}}_{|\mathcal{T}|}, \tilde{y}_{|\mathcal{T}|})\}$ that retains the key features of $\mathcal{S}$, with $|\mathcal{T}| \ll |\mathcal{S}|$, ensuring that this reduction in size does not compromise the dataset integrity. The learning objective is to minimize the performance disparity between the model trained on $\mathcal{T}$ and the one trained on $\mathcal{S}$, as expressed by the following constraint:

$$\sup\left\{\left|\ell(\phi_{\boldsymbol{\theta}_\mathcal{S}}(\boldsymbol{x}), y) - \ell(\phi_{\boldsymbol{\theta}_\mathcal{T}}(\boldsymbol{x}), y)\right|\right\}_{(\boldsymbol{x},y)\sim\mathcal{S}} \leq \epsilon, \tag{1}$$

where ϵ represents the allowable performance disparity between models trained on $\mathcal{T}$ versus those trained on $\mathcal{S}$. Here, $\boldsymbol{\theta}_\mathcal{S}$ parameterizes the neural network ϕ, optimized on $\mathcal{S}$ as follows:

$$\boldsymbol{\theta}_\mathcal{S} = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{S}}[\ell(\phi_{\boldsymbol{\theta}}(\boldsymbol{x}), y)]. \tag{2}$$

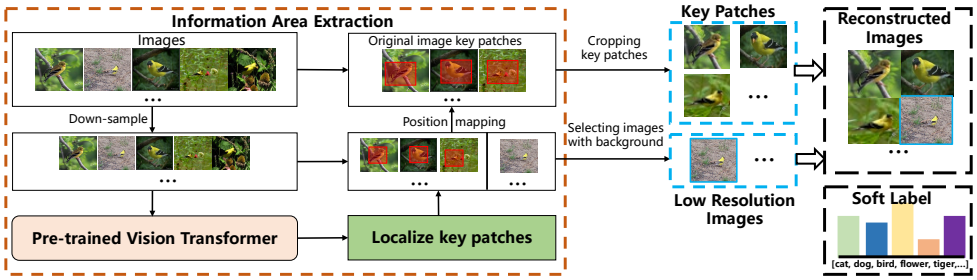


Figure 2: Overview of the FocusDC framework.

In this formulation, ℓ is the loss function, and $\boldsymbol{\theta}_{\mathcal{T}}$ is defined in a similar manner for the condensed dataset. This framework ensures that $\mathcal{T}$ maintains the essential characteristics of $\mathcal{S}$, allowing effective training on a smaller scale.

Vision Transformer. Vision Transformer (ViT) [3] adapts the Transformer architecture [9], originally developed for natural language processing, to the domain of image analysis. They treat image patches as sequential inputs, allowing the model to capture global dependencies across the image. Each image is segmented into patches, which are embedded and supplemented with positional encodings to maintain spatial information, denoted as: $\boldsymbol{x} = [\boldsymbol{x}_{\text{cls}}; \boldsymbol{E}(\boldsymbol{p}_1); \boldsymbol{E}(\boldsymbol{p}_2); \ldots; \boldsymbol{E}(\boldsymbol{p}_K)] + \boldsymbol{E}_{\text{pos}}$, where $\boldsymbol{E}$ is the embedding function, $\boldsymbol{p}_i$ are the patches, $\boldsymbol{x}_{\text{cls}}$ is the class token, and $\boldsymbol{E}_{\text{pos}}$ represents the positional encodings. The self-attention mechanism then calculates attention scores to determine the relevance of each patch relative to others:

$$\begin{aligned} \mathcal{A}(\mathbf{Q}, \mathbf{K}) = \text{Softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}) = [\mathbf{A}^1; \mathbf{A}^2; \ldots; \mathbf{A}^K], \\ \text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathcal{A}(\mathbf{Q}, \mathbf{K})\mathbf{V}, \end{aligned} \tag{3}$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are the query, key, and value matrices from $\boldsymbol{x}$, d is the embedding dimension of $\mathbf{K}$, and K is the number of patches. The average attention score $\boldsymbol{s}$ for an image reflects the outcome of a single-head self-attention mechanism. In multi-head self-attention, scores from all attention heads are averaged to yield the final image attention score. The class token $\boldsymbol{x}_{\text{cls}}$ is processed by a classifier $\mathcal{F}$ to derive the category prediction distribution $\mathbf{p}^c$:

$$\boldsymbol{s} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{A}^k = [s^1, s^2, \ldots, s^K], \mathbf{p}^c = \mathcal{F}(\boldsymbol{x}_{\text{cls}}) = [p_1^c, p_2^c, \ldots, p_C^c], \tag{4}$$

where C indicates the number of categories.

3.2. Focused Dataset Distillation with Attention

This section introduces FocusDC, a dataset distillation method that reconstructs compiled images by focusing on the target and representative background information of real images. Fig. 2 provides an overview. Further details are provided below.

Attention-guided Information Extraction: We utilize an attention mechanism to identify and extract regions with the highest attention scores from multiple images, thereby compiling images with enhanced detail. These regions are then combined to form a detailed composite image set, as illustrated in Fig. 2. The process initiates by performing the following steps on each image $\boldsymbol{x}_i \in \mathbb{R}^{H\times W\times Ch}$ within each category-specific subset $\mathcal{S}_c$ of the dataset $\mathcal{S}$: each $\boldsymbol{x}_i$ is downsampled to $\boldsymbol{x}_i'$ and segmented into non-overlapping patches of size $P \times P$. This downsampling produces $K = \frac{H}{P} \times \frac{W}{P}$ patches per image, which are subsequently reorganized into the structured form $\mathbb{R}^{\frac{H}{P}\times\frac{W}{P}\times P^2Ch}$, with each row and column representing a token. These tokens are embedded and fed into a pre-trained ViT model $\phi_{\boldsymbol{\theta}_{\mathcal{S}}}^{ViT}$, yielding predictive distributions $\mathbf{p}_i^c$ and attention scores $\boldsymbol{s}_i \in \mathbb{R}^K$. Likewise, we reorganize each attention score $\boldsymbol{s}_i$ into the format $\frac{H}{P} \times \frac{W}{P}$. To determine the size of the highest attention score region for each image $\boldsymbol{x}_i'$, we introduce an adjustable hyperparameter α, which specifies the number of patches $\lfloor \alpha\frac{H}{P} \times \alpha\frac{W}{P} \rfloor$. We then introduce a realism score

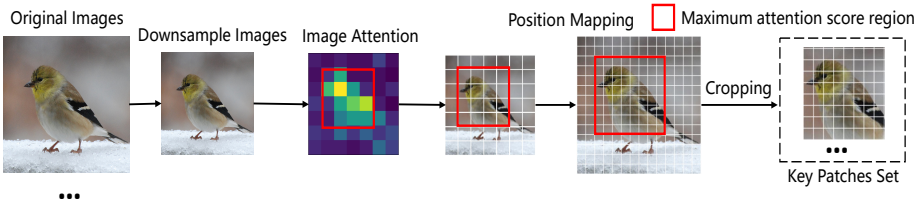


Figure 3: The FocusDC process of selecting key image patches.

s_i^{real} to identify the key patch for each image. Specifically, our realism score combines the prediction distribution $\mathbf{p}_i^c$ of each image with the highest attention region score s_i^{area}, defined as follows:

$$s_i^{\text{real}} = \max(\text{softmax}(\mathbf{p}_i^c)) + \eta s_i^{\text{area}}, \tag{5}$$

where η is a balancing factor. Intuitively, s_i^{real} indicates the need to select a representative image with a focus on the target region within it. This implies that our selection process should prioritize images that represent the overall scene accurately and emphasize the specific area of interest, ensuring that the target region is well-captured and highlighted in the chosen image.

After calculating the realism score s_i^{real}, we associate each score with its corresponding image $\mathcal{S}_c$ and sort the scores in descending order. Based on these scores, we select the top-M images from the sorted $\mathcal{S}_c$ and extract the regions with the highest attention scores. The center indices of these high attention regions are determined using the following formula:

$$(i, j) = \arg\max_{i,j} \sum_{p,q} \boldsymbol{s}^{i+p-\lfloor \frac{h}{2} \rfloor, j+q-\lfloor \frac{w}{2} \rfloor}, \tag{6}$$

where $h = \lfloor \alpha \frac{H}{P} \rfloor$, $w = \lfloor \alpha \frac{W}{P} \rfloor$, $p \in \{0, 1, \ldots, h-1\}$, and $q \in \{0, 1, \ldots, w-1\}$. Utilizing the positional mapping function ρ, we translate these indices to the dimensions of the original image $\boldsymbol{x}_i$, marking the key information region $\boldsymbol{x}_i^\star$ in the high-resolution image as:

$$\begin{aligned} \boldsymbol{x}_i^\star = \text{area}(\rho((i - \lfloor \alpha \frac{H}{2P} \rfloor, j + \lfloor \alpha \frac{W}{2P} \rfloor), \\ (i + \lfloor \alpha \frac{H}{2P} \rfloor, j - \lfloor \alpha \frac{W}{2P} \rfloor))). \end{aligned} \tag{7}$$

Finally, we compile the identified key patches into a set $\tilde{\mathcal{T}}_c = \{\boldsymbol{x}_i^\star\}_{i=1}^M$, where each sample $\boldsymbol{x}_i^\star$ is a crop of the high-resolution image containing fine details, thereby preserving maximum informational content for use in the compiled composites. To further enhance the diversity of the synthesized images, we randomly select N low-resolution sampled images from $\mathcal{S}_c$ that were not chosen as key information patches. These images are weighted based on their prediction confidence scores and added to $\mathcal{T}_c' = \{\boldsymbol{x}_i'\}_{i=1}^N$ as a background information set. Fig. 3 illustrates the process of selecting the set of key patches.

Information Reconstruction: The size of key patches is typically smaller than the target distilled images. Directly using these key patches as distilled images can result in sparse information distribution in the pixel space, thereby reducing the effectiveness of the learning model [31, 48, 49]. As shown in Table 8, using distilled image sets composed solely of key patches leads to a decreased model performance. Therefore, we combine the set of images containing key information patches $\tilde{\mathcal{T}}_c$ with the set of low-resolution images $\mathcal{T}_c'$ to supplement the class category c information with the typical context in which they appear. Specifically, we randomly select m patches from $\tilde{\mathcal{T}}_c$ and n low-resolution images from $\mathcal{T}_c'$ each time. The selected images are then concatenated to compile the final composite image $\tilde{\boldsymbol{x}}_j$:

$$\tilde{\boldsymbol{x}}_j = \text{concat}((\{\boldsymbol{x}_j^\star\}_{j=1}^m \subset \tilde{\mathcal{T}}_c), (\{\boldsymbol{x}_j'\}_{j=1}^n \subset \mathcal{T}_c')). \tag{8}$$

By default, we set the combined total of patches and images to $m + n = 4$ (see Fig. 4), where $m = 3$ represents the selection of three patches from the key information patch collection $\tilde{\mathcal{T}}_c$, and $n = 1$ corresponds to selecting one low-resolution image from the background information collection $\mathcal{T}_c'$ (see Table 8). Following the RDED [33] and SRe^{2}L [31], we apply a soft label approach [49] to the

Table 2: Comparison with SOTA baseline dataset distillation methods on the ImageNet-1K dataset.

Method	IPC							
	1	10	50	100	1	10	50	100
	ResNet-18 (69.8 ±0.1)				ResNet-50 (76.2±0.1)			
SRe^{2}L [31]	0.1±0.1	21.3±0.6	46.8±0.2	52.8±0.3	0.3±0.1	28.4±0.1	55.6±0.3	61.0±0.4
SCDD [32]	-	32.1±0.2	53.1±0.1	57.9±0.1	-	38.9±0.1	60.9±0.2	65.8±0.1
GVBSM [50]	-	31.4±0.5	51.8±0.4	55.7±0.4	-	35.4±0.8	58.7±0.3	62.2±0.3
RDED [33]	6.6±0.2	42.0±0.1	56.5±0.1	60.8±0.4	5.7±0.1	42.3±0.3	64.8±0.6	68.2±0.2
D3S [35]	-	39.1±0.3	60.2±0.1	**63.0±0.2**	-	41.9±0.7	65.8±0.1	68.2±0.1
FocusDC (Ours)	**8.8±0.2**	**45.3±0.1**	**61.7±0.1**	62.0±0.2	**6.8±0.1**	**46.3±0.2**	**69.1±0.3**	**71.0±0.1**
	MobileNet-V2 (71.8±0.1)				EfficientNet-B0 (76.3±0.1)			
SRe^{2}L [31]	0.3±0.1	10.2±2.6	31.8±0.3	42.8±0.6	0.4±0.2	11.4±2.5	34.8±0.4	49.6±0.5
RDED [33]	4.9±0.6	33.8±0.6	54.2±0.2	57.9±0.6	3.4±0.2	33.3±0.9	57.7±0.1	63.7±0.3
FocusDC (Ours)	**5.1±0.1**	**34.6±0.1**	**58.7±0.3**	**62.6±0.1**	**4.8±0.2**	**40.1±0.2**	**60.7±0.1**	**66.6±0.3**

compiled images. This method generates region-level soft labels $\tilde{y}_j^k = \ell(\phi_{\boldsymbol{\theta}_\mathcal{T}}(\tilde{\boldsymbol{x}}_j^k))$, where $\tilde{\boldsymbol{x}}_j^k$ is the k-th region in the distilled image, and $\tilde{y}_j^k$ is its corresponding soft label.

By iterating over each category c in $\mathcal{S}$, performing the information extraction and image reconstruction processes, and adding the generated images and labels $\{\tilde{\boldsymbol{x}}_j, y_j\}$ to the distilled dataset $\mathcal{T}$, we ultimately obtain the complete distilled dataset $\mathcal{T}$.

3.3. Model Training on Distilled Datasets

After assembling the distillation dataset $\mathcal{T}'$, we initiate training of a student model $\phi_{\boldsymbol{\theta}_{stu}}$ from random initialization using this dataset, in line with strategies proposed by Yin et al. [31] and Sun et al. [33]. For classification tasks, the training employs a cross-entropy loss function defined as:

$$\mathcal{L} = -\sum_j \sum_k \tilde{y}_j^k \log \phi_{\boldsymbol{\theta}_{stu}}(\tilde{\boldsymbol{x}}_j^k). \tag{9}$$

To optimize training efficiency for the detection task, we input the distilled images into YOLOv11x [51] to compute the classification and bounding box losses and supervise model updates using Kullback–Leibler divergence loss. To accelerate training, we employ YOLOv11x to generate GT boxes for synthesized images and train the model using the standard YOLOv11 protocol.

Table 3: Accuracy comparison (%) of SOTA baseline dataset distillation methods using ResNet101 (77.4±0.2) on ImageNet.

Method	IPC			
	1	10	50	100
SRe^{2}L [31]	0.6±0.1	30.9±0.1	60.8±0.5	62.8±0.2
SCDD [32]	-	39.6±0.4	61.0±0.3	65.6±0.2
GVBSM [50]	-	38.2±0.4	61.0±0.4	63.7±0.2
RDED [33]	5.9±0.4	42.1±1.0	61.2±0.4	69.5±0.5
D3S [35]	-	42.1±3.8	65.3±0.5	68.9±0.1
FocusDC (Ours)	**8.5±0.2**	**43.1±0.2**	**69.9±0.2**	**72.9±0.1**

In Appendix 6.2.1, we outline how a model, initially trained on a distilled dataset, undergoes Dynamic Fine-Tuning (DFT) on the data obtained by dynamically sampling the original dataset. This method leads to further performance enhancements across all architectures.

4. EXPERIMENT

4.1. Experiment Setting

Datasets and Implementation Details. We conducted rigorous and extensive validation of FocusDC on the large-scale ImageNet-1K dataset [5] to comprehensively evaluate its performance. The ImageNet-1K dataset consists of approximately 1.2 million training images with a resolution of

224×224 pixels, spanning 1000 categories. For key patch extraction, we utilized the Deit-S model [11], pre-trained by Hu et al. [52]. We maintain a constant side ratio α of 0.8 and η of 30. We set the value of N equal to IPC and M equal to 3×IPC, effectively limiting the size of the distillation dataset to the total number of pixels in the IPC image. We train target models including ResNet-$\{18, 50, 101\}$ [1], MobileNet-v2 [53], and EfficientNet-b0 [54] to validate the distilled datasets. All models are trained on the distilled dataset for 300 epochs with 224×224 image resolution. Our experiments were conducted using an NVIDIA 4090 GPU. Additional experimental details and Tiny-ImageNet [28] results are provided in Appendix 6.1 and 6.2.1 Table 13 , respectively.

Evaluation and Baselines. We compare our approach with several SOTA methods for distilling large-scale, high-resolution datasets, including SRe^{2}L [31], SCDD [32], GVBSM [50], D3S [35] and RDED [33]. In our evaluation process, we generate a unique distillation dataset for each IPC level (1, 10, 50, 100) for FocusDC and reuse it across multiple network architectures.

Table 4: Impact of different ViT models on FocusDC accuracy.

Distillation Architecture	IPC			
	1	10	50	100
Deit-S	8.8±0.2	45.3±0.1	61.7±0.1	62.0±0.2
LV-ViT-S	9.4±0.3	45.8±0.2	62.3±0.2	62.8±0.1

4.2. Overall Performance

ImageNet-1K Classification. Tables 2 and 3 present the experimental results of FocusDC on the ImageNet-1K dataset, showing its significant advantages across various architectures (e.g., ResNet-18, ResNet-50, ResNet-101, MobileNet-V2, EfficientNet-B0) and IPC settings. FocusDC consistently outperforms other methods, especially for low IPCs (1, 10, and 50), achieving higher accuracy, which is crucial for scenarios with limited samples or resource constraints. For instance, on ResNet-18, FocusDC achieves accuracies of 8.8% and 45.3% at IPCs of 1 and 10, respectively, significantly surpassing RDED and D3S. Even for higher IPCs (e.g., IPC = 100), FocusDC maintains strong performance, often achieving or nearing the best results on ResNet-50 and EfficientNet-B0. This demonstrates FocusDC's ability to excel under minimal and small-sample data conditions, adapting effectively across different models and IPC configurations.

Table 5: **Comparison of different Coreset selection-based dataset distillation baselines.** All methods use ResNet-18 as the validation model and IPC=10.

Dataset	Random	Herding	K-Means	FocusDC (Ours)
Tiny-ImageNet	7.5±0.1	9.0±0.3	8.9±0.2	**51.5±0.1**
ImageNet-1K	4.4±0.1	5.8±0.1	5.5±0.1	**45.3±0.1**

Additionally, we compare our method with diffusion-based image generation models [46, 55] in Table 6. Table 5 compares FocusDC with Coreset-based selection methods [56, 57] on ImageNet-1K, showing consistent superiority of FocusDC. Table 13 in Appendix 6.2.1 shows FocusDC's strong performance on Tiny-ImageNet, even at low IPCs, aligning with results on ImageNet-1K.

Table 6: Comparison of classification accuracy (%) when training with diffusion-based network generated datasets and FocusDC. ResNet-18 was used as a validation model.

IPC	DiT [55]	MinmaxDiffusion [46]	FocusDC (Ours)
10	39.6±0.4	44.3±0.5	**45.3±0.1**
50	52.9±0.6	58.6±0.3	**61.7±0.1**

COCO Object Detection: In the object detection task, we follow the official YOLOv11 [51] settings and use YOLOv11x as the teacher to soft-supervise YOLOv11n and YOLOv11s trained from scratch for 100 epochs. Fig. 1 reports the mAP of FocusDC-distilled datasets on the COCO validation set under different IPC settings. As IPC increases, performance consistently improves—for instance, YOLOv11s reaches 32.1% mAP at IPC 50. FocusDC is effective for detection because its distilled images contain multiple target-rich patches, each providing diverse object cues for the detector.

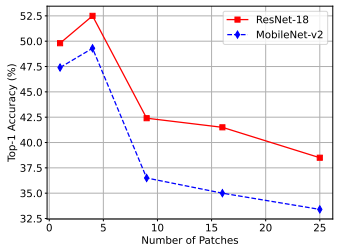


Figure 4: Impact of the number of patches in each distilled image.

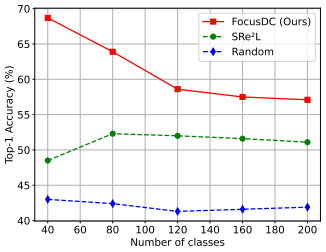


Figure 5: 5-step class-incremental learning on Tiny-ImageNet.

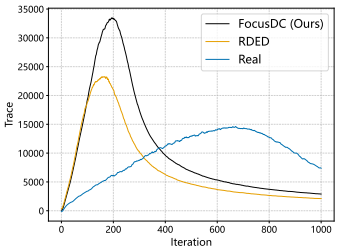


Figure 6: Curvature of loss landscapes with real-vs-distilled data.

4.3. Performance Analysis

Cross-Architecture Generalization: Table 4 evaluates the impact of different ViT models on FocusDC's performance on ImageNet-1K, using ResNet-18 for validation. The results demonstrate that our method maintains consistent performance across ViT architectures, corroborating the idea that the attention-based key patch selection in FocusDC is similarly effective for also different transformer architectures. Table 1 presents results from fine-tuning pre-trained models for 10 epochs on CIFAR-100 [58], Flowers-102 [59], and Food-101 [60], with datasets distilled using different methods. Our method improves generalization and downstream performance.

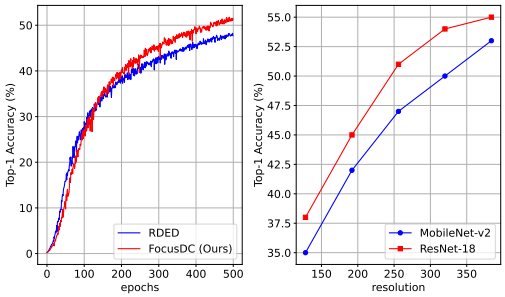


Figure 7: Model accuracy with varying epoch and resolution.

Training Epoch and Adaptive Resolution Synthesis: There is a lack of a unified benchmark for comparing methods, such as the number of training steps or used image resolution. This makes it hard to compare all SOTA baselines in the form they were initially presented to the community; e.g., those in [46] and [30], conducted across 1000 epochs. Nonetheless, we show the impact of training time on FocusDC and RDED in Fig. 7 (left). Accuracy improves with more training epochs, consistent with the findings of D3S [35]. Fig. 7 (right) shows the FocusDC synthesis accuracy at different resolutions, with accuracy improving as resolution increases, demonstrating the effectiveness of adaptive resolution control in image synthesis. Notably, during training, all images are resized to a fixed resolution of 224×224.

Comparison of Learning Efficiency: Fig. 6 highlights the effectiveness of our approach, with FocusDC outperforming RDED in learning efficiency. Higher Hessian matrix [61] trace values show that FocusDC adapts faster to new data and captures fundamental features more effectively, enhancing generalization.

Downstream Task: Continuous Learning Fig. 5 shows FocusDC's continual learning performance on TinyImageNet using ResNet18 with 5-step validation. FocusDC consistently outperforms the random baseline and matches or slightly exceeds SRe^{2}L [31] as classes increase from 40 to 200, showcasing its ability to maintain high accuracy and adapt robustly to new classes.

4.4. Ablation study

Effectiveness of Each Technique in FocusDC. To validate the effectiveness of all components within our FocusDC, we conduct ablation studies for each of them. Table 10 illustrates that all techniques employed in FocusDC are essential for achieving a remarkable final performance. We observed that label reconstruction at the patch level significantly improves accuracy, consistent with the findings of previous methods [31–33].

Table 7: Comparing key patch selection strategies using various metrics, including Herding [56], K-Means [57], and Realism [33], which are current SOTA methods.

Method	Random	Herding	K-Means	Realism	Min-AS	R-AS	Max-AS
Accuracy (%)	37.9±0.5	38.4±0.1	38.2±0.1	42.0±0.1	41.6±0.3	42.6±0.8	**45.3±0.1**

Table 8: Effect of the number of patches in each compiled image. The m key patches and n low-resolution images.

Patches	$m=4, n=0$	$m=3, n=1$	$m=2, n=2$	$m=1, n=3$	$m=0, n=4$
Accuracy	32.6±0.3	**34.6±0.1**	34.2±0.2	33.2±0.2	31.8±0.5

Table 9: Impact of η on FocusDC performance.

η	0	10	30	40	100
Accuracy	32.4±0.2	33.1±0.2	**34.6±0.1**	34.2±0.2	32.8±0.3

Effectiveness of Selecting Key Patches Through Realism Score: Table 7 demonstrates the effectiveness of different key patch selection strategies using realism scores. Our method, which utilizes the maximum attention score (Max-AS) as a score metric, surpasses all compared methods. Specifically, Max-AS achieves a 3.3% accuracy improvement over current SOTA methods: Herding [56], K-Means [57] and Realism [33]. Compared to its variants, the minimum attention score (Min-AS) and random attention score (R-AS), Max-AS achieves the highest accuracy by focusing on target regions while selecting the key patches and representative low-resolution images.

Impact of the number of patches on performance. Fig. 4 shows the effect of the number of patches in synthetic images. Performance decreases as the patch count increases because each patch becomes lower-resolution, making targets harder to localize. However, using only one patch yields limited diversity and also harms performance. Balancing resolution and diversity, we set the default patch number to 4.

Impact of the Number of Key Patches in Compiled Images: By adjusting the number of key patches m and low-resolution images n, each compiled image contains m key patches and n low-resolution views. We adopt the configuration that yields the best accuracy—three key patches combined with one low-resolution image providing global context—as our default setting.

Table 10: Effectiveness of different technologies in our method. ResNet-18 is used as the validation model with an IPC of 10.

FocusDC	+CF	+ABI	+EKP	+IR	+LR
Accuracy (%)	18.4±0.3	23.6±0.1	28.2±0.2	30.9±0.1	45.3±0.1

Effect of Hyperparameters η and α: Table 9 presents the impact of the balancing factor η on FocusDC's performance, with $\eta = 30$ chosen as the default value. As shown in Fig. 7 (right), α controls the resolution of image patches, which are synthesized into multi-resolution distilled images. Setting $\alpha = 0.8$ achieves the best trade-off between accuracy and computational efficiency.

5. CONCLUSION

In this paper, we introduce FocusDC, a novel method leveraging attention mechanisms for efficient data distillation on large-scale, high-resolution datasets. FocusDC extracts key patches from target regions while preserving critical information and realism, then integrates them with low-resolution contextual backgrounds to create distilled training images. This approach enhances dataset diversity and model generalization. Moreover, FocusDC is resolution-invariant, ensuring flexibility across different image scales. Extensive experiments and ablation studies validate its effectiveness in classification and object detection, providing insights into deep learning for large-scale data and complex models.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[2] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE CVPR*, pages 1–9, 2015.

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[4] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[6] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021.

[7] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[10] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[11] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[12] Zi-Hang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. All tokens matter: Token labeling for training better vision transformers. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 18590–18602, 2021.

[13] Andrey Ignatov, Radu Timofte, Andrei Kulik, Seungsoo Yang, Ke Wang, Felix Baum, Max Wu, Lirong Xu, and Luc Van Gool. Ai benchmark: All about deep learning on smartphones in 2019. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3617–3635. IEEE, 2019.

[14] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

[15] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018.

[16] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33: 2881–2891, 2020.

[17] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in Neural Information Processing Systems*, 34:20596–20607, 2021.

[18] Kristof Meding, Luca M Schulze Buschoff, Robert Geirhos, and Felix A Wichmann. Trivial or impossible–dichotomous data difficulty masks model differences (on imagenet and beyond). *arXiv preprint arXiv:2110.05922*, 2021.

[19] Haoru Tan, Sitong Wu, Fei Du, Yukang Chen, Zhibin Wang, Fan Wang, and Xiaojuan Qi. Data pruning via moving-one-sample-out. *Advances in Neural Information Processing Systems*, 36, 2024.

[20] Lei Zhang, Jie Zhang, Bowen Lei, Subhabrata Mukherjee, Xiang Pan, Bo Zhao, Caiwen Ding, Yao Li, and Dongkuan Xu. Accelerating dataset distillation via model augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11950–11959, 2023.

[21] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6514–6523, 2023.

[22] Jiawei Du, Yidi Jiang, Vincent YF Tan, Joey Tianyi Zhou, and Haizhou Li. Minimizing the accumulated trajectory error to improve dataset distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3749–3758, 2023.

[23] Ziyao Guo, Kai Wang, George Cazenavette, Hui Li, Kaipeng Zhang, and Yang You. Towards lossless dataset distillation via difficulty-aligned trajectory matching. *arXiv preprint arXiv:2310.05773*, 2023.

[24] Zhanyu Liu, Ke Hao, Guanjie Zheng, and Yanwei Yu. Dataset condensation for time series classification via dual domain matching. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1980–1991, 2024.

[25] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Generalizing dataset distillation via deep generative prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3739–3748, 2023.

[26] Yao Lu, Xuguang Chen, Yuchen Zhang, Jianyang Gu, Tianle Zhang, Yifan Zhang, Xiaoniu Yang, Qi Xuan, Kai Wang, and Yang You. Can pre-trained models assist in dataset distillation? *arXiv preprint arXiv:2310.03295*, 2023.

[27] Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Scaling up dataset distillation to imagenet-1k with constant memory. In *International Conference on Machine Learning*, pages 6565–6590. PMLR, 2023.

[28] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[29] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.

[30] Jang-Hyun Kim, Jinuk Kim, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song. Dataset condensation via efficient synthetic-data parameterization. In *International Conference on Machine Learning*, pages 11102–11118. PMLR, 2022.

[31] 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.

[32] Muxin Zhou, Zeyuan Yin, Shitong Shao, and Zhiqiang Shen. Self-supervised dataset distillation: A good compression is all you need. *arXiv preprint arXiv:2404.07976*, 2024.

[33] 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*, 2024.

[34] Zeyuan Yin and Zhiqiang Shen. Dataset distillation in large data era. *arXiv e-prints*, pages arXiv–2311, 2023.

[35] Noel Loo, Alaa Maalouf, Ramin Hasani, Mathias Lechner, Alexander Amini, and Daniela Rus. Large scale dataset distillation with domain shift. In *Forty-first International Conference on Machine Learning*.

[36] Yongchao Zhou, Ehsan Nezhadarya, and Jimmy Ba. Dataset distillation using neural feature regression. *arXiv 2206.00719*, 2022.

[37] Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. Dataset distillation with infinitely wide convolutional networks. *Advances in Neural Information Processing Systems*, 34: 5186–5198, 2021.

[38] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4750–4759, 2022.

[39] Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and Yang You. Cafe: Learning to condense dataset by aligning features. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12186–12195, 2022. doi: 10.1109/CVPR52688.2022.01188.

[40] Saehyung Lee, Sanghyuk Chun, Sangwon Jung, Sangdoo Yun, and Sungroh Yoon. Dataset condensation with contrastive signals. In *International Conference on Machine Learning*, pages 12352–12364. PMLR, 2022.

[41] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. *arXiv preprint arXiv:2006.05929*, 2020.

[42] Zhiwei Deng and Olga Russakovsky. Remember the past: Distilling datasets into addressable memories for neural networks. *Advances in Neural Information Processing Systems*, 35:34391–34404, 2022.

[43] Songhua Liu, Kai Wang, Xingyi Yang, Jingwen Ye, and Xinchao Wang. Dataset distillation via factorization. *Advances in neural information processing systems*, 35:1100–1113, 2022.

[44] Bo Zhao and Hakan Bilen. Synthesizing informative training samples with gan. *arXiv preprint arXiv:2204.07513*, 2022.

[45] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pages 12674–12685. PMLR, 2021.

[46] Jianyang Gu, Saeed Vahidian, Vyacheslav Kungurtsev, Haonan Wang, Wei Jiang, Yang You, and Yiran Chen. Efficient dataset distillation via minimax diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[47] Duo Su, Junjie Hou, Guang Li, Ren Togo, Rui Song, Takahiro Ogawa, and Miki Haseyama. Generative dataset distillation based on diffusion model. *arXiv preprint arXiv:2408.08610*, 2024.

[48] Sangdoo Yun, Seong Joon Oh, Byeongho Heo, Dongyoon Han, Junsuk Choe, and Sanghyuk Chun. Re-labeling imagenet: from single to multi-labels, from global to localized labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2340–2350, 2021.

[49] Zhiqiang Shen and Eric Xing. A fast knowledge distillation framework for visual recognition. In *European conference on computer vision*, pages 673–690, 2022.

[50] Shitong Shao, Zeyuan Yin, Muxin Zhou, Xindong Zhang, and Zhiqiang Shen. Generalized large-scale data condensation via various backbone and statistical matching. *arXiv preprint arXiv:2311.17950*, 2023.

[51] Rahima Khanam and Muhammad Hussain. Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*, 2024.

[52] Youbing Hu, Yun Cheng, Anqi Lu, Zhiqiang Cao, Dawei Wei, Jie Liu, and Zhijun Li. Lf-vit: Reducing spatial redundancy in vision transformer for efficient image recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 2274–2284, 2024.

[53] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[54] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[55] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[56] Max Welling. Herding dynamical weights to learn. In *Proceedings of the 26th annual international conference on machine learning*, pages 1121–1128, 2009.

[57] Edward W Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *biometrics*, 21:768–769, 1965.

[58] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[59] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008.

[60] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part VI 13*, pages 446–461. Springer, 2014.

[61] William Yang, Ye Zhu, Zhiwei Deng, and Olga Russakovsky. What is dataset distillation learning? *arXiv preprint arXiv:2406.04284*, 2024.

[62] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[63] Dong Yin, Ramchandran Kannan, and Peter Bartlett. Rademacher complexity for adversarially robust generalization. In *International conference on machine learning*, pages 7085–7094. PMLR, 2019.

[64] Víctor Elvira, Luca Martino, and Christian P Robert. Rethinking the effective sample size. *International Statistical Review*, 90(3):525–550, 2022.

[65] Raphael Gontijo-Lopes, Sylvia Smullin, Ekin Dogus Cubuk, and Ethan Dyer. Tradeoffs in data augmentation: An empirical study. In *International Conference on Learning Representations*, 2021.

[66] Mengzhao Chen, Mingbao Lin, Zhihang Lin, Yuxin Zhang, Fei Chao, and Rongrong Ji. Smmix: Self-motivated image mixing for vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17260–17270, 2023.

[67] Mengzhao Chen, Mingbao Lin, Ke Li, Yunhang Shen, Yongjian Wu, Fei Chao, and Rongrong Ji. Cf-vit: A general coarse-to-fine method for vision transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7042–7052, 2023.

[68] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

6. Appendix

6.1. IMPLEMENTATION DETAILS

6.1.1. Pre-training ViT Models

For the ImageNet-1K dataset, we directly use the model pre-trained by LF-ViT [52], which is based on the implementations of Deit-S [52] and LV-ViT-S [12]. This model performs inference at both the standard resolution of 224 $\times$ 224 and a higher resolution of 288 $\times$ 288, efficiently extracting crucial information patches for dataset distillation. To further reduce inference time, we disable the Focus stage in the LF-ViT implementation. More details and features of LF-ViT can be found on the official website. For the lower resolution and smaller scale Tiny-ImageNet dataset, we train a modified version of the Deit-S-based LF-ViT [52] from scratch to extract key information patches. Specifically, we reduce the model's depth to 4 layers, set the patch size to 4×4, adjust the embedding dimension to 192, and reduce the number of heads to 3. This modified model is trained from scratch using the same hyperparameters as those used for ImageNet-1K. The source code of all experiments is available at: https://github.com/edgeai1/FocusDC

6.1.2. FocusDC Implementation Details

We maintain a fixed side ratio $\alpha = 0.8$ and a balancing factor $\eta = 30$ for both the ImageNet-1K and Tiny-ImageNet datasets. To compile each image $\tilde{\boldsymbol{x}}_j$ in the distilled dataset $\mathcal{T}$, we set N and M to IPC and 3$\times$IPC, respectively. The compile process involves concatenating three key patches from the key information collection $\tilde{\mathcal{T}}_c$ and one low-resolution background image from $\mathcal{T}_c'$, resulting in the compiled image as described by Eq. 8. For instance, at an IPC of 100, we select 300 key information patches and 100 downsampled low-resolution images with background information, ensuring the synthesis of a diverse and representative image. This approach adapts to different IPC values to accurately reflect the dataset's variability. Aligned with techniques from SRe^{2}L [31] and RDED [33], we employ Fast Knowledge Distillation [49] to relabel distilled images. Each distilled image $\tilde{\boldsymbol{x}}_j$ is randomly cropped into several patches, with their coordinates recorded within $\tilde{\boldsymbol{x}}_j$. Soft labels $\tilde{y}_j^k$ are generated and stored for each k-th patch. These labels are aggregated to construct a comprehensive label $\tilde{y}_j$ for each image, facilitating nuanced and accurate labeling reflective of the diverse visual features captured in the compiled images.

Table 11: Training hyper-parameters for Tiny-ImageNet and ImageNet-1K.

Config	Tiny-ImageNet	ImageNet-1K
Optimizer	SGD	AdamW
Base learning rate	0.2	0.001
Weight decay	1e-4	0.01
Optimizer momentum	0.9	$\beta_1, \beta_2 = 0.9, 0.999$
Batch size	256	128
Learning rate schedule	Cosine decay	Cosine decay
Training epoch	800	300
Augmentation	RandomResizedCrop	RandomResizedCrop

Training on Distilled Dataset: We use a model with the same architecture as the validation model, pre-trained on the corresponding original and full datasets, to generate soft labels for the synthesized images. For Tiny-ImageNet, our teacher model is pre-trained on the complete Tiny-ImageNet dataset, following the hyperparameters in [31]. When training the validation model on the distilled Tiny-ImageNet dataset, we use the hyperparameters shown in Table 11. For ImageNet-1K, all teacher models use pre-trained models from the `torchvision` library. When training the validation model on the distilled ImageNet-1K dataset, we follow the parameters in Table 11. Both datasets are augmented by CutMix with a mix probability $p = 1.0$ and a beta distribution $\beta = 1.0$.

For the object detection task, we selected samples from the ImageNet-1K dataset corresponding to the categories in COCO2017 [62] and generated a dataset based on the IPC settings. YOLOv11x [51] was used as the teacher model to annotate this dataset. Then, YOLOv11s and YOLOv11n

Table 12: Our FocusDC incorporates dynamic fine-tuning to further improve performance. It is worth noting that to further highlight the accuracy improvement brought by dynamic fine-tuning, the accuracy of FocusDC is based on the results after training for 1000 epochs.

Architecture	Method	IPC			
		1	10	50	100
ResNet-18 (69.8)	FocusDC	10.7±0.2	52.5±0.1	63.1±0.1	68.0±0.2
	FocusDC + DFT	14.7±0.1	57.6±0.1	65.8±0.2	69.1±0.1
ResNet-50 (76.2)	FocusDC	6.92±0.1	56.5±0.2	70.1±0.3	71.1±0.2
	FocusDC + DFT	12.3±0.2	62.9±0.2	72.8±0.2	74.3±0.2
ResNet-101 (77.4)	FocusDC	7.3±0.2	53.8±0.2	71.5±0.2	73.5±0.1
	FocusDC + DFT	14.7±0.3	58.3±0.2	72.6±0.3	76.4±0.1
MobileNet-V2 (71.8)	FocusDC	8.4±0.1	49.5±0.1	61.6±0.3	66.0±0.1
	FocusDC + DFT	12.1±0.3	56.0±0.1	66.4±0.1	69.0±0.2
EfficientNet-B0 (76.3)	FocusDC	12.7±0.2	50.4±0.2	67.9±0.1	68.5±0.2
	FocusDC +DFT	17.6±0.4	59.9±0.2	73.4±0.1	74.5±0.2

were trained from scratch on the annotated dataset, and their performance was evaluated on the COCO2017 validation set. All training hyperparameters were kept identical to the official YOLOv11 [51] configuration.

Dynamic Fine-Tuning Parameter Settings: During the Dynamic Fine-Tuning (DFT) process (detailed in Appendix 6.2.1), we randomly select images with the same IPC from the original dataset in each iteration to form a new dataset for fine-tuning. The hyperparameters for fine-tuning match those used for training the validation model on the synthesized dataset. We set the learning rate to 0.00025, with 50 epochs and a batch size of 64. The learning rate for MobileNet-v2 during DFT is set to 0.001.

6.2. FURTHER EXPERIMENTAL RESULTS

6.2.1. Dynamic Fine-Tuning

Following the training of model $\phi_{\boldsymbol{\theta}_{stu}}$ on the distilled dataset $\mathcal{T}$, we implement the Dynamic Fine-Tuning (DFT) process. The DFT process involves fine-tuning the model on subsets of the original dataset that are dynamically sampled at each epoch. To preserve consistency with the structural properties of the synthetic dataset, images are randomly selected at an IPC level from each category to form new datasets for fine-tuning. This strategy is systematically applied throughout each epoch, introducing variability and generating a unique dataset for fine-tuning in every cycle. This approach significantly enhances the diversity of the data without additional training overhead, thereby boosting the model's generalization ability across diverse data representations. Furthermore, the DFT methodology not only capitalizes on the attributes of synthetic data but also closely aligns the model's performance with real-world data distributions, culminating in notable enhancements in performance.

ImageNet-1K Dataset: Table 12 presents the experimental results of training FocusDC for 1000 epochs and combining it with DFT on the ImageNet-1K dataset. We find that DFT further improves the performance of FocusDC across all architectures. In particular, when IPC=100, FocusDC + DFT demonstrates exceptionally small declines in accuracy—0.7%, 1.9%, 1.0%, 2.8%, and 1.8% across the evaluated models—almost achieving performance equivalent to training with the complete dataset. These minimal accuracy losses highlight the robustness of FocusDC when augmented by DFT, effectively leveraging the combined strengths of focused data distillation and iterative fine-tuning. The success of this approach underscores that merging FocusDC with DFT offers a powerful and efficient strategy for minimizing accuracy losses in high-scale learning environments, making it particularly suitable for scenarios where resources are limited but high performance is imperative.

Table 13: Comparison with SOTA baseline dataset distillation methods on the Tiny-ImageNet dataset.

Architecture	Method	IPC			
		1	10	50	100
ResNet-18 (59.6)	SRe^{2}L	2.62±0.1	16.1±0.2	41.1±0.4	49.7±0.3
	SCDD	-	31.6±0.1	45.9±0.2	-
	GVBSM	-	47.6±0.3	51.0±0.4	-
	RDED	9.7±0.4	41.9±0.2	**58.2±0.1**	59.1±0.1
	FocusDC (Ours)	16.5±0.2	49.4±0.1	56.7±0.1	59.2±0.1
	FocusDC + DFT (Ours)	**21.2±0.1**	**51.1±0.1**	56.9±0.1	**59.4±0.1**
ResNet-50 (62.8)	SRe^{2}L	2.0±0.4	15.5±0.5	42.2±0.5	51.2±0.4
	GVBSM	-	48.7±0.2	52.1±0.3	-
	RDED	8.1±0.3	45.3±0.2	**61.6±0.3**	**62.6±0.1**
	FocusDC (Ours)	14.6±0.3	53.4±0.1	59.8±0.2	62.0±0.2
	FocusDC + DFT (Ours)	**19.9±0.2**	**54.1±0.1**	60.9±0.2	62.2±0.2
ResNet-101 (67.0)	SRe^{2}L	1.9±0.1	14.6±1.1	42.5±0.2	51.5±0.3
	GVBSM	-	48.8±0.4	52.3±0.1	-
	RDED	3.8±0.1	22.9±3.3	41.2±0.4	65.2±1.1
	FocusDC (Ours)	13.2±0.2	55.5±0.3	63.2±0.2	66.4±0.2
	FocusDC + DFT (Ours)	**19.4±0.2**	**56.3±0.2**	**64.1±0.2**	**67.0±0.1**
MobileNet-V2 (45.2)	SRe^{2}L	2.0±0.3	7.3±0.2	19.5±0.4	22.7±0.6
	RDED	4.1±0.3	27.4±0.3	40.1±0.2	42.6±0.3
	FocusDC (Ours)	5.8±0.2	34.8±0.2	42.2±0.1	44.6±0.2
	FocusDC + DFT (Ours)	**5.9±0.3**	**36.6±0.2**	**43.6±0.1**	**45.0±0.3**
EfficientNet-B0 (41.6)	SRe^{2}L	1.0±0.3	7.8±0.4	17.5±0.7	20.9±0.3
	RDED	1.3±0.1	18.3±0.4	38.2±0.3	40.4±0.2
	FocusDC (Ours)	7.5±0.1	32.9±0.2	40.4±0.2	41.4±0.1
	FocusDC + DFT (Ours)	**9.0±0.1**	**33.5±0.2**	**41.2±0.3**	**42.7±0.1**

Table 14: Compiled time and memory consumption on ImageNet-1K using a single RTX-4090 GPU. Time Cost is measured in seconds for generating 100 images simultaneously. Peak GPU memory usage is recorded for a batch size of 100, following the official SRe^{2}L [31] implementation. RDED-All indicates selection for all images in each category, whereas RDED only a random sample of 300 images per category.

Architecture	Method	Time Cost (s)	Peak Memory (GB)
ResNet-18	SRe^{2}L	211.32	9.14
	RDED	3.99	1.57
	RDED-All	26.34	8.63
MobileNet-V2	SRe^{2}L	378.32	12.93
	RDED	6.50	2.35
	RDED-All	31.27	11.06
EfficientNet-B0	SRe^{2}L	441.24	11.92
	RDED	7.32	2.34
	RDED-All	37.83	10.96
Deit-S	FocusDC (Ours)	8.67	6.84
LV-ViT-S	FocusDC (Ours)	10.72	8.57

Tiny-ImageNet Dataset: Table 13 evaluates our method, FocusDC, integrated with DFT on the Tiny-ImageNet dataset, showing similar trends as observed with the ImageNet-1K dataset. Notably, using EfficientNet-b0 at an IPC of 100, FocusDC not only matches but also exceeds the performance of baseline models by 1.1±0.1%. This improvement stems from DFT's random selection of IPC samples each round, enhancing the diversity of training data and thus boosting performance. This result highlights the benefits of combining FocusDC with DFT to optimize performance under data constraints.

6.2.2. Additional Experiments

Compiled Time and Memory Consumption: Table 14 presents the compiled time and memory consumption when utilizing a single RTX-4090 GPU on the ImageNet-1K dataset. Unlike SRe^{2}L, which consumes substantial resources, FocusDC significantly reduces both compiled time and memory usage. Specifically, FocusDC cuts the compiled time down to 8.67 seconds for Deit-S and 10.72 seconds for LV-ViT-S, while maintaining peak memory usage below 7 GB for Deit-S and slightly above 8 GB for LV-ViT-S [12]. Compared with RDED, FocusDC demonstrates a competitive advantage by achieving a more balanced utilization of time and GPU memory, thereby presenting a resource-efficient solution for dataset distillation. The high efficiency of FocusDC is attained through a strategy of downsampling images before their input into the ViT model. This approach not only reduces the computational load but also enables a more flexible allocation of GPU resources through adaptive resizing of mini-batches. This efficiency is primarily due to the memory demands in our distillation process, which occur mainly during the parallel extraction of key informative patches within a mini-batch. Furthermore, the optimization-free nature of FocusDC means that the distillation time per image depends on the size of the pre-trained ViT model used.

Table 15: When the input resolution for ViT is increased from 224×224 to 288×288, we evaluate the accuracy of the compiled images generated by FocusDC. All accuracies were obtained after training for 1000 epochs on their respective datasets.

Architecture	IPC			
	1	10	50	100
R18	10.7±0.2	52.5±0.1	63.1±0.1	68.0±0.2
R18#288	9.6±0.2	52.6±0.1	64.0±0.1	67.7±0.2
Mv2	8.4±0.1	49.5±0.1	61.6±0.3	66.0±0.1
M2#288	7.7±0.1	50.1±0.1	61.5±0.2	64.2±0.2

Scaling up to Higher Resolutions: When the input resolution of ViT is expanded from 224×224 to 288×288, under the same hyperparameters, we evaluate the accuracy of compiled images using ResNet-18 and MobileNet-v2 on the ImageNet-1K dataset, as shown in Table 15. We discover that despite increasing the resolution of the image input to ViT from 224×224 to 288×288, there is a slight decrease in accuracy. This phenomenon could be attributed to two factors. Firstly, a larger image resolution makes it more difficult to locate targets within the image, potentially leading to a decrease in the accuracy of the compiled dataset. Secondly, when training the validation model from scratch, all images are resized to the resolution of 224×224. Reducing a higher-resolution image to this lower standard may result in more significant information loss.

Table 16: Comparative analysis of the accuracy and computational cost (measured in FLOPs) of training Deit-S on original versus downsampled images of ImageNet-1K.

Resolutions	224×224	112×112
Accuracy	79.8%	73.3%
FLOPs	4.60G	1.10G

The advantages of downsampling: The FocusDC synthetic dataset uses downsampled images to locate target regions for the following reasons: (1) Significant computational savings: As shown in Table 16, downsampling reduces FLOPs by 4.2 times. (2) Facilitates dataset synthesis: It allows us to directly select low-resolution background images from the downsampled images to synthesize the final distilled image.

6.3. THEORETICAL ANALYSIS

6.3.1. Background and Definitions

To analyze how the dataset distillation with an attention-based region selection affects the generalization ability of models on a testing dataset, we employ Rademacher Complexity [63] as a theoretical framework. We first present the setup and the analysis of our proposed FocusDC method, followed by the empirical validation and the insights.

Original Dataset $\mathcal{S}$: The original dataset, denoted as $\mathcal{S}$, consists of $|\mathcal{S}|$ samples, represented by $\{\boldsymbol{x}_i\}_{i=1}^{|\mathcal{S}|}$.

Distilled Dataset $\mathcal{T}$: The distilled dataset, $\mathcal{T}$, is created by merging m samples from $\mathcal{S}$ based on key regions identified by an attention mechanism such as a Vision Transformer (ViT) and n samples with background information. This results in $\mathcal{T}$ samples, $\{\tilde{\boldsymbol{x}}_i\}_{i=1}^{|\mathcal{T}|}$, where $|\mathcal{T}| < |\mathcal{S}|$.

Rademacher Complexity: Rademacher Complexity measures the capacity of a class of functions to fit random noise, providing a metric for the complexity and generalization capability of hypothesis classes:

$$\hat{R}_{\mathcal{S}}(\mathcal{H}) = \mathbb{E}_{\sigma}\left[\sup_{h\in\mathcal{H}} \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|} \sigma_i h(\boldsymbol{x}_i)\right],$$

where σ_i are independent random variables taking values $+1$ or -1 with equal probability. We apply this metric when evaluating the distilled datasets because it can provide insight into whether the distillation process preserves the richness of the hypothesis space or if it overly simplifies the dataset, potentially losing important variances needed for higher generalization.

6.4. Impact of Dataset Distillation of FocusDC

For the distilled dataset $\mathcal{T}$, the Rademacher Complexity becomes:

$$\hat{R}_{\mathcal{T}}(\mathcal{H}) = \mathbb{E}_{\sigma}\left[\sup_{h\in\mathcal{H}} \frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|} \sigma_i h(\tilde{\boldsymbol{x}}_i)\right].$$

Each distilled data instance $\tilde{\boldsymbol{x}}_i = \text{concatenate}(\{\boldsymbol{x}_q^\star\}_{q=1}^m, \{\boldsymbol{x}_l'\}_{l=1}^n)$, where $\boldsymbol{x}^\star$ represents the key sub-region data and $\boldsymbol{x}'$ means the down-scaled low resolution data with backgound information.

Note that the term $1/|\mathcal{T}|$ determines the scaling of the sum of fits to random labels (noise) in the Rademacher Complexity formula. When analyzing a dataset that has undergone distillation to produce $\mathcal{T}$, where each sample $\tilde{\boldsymbol{x}}_i$ aggregates the informational content of multiple samples from the original dataset, the actual number of samples $|\mathcal{T}|$ might not accurately reflect the dataset's complexity. Instead, the Efficient Sample Size (ESS) [64] is applied to represent the number of independent observations in a dataset that would provide the same amount of information as the actual dataset, which can be noted as $|\mathcal{T}_{\text{eff}}|$. If $|\mathcal{T}_{\text{eff}}|$ represents a more accurate measure of the independent information content in $\mathcal{T}$, the complexity measure can be adjusted to:

$$\hat{R}_{\mathcal{T}}(\mathcal{H}) = \mathbb{E}_{\sigma}\left[\sup_{h\in\mathcal{H}} \frac{1}{|\mathcal{T}_{\text{eff}}|}\sum_{i=1}^{|\mathcal{T}|} \sigma_i h(\tilde{\boldsymbol{x}}_i)\right].$$

This adjustment recognizes that the effective diversity and informational independence in $\mathcal{T}$ might be greater than simply counting $|\mathcal{T}|$, hence potentially leading to a more accurate estimation of how the hypothesis class $\mathcal{H}$ will perform.

The complexity induced by each new sample $\tilde{\boldsymbol{x}}_i$ can reduce the variance among samples, as they inherently represent a more uniform distribution of the key features and contexts of the original dataset. The formula for Rademacher Complexity has to consider the effective sample size $|\mathcal{T}_{\text{eff}}|$ that accounts for this aggregation:

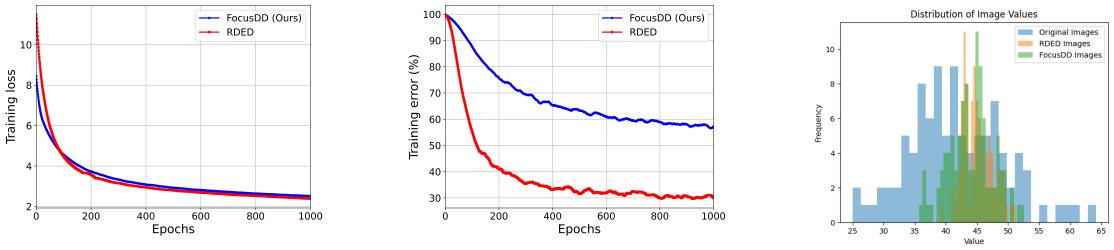
$$|\mathcal{T}_{\text{eff}}| = |\mathcal{T}| \times (m * \gamma + n * \beta),$$

where γ and β represent the degression parameters due to selecting only the key regions or using down-scaled data, which range from 0 to 1. The setting $\gamma = \beta = 1$ means that we naively concatenate $m + n$ original data instances.

Similarly, we can determine $\tilde{\boldsymbol{x}}_i$ and $|\mathcal{T}_{\text{eff}}|$ for two baseline methods as shown in Table. 17: Naïve and RDED [33]. A higher $|\mathcal{T}_{\text{eff}}|$ indicates that each sample in $\mathcal{T}$ contains more "independent-like" information than initially apparent, suggesting that $\mathcal{T}$ may exhibit a lower Rademacher Complexity

Table 17: **Rademacher Complexity Comparison with the same IPC.** Naïve denotes randomly selecting $|\mathcal{T}|$ samples from the original dataset, RDED concatenates $(m+n)$ random sub-region samples. τ is a regression parameter due to selecting only the sub-regions.

Method	$\tilde{\boldsymbol{x}}_i$	$\lvert\mathcal{T}_{\text{eff}}\rvert$
Naïve	$\boldsymbol{x}_i$	$\lvert\mathcal{T}\rvert$
RDED	concatenate$(\{\boldsymbol{x}_j^{\text{rand}}\}_{j=1}^{m+n})$	$\lvert\mathcal{T}\rvert \times (m+n) * \tau$
FocusDC (Ours)	concatenate$(\{\boldsymbol{x}_q^{\star}\}_{q=1}^{m}, \{\boldsymbol{x}_l'\}_{l=1}^{n})$	$\lvert\mathcal{T}\rvert \times (m * \gamma + n * \beta)$



(a) (b) (c)

Figure 8: The diversity of the compiled dataset is assessed by analyzing the training loss and accuracy metrics on the compiled image training set. (a) Training loss. (b) Training error. (c) The signal-to-noise distribution of images within the same category for the full dataset and those distilled by RDED and FocusDC. Training loss on compiled images. All methods employ MobileNet-v2 and are executed on the ImageNet-1K dataset with IPC = 10.

than expected if assessed solely based on $|\mathcal{T}|$. Generally, a lower Rademacher Complexity correlates with better generalization capabilities, indicating that models trained on $\mathcal{T}$ might generalize better than anticipated based solely on $|\mathcal{T}|$. This enhanced generalization is why RDED and our proposed method significantly outperform the Naïve approach, which relies on random sample selection.

Our method employs strategies to achieve a larger $|\mathcal{T}_{\text{eff}}|$ than RDED. Our realism score s_i^{real} combines the predictive confidence score and the maximum attention region score. When selecting samples, it does not only consider the information richness of the samples but also the information density of the target regions within these samples. Together, these factors improve $|\mathcal{T}_{\text{eff}}|$ and enhance generalization capabilities, as confirmed by the results in Table 8 for $m \neq 0$, which reflect the combined effect of both strategies.

Our method, FocusDC, is also designed to reduce model complexity within the Hypothesis Space. Richer samples may enable the functions h in $\mathcal{H}$ to be less complex, as each sample encompasses a broader range of information, potentially simplifying the learning problem. This hypothesis is supported by the results in Table 2, which demonstrate that simpler backbone models using FocusDC data achieve outcomes comparable to those of more complex models.

Quantifying the Diversity and SNR of Synthetic Images. We employ the method outlined in Gontijo-Lopes et al. [65] to assess the diversity of compiled images. According to Gontijo-Lopes et al. [65], greater dataset diversity presents more challenges for the training process to converge, often resulting in larger loss values and longer training times. Fig. 8(a) compares the training loss of our method with the SOTA method RDED [33] on compiled datasets. Initially, our FocusDC method starts with lower loss values but ends with higher losses than RDED after training. Moreover, Fig. 8(b) illustrates significant differences in accuracy tests on the training dataset, indicating that images synthesized using our method are more diverse and thus harder to train. This observation aligns with the conclusions in Gontijo-Lopes et al. [65], confirming that our approach generates more diverse compiled images, making the training process more challenging but potentially leading to more robust models.

Figure 9: Visualization of the FocusDC-distilled images on different tasks.

Fig. 8(c) illustrates the distribution of signal-to-noise ratios (SNR)[1] for the original dataset and datasets processed by two different distillation methods, within the same category. The SNR distribution of the original images is relatively concentrated, with most values ranging between 30 and 58. The SNR of images processed by RDED [33] shifts to the right, primarily distributed between 42 and 50. In contrast, images processed by FocusDC exhibit a wider SNR distribution, spanning from 36 to 53. Although the average SNR of RDED images is the highest at 45.1, the average SNR for FocusDC images is 44.0, closer to the original dataset's average SNR of 41.7. This indicates that the FocusDC method effectively enhances image quality while preserving the characteristics of the original data, thereby demonstrating superior balanced performance in practical applications.

6.4.1. Remarks

The proposed distillation method, FocusDC, is expected to enhance generalization by utilizing more informative and representative samples. The associated reduction in Rademacher Complexity indicates a diminished capacity for fitting random noise, which typically suggests improved performance on unseen data.

The practical implementation may encounter challenges, such as increased computational overhead from processing larger $\tilde{\boldsymbol{x}}_i$ values. Furthermore, there is a risk of information redundancy if the parameters m and n are not optimally selected.

6.5. VISUALIZATIONS of SYNTHETIC IMAGE

Fig. 9 demonstrates the images synthesized by FocusDC for classification and detection tasks. By leveraging attention mechanisms to localize key targets, FocusDC ensures that each synthesized image contains relevant objects of interest. This capability enables effective application to dense tasks such as object detection. Fig. 10 in Appendix 6.5 visualizes compiled images generated by different SOTA methods. SRe^{2}L, SCDD, and GVBSM produce blurrier images, due to overreliance on specific models during dataset compression, which hampers generalization. In contrast, RDED and our FocusDC generate more realistic images by cropping key patches from real image locations. Unlike RDED, our method includes both key patches and contextual backgrounds, enhancing realism and diversity. Attention mechanism used in our method, validated in the vision community [52, 66–68], improves interpretability and offers deeper insights into dataset distillation.

[1]We applied a 3×3 Laplacian kernel to filter the images to extract their high-frequency components. Then, we calculated the sum of the absolute values of the convolution results between the image and this matrix, using this to estimate the standard deviation of the noise. Finally, based on the definition of signal-to-noise ratio, we computed the SNR distribution for the entire dataset.

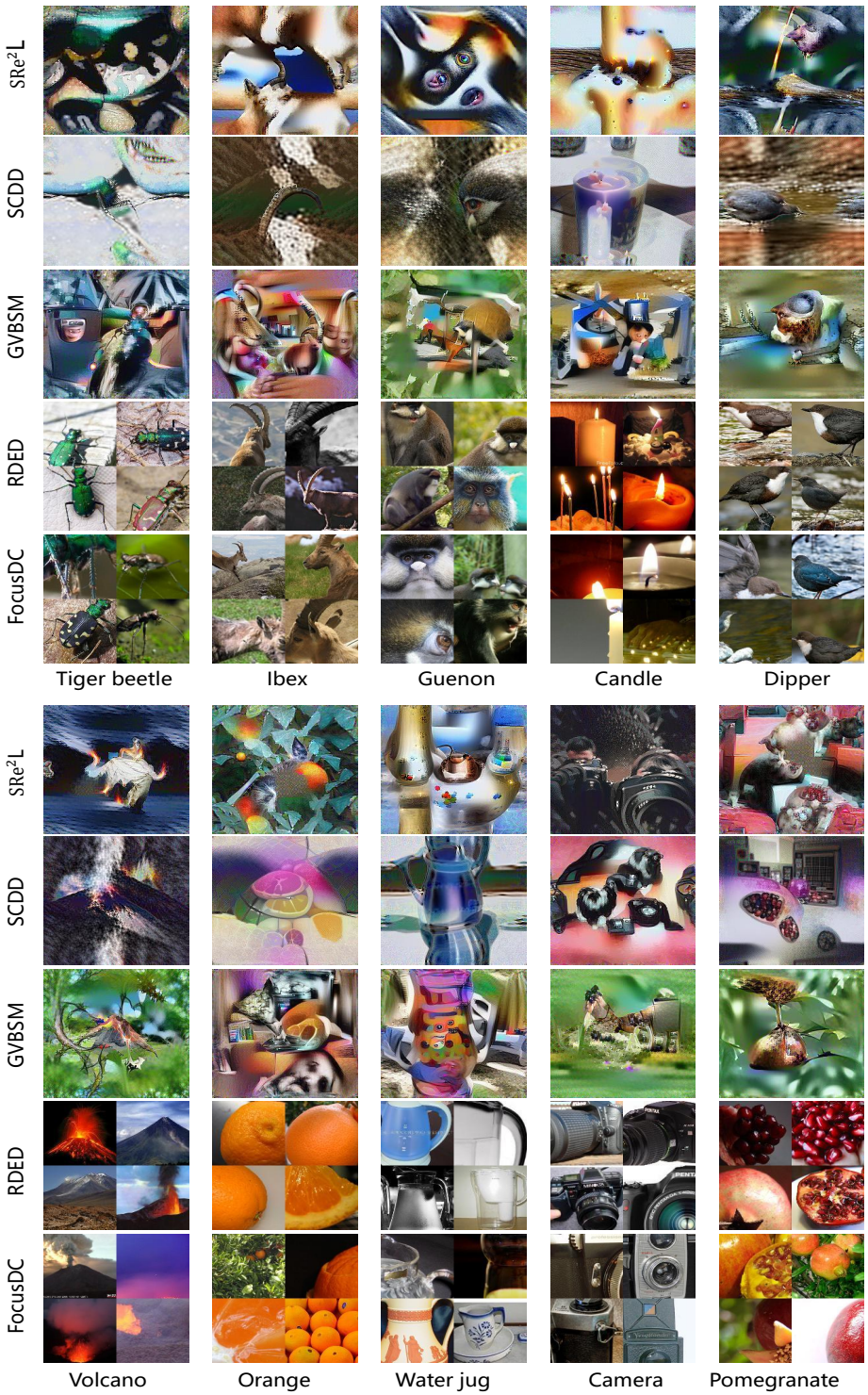


Figure 10: Compiled data visualization on ImageNet-1K from SRe^{2}L [31], SCDD [32], GVBSM [50], RDED [33] and FocusDC.