Color-Oriented Redundancy Reduction in Dataset Distillation

Bowen Yuan **Zijian Wang** **Mahsa Baktashmotlagh** **Yadan Luo** **Zi Huang**
{bowen.yuan, zijian.wang, m.baktashmotlagh, y.luo, helen.huang}@uq.edu.au
The University of Queensland

Abstract

Dataset Distillation (DD) is designed to generate condensed representations of extensive image datasets, enhancing training efficiency. Despite recent advances, there remains considerable potential for improvement, particularly in addressing the notable redundancy within the color space of distilled images. In this paper, we propose AutoPalette, a framework that minimizes color redundancy at the individual image and overall dataset levels, respectively. At the image level, we employ a palette network, a specialized neural network, to dynamically allocate colors from a reduced color space to each pixel. The palette network identifies essential areas in synthetic images for model training and consequently assigns more unique colors to them. At the dataset level, we develop a color-guided initialization strategy to minimize redundancy among images. Representative images with the least replicated color patterns are selected based on the information gain. A comprehensive performance study involving various datasets and evaluation scenarios is conducted, demonstrating the superior performance of our proposed color-aware DD compared to existing DD methods. The code is available at `https://github.com/KeViNYuAn0314/AutoPalette`.

1 Introduction

Large-scale training data is essential for achieving high model performance. However, the sheer volume of the data poses significant challenges, including computational inefficiency, prolonged training times, and substantial storage overhead. *Data Distillation* (DD) [40] offers a promising solution to this problem. By synthesizing a smaller dataset from the original dataset, DD allows models trained on the distilled dataset to attain comparable performance to those trained on the full dataset, thereby reducing the resources needed for training.

Existing DD primarily minimizes the difference between the network trained on the full dataset and the network trained on the synthetic dataset. Different surrogate functions have been implemented to quantify such differences, including performance matching [40, 29], feature distribution matching [39, 46] and model gradient matching [47, 3]. Generally speaking, DD considers the synthetic images as parameters and directly optimizes them. Building on this concept, parameterization-based dataset distillation (PDD) extends DD by enhancing the storage utility and reducing redundancy in the image space. Parameterization-based DD methods represent the synthetic dataset in a lower-dimensional space and then, reconstruct synthetic images for model training. Current parameterization-based DD includes: learning in a spatially down-sampled space [20], factorizing distilled images [9, 24], optimizing latent embeddings and generators [45, 4], and selecting informative frequency bands [35].

While the existing PDD methods have shown promising results, most of the methods overlook the redundancy in the color space, thereby falling short of achieving optimal parameterization performance. We argue that reducing the number of unique colors within one image can have *minimal* impact on the low-level discriminative features (*e.g.*, shapes, edges) required for model

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-1687

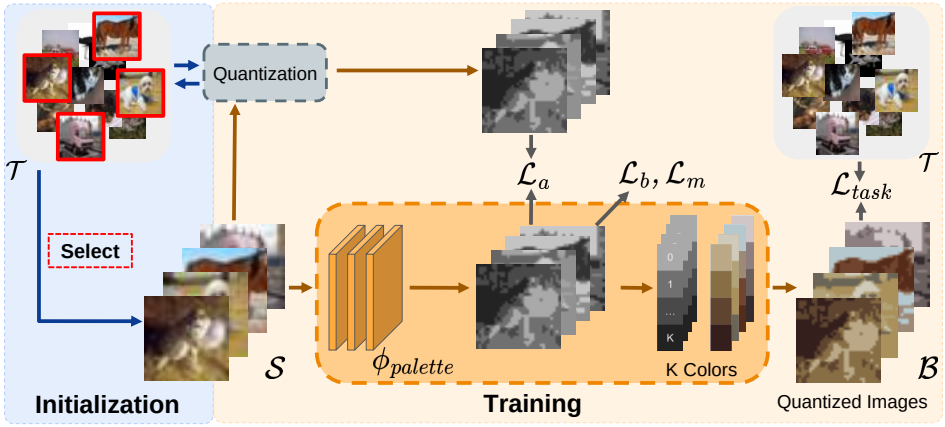

Figure 1: The overview of the proposed AutoPalette framework. Initialization: We compare the information gain of quantized images to select the images used in the initialization stage. Training: We forward the synthetic data to the palette network to obtain the color-reduced images. The objective functions of palette network include $\mathcal{L}_a$, $\mathcal{L}_b$, $\mathcal{L}_m$ and $\mathcal{L}_{task}$. The synthetic dataset is updated by solely optimizes $\mathcal{L}_{task}$.

training. Moreover, images within the same class typically share similar color distributions; therefore, dedicating storage to *unique* class patterns rather than storing the replicated color information would be more cost-effective.

To address the limitations of existing PDD approaches, we propose a color-oriented redundancy reduction framework, namely AutoPalette. Specifically, AutoPalette contains an efficient plug-and-play palette network to tackle the issue of color space redundancy within one image. This palette network transforms 8-bit color images into representations with fewer colors (*e.g.*, 4-bit) by aggregating pixel-level color information from input images. To enhance the color utility, we design two additional losses on top of the dataset distillation loss: the maximum color loss and the palette balance loss. The maximum color loss ensures that each color in the reduced color space is allocated to at least one pixel, while the palette balance loss balances the number of pixels allocated to each color. The palette network synthesizes image datasets with a reduced color space while preserving essential features of the original images. Furthermore, we equip AutoPalette with a color-guided initialization module to suppress the redundancy in-between synthetic images. The module selects the samples with low replication after color condensation as the synthetic set initialization, whereas information gain is adopted to quantify replication.

Contributions. We propose AutoPalette, a color-oriented redundancy reduction framework for data distillation tasks, enhancing storage efficiency by reducing the number of colors in the images while preserving essential features; We seamlessly equip the distillation framework with a guided initialization strategy that selects images with diverse structures in the reduced color space for initialization; Extensive experimental results on three benchmark datasets show that the model trained on the 4-bit images synthesized by our framework achieve competitive results compared to the models trained on 8-bit images synthesized by other DD methods. With the same storage budget, our method outperforms others by 1.7%, 4.2% on CIFAR10 and CIFAR100.

2 Related Work

2.1 Dataset Distillation

Dataset distillation aims to synthesize a small but informative dataset, enabling models trained on these synthetic data to achieve comparable performance to those trained on the complete dataset. Wang *el al.* [40] firstly proposed a meta-model learning approach to optimize a synthetic dataset matching the model performance of large scale dataset. Early works adopt performance matching frameworks [28, 30, 49, 37], and optimize synthetic data using model performance rolling over the training process on the original dataset. Distribution matching methods [39, 46, 48, 32] address high complexity issues in bi-level optimization by matching one step feature distributions between synthetic

https://doi.org/10.52202/079017-1687

(a) (b) (c)

Figure 2: The visualization of (a) images under 8, 6, 3, 1-bit color depths (b-c) color condensed synthetic images and their color palette. (b) our full model (c) our full model without palette loss. The larger difference among rows of a color palette indicates better color utilization.

data and original real data. Gradient matching [47, 44, 25] and trajectory matching [3, 6, 12, 10] approaches aim to match model parameters' gradients for single or multiple training steps, leading networks trained on synthetic data and original data to follow similar gradient descent trajectories.

2.2 Parameterization-based Dataset Distillation

Apart from finding matching objectives between the synthetic dataset and the original full dataset, another aspect of data distillation involves appropriately parameterizing synthetic data in different yet more efficient representatives in memory space. Without storing synthetic data as individual spatial representations, parameterization comprehends mutual characteristics between data instances and regenerates more data instances of the original input representations. IDC [20] stores images in a low-resolution manner to conserve storage resources, and upsamples to the original scale for usage. Factorization methods conjecture inter-class data share mutual and independent information and generate synthetic data based on combinations of bases. Bases can be either spatial representations [9], frequency domains [41], or embeddings decoded by networks [23, 24, 41, 45, 5, 38].

2.3 Color Quantization

Color quantization [31, 8, 1, 43] intends to aggregate similar colors and transform them using one representative color. Accordingly, images with a reduced color palette require less storage as the pixel values can be encoded in fewer bits. To uphold optimal image authenticity, traditional color quantization methods such as Median Cut [15], dithering [13], and OCTree [11] typically employ color quantization as a color clustering problem. They commonly devise strategies to identify similar or neighboring colors for quantization purposes. On the other hand, parameter-based methods [36, 27, 16, 17] not only rely on predefined heuristics but also leverage neural networks to learn patterns and relationships to compress images to lower bits.

3 Methodology

3.1 Notations and Preliminary

Dataset distillation aims to learn a small but representative synthetic dataset $\mathcal{S} = \{(\tilde{x}^i, \tilde{y}^i)\}_{i=0}^{|\mathcal{S}|}$ from a given large scale dataset $\mathcal{T} = \{(x^i, y^i)\}_{i=0}^{|\mathcal{T}|}$. Here, $|\mathcal{S}|$ and $|\mathcal{T}|$ denote the number of samples in the synthetic dataset and original large dataset, where $|\mathcal{S}| \ll |\mathcal{T}|$. By training on the synthetic dataset $\mathcal{S}$, a model $\phi(\cdot;\theta)$ is aimed to achieve performance comparable to that of a model trained on the original dataset $\mathcal{T}$. The objective of dataset distillation can be formulated as a bi-level optimization problem:

$$\min_{\mathcal{S}} \mathbb{E}_\theta[\mathcal{L}(\mathcal{T}, \theta_{\mathcal{S}})], \text{ where } \theta_{\mathcal{S}} = \arg\min_{\theta} \mathcal{L}(\mathcal{S}, \theta), \tag{1}$$

where $\mathcal{L}(\cdot,\cdot)$ and θ represent loss function and parameters of the networks, respectively. The inner-loop optimizes the network on the synthetic dataset and the outer loop evaluates the trained network on the real dataset.

https://doi.org/10.52202/079017-1687

The bi-level meta learning in Eq. 1 requires inner-loop training during every training steps, and thus suffers from inevitable computational cost. Therefore, some of the existing methods [47, 3, 46] try to avoid unrolled back-propagation in the inner loop using various surrogate objectives, thereby the optimal synthetic dataset $\mathcal{S}^*$ can be obtained by optimizing:

$$\mathcal{S}^* = \arg\min_{\mathcal{S}} \mathbb{E}_\theta[\mathcal{L}(\phi(\mathcal{T};\theta), \phi(\mathcal{S};\theta))]. \tag{2}$$

In parameterization methods for dataset distillation, the synthetic dataset $\mathcal{S}$ is stored in more efficient representations consisting of bases $\mathcal{B} \in \mathbb{R}^{N\times C\times H\times W}$ and a set of transformation functions $F : \mathcal{B} \rightarrow \mathcal{S}$, which generate the synthetic dataset. Here, N denotes the number of bases, C represents the channel, and H and W are the height and width of the bases, respectively. In this paper, we propose a color transformation that reduces the number of unique colors in the image bases while preserving the essential details after color reduction for model training.

3.2 Overview

In this paper, we explore a new dimension of parameterization-based dataset distillation, concentrating on optimizing storage efficiency by minimizing color space redundancy. We argue that complex color representation within storage-sensitive distilled images is not crucial for training networks. Instead, the limited storage budget for distilled images should be allocated to novel samples that exhibit diverse object structures. The overall framework of AutoPalette is illustrated in Figure 1.

The proposed AutoPalette framework for dataset distillation consists of two components, including a palette network and a color-guided initialization strategy. The palette network is designed to enhance the color utility of the synthetic images in reduced color space by generating pixel-level color mappings. Accordingly, the original images $\tilde{x} \in \mathbb{R}^{C\times H\times W} \in \mathcal{S}$ are transformed into color-condensed images $\mathbf{b} \in \mathbb{R}^{C\times H\times W} \in \mathcal{B}$ with a reduced color spectrum, where we denote the process as $\phi_{palette} : \mathcal{S} \rightarrow \mathcal{B}$. RGB image dataset generally contains 256 distinct colors for each channel, and the palette network aims to reduce the number of colors by generating a color palette containing only K colors. As such, the synthetic dataset can be stored in a low-bit format, rather than the conventional 8-bit format. To better leverage the diverse color feature information within the original dataset, we equip the proposed framework with a novel initialization strategy, which employs the generalized graph cut function to select representative images for initialization. Our method dynamically evaluates the impact of real images based on their color structures and initializes the synthetic dataset with the images that yield the highest information gains of graph cut functions. The subsequent sections will illustrate each module in detail.

3.3 Color Reduction via Palette Network.

The core of our proposed parameterization method for dataset distillation is to condense the number of unique colors in an image so that the image can be stored with a more efficient manner. One of the key challenges of reducing the unique color number in dataset distillation lies in local discriminative feature preservation. When an image is represented with a smaller range of color, it is inevitable for some pixels to be merged to their neighbour color blocks. In this case, some of the local discriminative features (*e.g.*, edge, shape, *etc.*) can be erased or distorted in the color-reduced images, hindering the network trained on them to achieve the optimal performance.

To alleviate this issue, we design a simple yet effective network, namely palette network, which learns the pixel-level color allocation in the pruned color space with the discriminative feature maximally preserved. Particularly, the color palette network predicts the probability map $\mathbf{m} \in \mathbb{R}^{C\times H\times W\times K}$, which indicates the probability of a pixel being allocated to a color of the K-dimensional reduced color space by forwarding an image $\tilde{x}$ to the palette network:

$$\mathbf{m} = \phi_{color}(\tilde{x}; \theta_c), \tag{3}$$

where θ_c is the parameters of the palette network.

Given a base image $\mathbf{b}$ and its corresponding probability map $\mathbf{m}$, we formulate the palette $\tilde{\mathbf{m}} \in \mathbb{R}^{C\times K}$ as the average pixel values of all pixels assigned to the same color buckets index:

$$\tilde{\mathbf{m}}_{c,k} = \frac{\sum_{c,i,j} \tilde{x}_{c,i,j} \cdot \delta^{\mathbf{m}}_{c,i,j}(k)}{\sum_{c,i,j} \delta^{\mathbf{m}}_{c,i,j}(k)}, \tag{4}$$

where c and k denote the c-th channel for the k-th quantized color, i and j denote the vertical and horizontal pixel position in an image, and $\delta^{\mathbf{m}}$ is the Kronecker delta function, which equals 1 if $\arg\max \mathbf{m}_{c,i,j}$ equals k.

Once the color palette $\tilde{\mathbf{m}}$ and a probability map $\mathbf{m}$ are generated for an image, its color condensed image $\mathbf{b}$, with the number of unique colors per channel reduced to K, can be generated by an index searching process:

$$\mathbf{b}_{c,i,j} = \tilde{\mathbf{m}}[c, \mathbf{h}], \text{ where } \mathbf{h} = \arg\max_{k} \mathbf{m}_{c,i,j}. \tag{5}$$

To learn the palette network, a straightforward solution is to optimize the distillation task loss during the training stage. However, we can observe from Fig. 2 that solely relying on the task loss for training the palette network leads to most of the palette being inactivated. The network tends to assign pixels to a limited number of color buckets, which strongly limits the capacity and expressiveness of the distilled images. Therefore, two additional losses, namely **maximum color loss** and **palette balance loss**, are incorporated to enhance the utility of color buckets in synthetic images. In particular, maximum color loss, $\mathcal{L}_m$ encourages the palette network to generate color allocation such that each color bucket is at least filled with one pixel within color palette. By aggregating the maximum confidences from probability index maps across the spatial dimensions, we define the maximum color loss as:

$$\mathcal{L}_m = -\frac{1}{CK} \sum_{c=1}^{C} \sum_{k=1}^{K} \max_{(h,w)} (\mathbf{m}_{c,h,w,k}), \tag{6}$$

While the maximum color loss ensures the activation of each color bucket in the palette, the distribution of pixel numbers in color buckets can still be extremely imbalanced. Therefore, our framework leverages a palette balance loss $\mathcal{L}_b$, which encourages a more balanced usage of the buckets within the color palette by promoting color-wise entropy. We formulate the palette balance loss as the entropy of $\mathbf{m}$ over the spatial dimensions:

$$\mathcal{L}_b = \frac{1}{CK} \sum_{c=1}^{C} \sum_{k=1}^{K} P(\frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbf{m}_{c,i,j,k}) \log P(\frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbf{m}_{c,i,j,k}), \tag{7}$$

where $P(\cdot)$ represents the softmax function of $\mathbf{m}$ over the spatial dimensions.

By integrating the palette network with the complementary losses, we obtain the color condensed images while preserving the informative features.

3.4 Color Guided Initialization Module

The empirical results of previous studies have shown a strong correlation [3, 46] between the original images selected during initialization and the resulting distilled images in terms of visual appearance. In light of this finding, we propose an initialization method aimed at solving the redundancy problem in color-condensed synthetic images. However, since we do not have access to the optimized palette network, it is prohibitive to directly measure the information overlap within a class after color condensation. To mitigate this issue, we propose to leverage the traditional color quantization approach [15] to approximate the output of the palette network. Here, we denote the quantized full dataset as $\mathcal{T}^Q$. Our proposed initialization strategy leverages conditional gain within submodular information theoretics to identify the most diverse images of each class after color condensation. Specifically, the conditional gain $G(\mathcal{A}|\mathcal{C})$ implies the gain of information by adding set $\mathcal{C}$ to set $\mathcal{A}$, where $\mathcal{A}, \mathcal{C} \subset \mathcal{T}^Q$ and $\mathcal{A} \cap \mathcal{C} = \emptyset$. Formally, we have:

$$G(\mathcal{A}|\mathcal{C}) = G(\mathcal{T}^Q) - G(\mathcal{C}), \tag{8}$$

where $G(\cdot)$ denotes a submodular function. Submodular information functions [18] describe a set of combinatorial functions that satisfy the Shannon inequality [34, 26] and can effectively model the diversity of a subset. In our implementation, we adopt a monotone submodular function, namely generalized graph cut [2], which maximizes the similarities between samples in $\mathcal{A}$ and $\mathcal{C}$ and minimizes and dissimilarities among the samples in $\mathcal{A}$. The generalized graph cut function $G^*(\cdot)$ is defined as follows:

$$G^*(\mathcal{A}|\mathcal{C}) = \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{A}} Sim(i, j) - \sum_{j_1, j_2 \in \mathcal{A}} Sim(j_1, j_2), \tag{9}$$

https://doi.org/10.52202/079017-1687

where i and j denote data samples from the sets $\mathcal{C}$ and $\mathcal{A}$, respectively. $Sim(\cdot,\cdot)$ is a similarity function between two samples. Instead of directly measuring the feature level similarity, we propose to measure the similarity between the last layer gradients $\nabla\theta$ as follows:

$$Sim(i,j) = cos(\nabla_\theta \mathcal{L}_{CE}(Q(\tilde{x}^i),\theta), \nabla_\theta \mathcal{L}_{CE}(Q(\tilde{x}^j),\theta)), \tag{10}$$

where $cos(\cdot,\cdot)$ is the cosine similarity function, Q denotes the Median Cut quantization method, $\tilde{x}^i$ and $\tilde{x}^j$ are the ith and jth samples of set $\mathcal{T}$, and ∇_θ is the gradient of cross-entropy loss between the prediction and the ground truth label on the last layer of the network. By substituting Eq.(9) into Eq. (8), we select a representative sample for inclusion in $\mathcal{A}$ by:

$$\arg\max_c G^*(\mathcal{A}) - 2\sum_{i\in\mathcal{A}}\sum_{c\in\mathcal{C}} Sim(i,c). \tag{11}$$

The proof for the graph cut conditional gain is provided in Appendix A.1. From Eq. (11), we can see that the data sample obtaining the highest conditional gain may be selected. Intuitively, we select the most representative sample from the unselected set $\mathcal{C}$, whilst ensuring it is dissimilar to the already selected samples in $\mathcal{A}$. The entire color diversity selection process is provided in Algorithm 1.

By far, our initialization method can select diverse and representative samples of each class in the approximated quantization set. To minimize the difference between the approximation set and the output of the palette network, we put forward a regularization term $\mathcal{L}_a$. The regularization term not only constrains the color allocation shifting of palette network, but also enhances allocation consistency, so similar colors are grouped together with higher fidelity. The regularization term $\mathcal{L}_a$ is defined as:

$$\mathcal{L}_a = \frac{1}{N}\|\mathbf{h}\odot\mathbf{h}^\intercal - \mathbf{h}'\odot\mathbf{h}'^\intercal\|_2^2, \tag{12}$$

where $\mathbf{h}'$ denotes the the $\arg\max$ of the color mapping indices by Median Cut over the color space, and $\odot$ denotes the element-wise multiplication resulting in a self correlation matrix for palette bucket allocations of the palette network and Median Cut. When creating index mappings for pixels, different methods might cluster the same pixels into the same group, but the indices may not match. By utilizing $\mathcal{L}_a$, we emphasize clustering resemblance, disregarding the order of clustering indices.

3.5 Overall Dataset Distillation Objective

Our framework aims to create a color-condensed synthetic version of the original dataset while maximally preserving task-related information. In line with other parameterization-based methods, we incorporate the dataset distillation loss $\mathcal{L}_{task}$ from existing works into our framework.

As such, to update the palette network, we have the overall loss function defined as:

$$\arg\min_{\theta_c} \mathcal{L}_{palette} = \mathcal{L}_{task} + \alpha\mathcal{L}_m + \beta\mathcal{L}_b + \gamma\mathcal{L}_a, \tag{13}$$

where α, β, γ are the coefficients as the weights of loss components. The synthetic set $\mathcal{S}$ is optimized as follows:

$$\arg\min_{\mathcal{S}} \mathcal{L}_{task} = \mathcal{L}(\phi(\mathcal{T};\theta), \phi(\mathcal{B};\theta)), \text{where } \mathcal{B} = \phi_{palette}(\mathcal{S};\theta_c), \tag{14}$$

where $\phi_{palette}(\cdot;\theta_c)$ denotes the color quantization process using the palette network.

3.6 Storage Analysis

In our experiments, the images follow the 256-color storage convention, where pixel values occupy 8-bit storage space. Given the storage budget of images per class (IPC), the maximum storage budget for one class is capped at $8 \times \text{IPC} \times CHW$, where C, H and W represent the channel, height and width of the images, respectively. When representing a colorful image pixel value with n bits, where $1 \leq n < 8$, there can be at most 2^n distinct colors per image. This must satisfy the condition $\sum_{i=1}^{2^{8-n}} N_i \leq 2^8$, where N_i is the number of colors for the i-th color reduced image and each $N_i \leq 2^n$. Therefore, for images with n-bit format, up to 2^{8-n} colors can be represented in the storage budget using a bitmap index with small bytes. The bitmap index indicates the image number associated with the current lower bit color value.

Table 1: Test accuracy (%) of previous works and our method on ConvNet D3. Our synthetic images are reduced from 256 colors to 64 colors. Our method outperforms previous methods and achieves state-of-the-art performance.

Dataset		CIFAR10			CIFAR100		
IPC		1	10	50	1	10	50
Coreset	Random	14.4±0.2	26.0±1.2	43.4±1.0	4.2±0.3	14.6±0.5	30.0±0.4
	Herding [42]	21.5±1.3	31.6±0.7	40.4±0.6	8.4±0.3	17.3±0.3	33.7±0.5
	K-Center [33]	23.3±0.9	36.4±0.6	48.7±0.3	8.6±0.3	20.7±0.2	33.6±0.4
Distillation	DD [40]	-	36.8±1.2	-	-	-	-
	DM [46]	26.0±0.8	48.9±0.6	63.0±0.4	11.4±0.3	29.7±0.3	43.6±0.4
	DC [47]	28.3±0.5	44.9±0.5	53.9±0.5	12.8±0.3	25.2±0.3	-
	TM [3]	46.3±0.8	65.3±0.7	71.6±0.2	24.3±0.3	40.1±0.4	47.7±0.2
	DATM [12]	46.9±0.5	66.8±0.2	76.1±0.3	27.9±0.2	47.2±0.4	**55.0**±0.2
Parameterization	IDC [20]	50.0±0.4	67.5±0.5	74.5±0.1	-	-	-
	HaBa [24]	48.3±0.8	48.3±0.8	48.3±0.8	33.4±0.4	40.2±0.2	47.0±0.2
	RTP [9]	**66.4**±0.4	71.2±0.4	73.6±0.5	34.4±0.4	42.9±0.7	-
	SPEED [41]	63.2±0.1	73.5±0.2	77.7±0.4	**40.0**±0.4	45.9±0.3	49.1±0.2
	FReD [35]	60.6±0.8	70.3±0.3	75.8±0.1	34.6±0.4	42.7±0.2	47.8±0.1
	AutoPalette	58.6±1.1	**74.3**±0.2	**79.4**±0.2	38.0±0.1	**52.6**±0.3	53.3±0.8

4 Experiments

In this section, we first evaluate the effectiveness of our method in comparison with other parameterization methods on various datasets. Afterwards, we perform experiments on the relations between synthetic image color number and model performance. We also conduct ablation studies and assess the efficacy of each proposed component to distillation performance.

4.1 Experimental Setting

We conduct experiments of our model on various benchmark datasets, including CIFAR-10 [21], CIFAR-100 [21] and ImageNet [7]. We compare our parameterization method with core-set methods and other existing DD works containing baselines such as DD [40], DM [46], DC [47], TM [3], and parameterization techniques including IDC [20], HaBa [24], RTP [9], SPEED [41], FReD [35]. Experiments are performed on different distillation memory budget settings for 1/10/50 images per class (IPC). We follow the previous works to use a ConvNetD3 for the CIFAR family and ConvNetD5 for ImageNet as the training and evaluation network. We follow the DATM [12] implementation based on trajectory matching, without soft label initialization using correctly predicted samples. Each experiment is evaluated on 5 randomly initialized networks, and the mean and standard deviation of the evaluation accuracy are recorded. We set loss coefficients α=1, β=1, γ=3 for all experiments if not specified. All experiments can be conducted on 2×Nvidia H100 GPUs that have 80GB RAM for each or 4×Nvidia V100 GPUs that have 32GB RAM for each.

4.2 Experimental Results

Results on CIFAR10 and CIFAR100. We perform experiments under paramterization settings on CIFAR10 [21], CIFAR100 [21]. We set color palette network to condense the number of colors of a single image from 256 to 64, such that despite huge color space reduction quantized images still preserve much fidelity of images. As shown in Table 1, our method achieves superior performance than other parameterization works in various tasks. Notably, in the experiments when IPC equals 10 and 50, our method significantly outperforms other methods. In CIFAR100 experiments, our model achieves 52.6% and 53.3% classification accuracy when IPC is respectively 10 and 50, which increases 6.7% and 4.2% higher than previous state-of-the-art parameterization methods. These outstanding performances highlight that reducing the color redundancy within the synthetic dataset can improve the storage utility and thereby improve the distillation result.

Results on ImageNet. Following [3], we conduct experiments on six subsets of ImageNet, where each subset consists of 10 classes and the images are of resolution 128×128. We conduct experiments with the storage budget of IPC=10. ConvNetD5 is employed as the backbone model for training

https://doi.org/10.52202/079017-1687

Table 2: Test accuracy (%) on ImageNet-Subset: ImageNette, ImageWoof, ImageFruit, ImageMeow, ImageSquawk, ImageYellow. All experiments are conducted on CIFAR10 with IPC=10 storage budget for parameterization methods.

Dataset	ImageNette	ImageWoof	ImageFruit	ImageMeow	ImageSquawk	ImageYellow
TM [3]	63.0±1.3	35.8±1.8	40.3±1.3	40.4±2.2	52.3±1.0	60.0±1.5
HaBa [24]	64.7±1.6	38.6±1.3	42.5±1.6	42.9±0.9	56.8±1.0	63.0±1.6
FrePo [35]	66.5±0.8	42.2±0.9	-	-	-	-
SPEED [41]	72.9±1.5	44.1±1.4	**50.0**±0.8	52.0±1.3	**71.8**±1.3	70.5±1.5
AutoPalette	**73.2**±0.6	**44.3**±0.9	48.4±1.8	**53.6**±0.7	68.0±1.4	**72.0**±1.6

and evaluation. From Table 2, we can see our method outperforms other PDD methods on most of ImageNet subsets, including ImageNette, ImageWoof, ImageMeow and ImageYellow, while results of the other subsets still achieve comparable performance with previous state-of-the-art results. Specifically, our method achieves 44.3% and 72.0% on hard datasets ImageWoof and ImageYellow, increasing 0.2% and 1.5% than previous best PDD methods. We also observe that for subsets with distinct classes, our method achieves promising results, which is because color condensation effectively preserves the key semantics necessary for accurate classification. On the other hand, for fine-grained subsets where the classes are similar, we observe inferior performance. This is likely because fine-grained details are blurred in images represented by fewer colors, thereby making it challenging to differentiate between classes.

Compatibility of Distillation Frameworks. While we take trajectory matching as our primary distillation method, we demonstrate that our framework can effortlessly be equipped to improve other dataset distillation methods. As illustrated in Table 5, our method shows a significant performance boost across all IPC settings and datasets when adopted to the distribution matching method. Especially, our method increases the test accuracy up to 15% when IPC=10, and 15.7% for IPC=1 on CIFAR100. This observation aligns with our objectives that our color-oriented redundancy management framework should be adapted across different standard DD frameworks The performance improvement underscores the high compatibility of our methods with diverse DD frameworks.

4.3 Ablation Study

Here, we focus on comparing variants of our proposed framework. Therefore, we fix the number of synthetic images to 10 per class, rather than fully utilizing the available storage capacity.

Effectiveness of Loss Components. To validate the contribution of each loss term to the overall framework, we conduct experiments on CIFAR10 with IPC=10. Specifically, we construct three variants of our model by removing $\mathcal{L}_m$, $\mathcal{L}_b$, and $\mathcal{L}_a$, correspondingly. We show the experimental result in Table 3. Under the same experimental conditions, eliminating specific loss functions will suffer from performance decline. The results demonstrate the essential role played by each loss function in optimizing the palette network.

Effectiveness of Selection Criteria in the Color-guided Initialization. We compare our proposed initialization with two baseline sample selection criteria, including Random Real, and Graph Cut Real. Random real is widely adopted by DD methods, which randomly select images from the full dataset as the initialization of the synthetic dataset. In Graph Cut Real, we apply graph cut on 8-bit images

Table 3: Test accuracy (%) when a certain loss component is removed during training.

$\mathcal{L}_m$	$\mathcal{L}_b$	$\mathcal{L}_a$	Accuracy
✗			64.00
	✗		61.40
		✗	60.14
✓	✓	✓	**66.20**

Table 4: Evaluation on the effectiveness of submodular selection using quantized images, in comparison with random initialization and submodular selection using full color images.

Initialization Method	Accuracy
Random Real	60.84
Graph Cut Real	61.41
Graph Cut on Quantized Image	**62.13**

Table 5: Test accuracy (%) on different DD frameworks including DM and TM across various IPC settings. We adopt ConvNetD3 as the backbone network to distil the synthetic dataset on CIFAR10 and CIFAR100.

Framework		DM [46]			TM [3]		
Dataset	IPC	Vanilla	AutoPalette	Increase	Vanilla	AutoPalette	Increase
CIFAR10	1	26.0±0.8	35.5±0.4	9.5↑	46.3±0.8	58.6±1.1	12.3↑
	10	48.9±0.6	60.9±0.1	12.0↑	65.3±0.7	74.3±0.2	9.0↑
	50	63.0±0.4	71.6±0.4	8.6↑	71.6±0.2	79.4±0.2	7.8↑
CIFAR100	1	11.4±0.3	20.9±0.1	9.5↑	24.3±0.3	38.0±0.1	15.7↑
	10	29.7±0.3	44.7±0.1	15.0↑	40.1±0.4	52.6±0.3	12.5↑
	50	43.6±0.4	50.1±0.1	6.5↑	47.7±0.2	54.1±0.8	5.6↑

and select the most representative samples with high information gain. Compared with Graph Cut Real, the selection criteria used in our framework computes the information gain over the quantized images. From table 4, we can see that our methods using quantized images exhibit better performance than comparison approaches. The graph cut with original full-color images also outperforms the baseline model using randomly selected real images as initialization, confirming the effectiveness of computing information gain over color-reduced images

Effectiveness of Color-guided Initialization under Different Color Depth. Our experiments compare the performance of two ways to initialize the base images: one employing our method of color-guided initialization (denotes GraphCut), and randomly selecting real images as the base images (denotes Baseline). We contrast two approaches on CIFAR10 with IPC=10, spanning from synthesizing images with low-bit quantization to those with higher-bit quantization. From Figure 3, we observe that our method brings better performance when quantized images are represented in lower bits. Starting from 16 colors per pixel, we observe a gradual convergence in performance as we move towards utilizing full-color space. Meanwhile, as can be seen, when K=32 and 64, it achieves the best trade-off, as it still attains competitive performance comparable to that of full-color budget images, while substantially reducing the storage space required for quantized images.

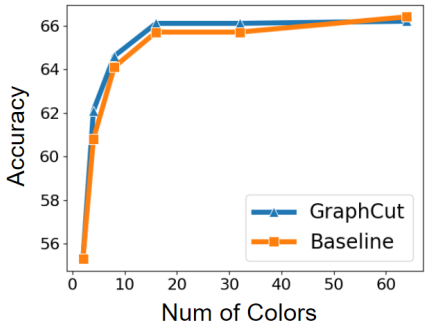

Figure 3: Comparison between the performance of submodular color diversity initialization and random real images initialization.

5 Conclusions, Limitations, and Future Work

In this paper, we aim to solve the color-redundancy issue within data distillation from both the image level and the dataset level. For condensing the colors within images, we utilize a palette network to capture the color redundancy between pixels and represent images using fewer colors. Beyond this, to reduce repetitive patterns in between synthetic images design a guided image initialization module that selects samples by maximising information gain. Extensive experimental results demonstrate that our AutoPalette framework can effectively reduce color redundancy and simultaneously preserve the essential low-level feature for model training.

Limitations. For instance, images from different classes may have a bias towards color usage. Images of one class may be sufficient to be represented by fewer colors than those from other classes, in which case an imbalanced color budget arrangement may be a better option. In future, it is also promising to explore the dynamic color depth allocation, which allocates more budgets to difficult classes, thereby improving the distillation performance.

Acknowledgments and Disclosure of Funding

This research is partially supported by the Australian Research Council (DE240100105, DP240101814, DP230101196)

https://doi.org/10.52202/079017-1687

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[2] Yuri Boykov and Olga Veksler. Graph cuts in vision and graphics: Theories and applications. In *Handbook of mathematical models in computer vision*, pages 79–96. Springer, 2006.

[3] 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.

[4] 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 (CVPR)*, pages 3739–3748, 2023.

[5] 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.

[6] 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.

[7] 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.

[8] Yining Deng, Charles Kenney, Michael S Moore, and BS Manjunath. Peer group filtering and perceptual color image quantization. In *1999 IEEE International Symposium on Circuits and Systems (ISCAS)*, volume 4, pages 21–24. IEEE, 1999.

[9] 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.

[10] 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.

[11] Michael Gervautz and Werner Purgathofer. A simple method for color quantization: Octree quantization. In *New Trends in Computer Graphics: Proceedings of CG International'88*, pages 219–231. Springer, 1988.

[12] 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.

[13] Jialing Han. An adaptive grayscale watermarking method in spatial domain. *Journal of Information and Computational Science*, 12:4759–4769, 08 2015.

[14] 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.

[15] Paul Heckbert. Color image quantization for frame buffer display. *ACM Siggraph Computer Graphics*, 16(3):297–307, 1982.

[16] Yunzhong Hou, Liang Zheng, and Stephen Gould. Learning to structure an image with few colors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10116–10125, 2020.

[17] Yunzhong Hou, Liang Zheng, and Stephen Gould. Learning to structure an image with few colors and beyond, 2022.

[18] Rishabh Iyer, Ninad Khargoankar, Jeff Bilmes, and Himanshu Asanani. Submodular combinatorial information measures with applications in machine learning. In *Algorithmic Learning Theory*, pages 722–754. PMLR, 2021.

[19] Simonyan Karen. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv: 1409.1556*, 2014.

[20] 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.

[21] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[23] Hae Beom Lee, Dong Bok Lee, and Sung Ju Hwang. Dataset condensation with latent space knowledge factorization and sharing. *arXiv preprint arXiv:2208.10494*, 2022.

[24] 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.

[25] Yanqing Liu, Jianyang Gu, Kai Wang, Zheng Zhu, Wei Jiang, and Yang You. Dream: Efficient dataset distillation by representative matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17314–17324, 2023.

[26] William McGill. Multivariate information transmission. *Transactions of the IRE Professional Group on Information Theory*, 4(4):93–111, 1954.

[27] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Practical full resolution learned lossless image compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10629–10638, 2019.

[28] Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. *arXiv preprint arXiv:2011.00050*, 2020.

[29] Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. Dataset distillation with infinitely wide convolutional networks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 5186–5198, 2021.

[30] 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.

[31] Michael T Orchard, Charles A Bouman, et al. Color quantization of images. *IEEE transactions on signal processing*, 39(12):2677–2690, 1991.

[32] Ahmad Sajedi, Samir Khaki, Ehsan Amjadian, Lucy Z Liu, Yuri A Lawryshyn, and Konstantinos N Plataniotis. Datadam: Efficient dataset distillation with attention matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17097–17107, 2023.

[33] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

[34] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001.

[35] Donghyeok Shin, Seungjae Shin, and Il-Chul Moon. Frequency domain-based dataset distillation. *Advances in Neural Information Processing Systems*, 36, 2024.

[36] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016.

 https://doi.org/10.52202/079017-1687

[37] Paul Vicol, Jonathan P Lorraine, Fabian Pedregosa, David Duvenaud, and Roger B Grosse. On implicit bias in overparameterized bilevel optimization. In *International Conference on Machine Learning*, pages 22234–22259. PMLR, 2022.

[38] Kai Wang, Jianyang Gu, Daquan Zhou, Zheng Zhu, Wei Jiang, and Yang You. Dim: Distilling dataset into generative model. *arXiv preprint arXiv:2303.04707*, 2023.

[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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12196–12205, 2022.

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

[41] Xing Wei, Anjia Cao, Funing Yang, and Zhiheng Ma. Sparse parameterization for epitomic dataset distillation. *Advances in Neural Information Processing Systems*, 36, 2024.

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

[43] Xiaolin Wu. Color quantization by dynamic programming and principal analysis. *ACM Transactions on Graphics (TOG)*, 11(4):348–372, 1992.

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

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

[46] 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.

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

[48] Ganlong Zhao, Guanbin Li, Yipeng Qin, and Yizhou Yu. Improved distribution matching for dataset condensation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7856–7865, 2023.

[49] Yongchao Zhou, Ehsan Nezhadarya, and Jimmy Ba. Dataset distillation using neural feature regression. *Advances in Neural Information Processing Systems*, 35:9813–9827, 2022.

A Appendix

A.1 Proof for Conditional Information Gain of Graph Cut

The generalized graph cut set function is defined as:

$$f(\mathcal{A}) = \lambda \sum_{i \in \mathcal{T}} \sum_{a \in \mathcal{A}} Sim(i, a) - \sum_{a_1, a_2 \in \mathcal{A}} Sim(a_1, a_2), \tag{15}$$

where $\mathcal{A} \subset \mathcal{T}$, and Sim is a similarity function. The submodular conditional gain is defined as:

$$f(\mathcal{A}|\mathcal{B}) \triangleq f(\mathcal{A} \cup \mathcal{B}) - f(\mathcal{B}), \tag{16}$$

which indicates the gain by adding samples in set $\mathcal{B}$ to $\mathcal{A}$. By substituting Eq. (16) with (15), we can obtain:

$$\begin{aligned} f(\mathcal{A}|\mathcal{B}) =& \lambda \sum_{i \in \mathcal{T}} \sum_{c \in \mathcal{A} \cup \mathcal{B}} Sim(i, c) - \sum_{c_1, c_2 \in \mathcal{A} \cup \mathcal{B}} Sim(c_1, c_2) \\ & - \lambda \sum_{i \in \mathcal{T}} \sum_{b \in \mathcal{B}} Sim(i, b) + \sum_{b_1, b_2 \in \mathcal{B}} Sim(b_1, b_2). \end{aligned} \tag{17}$$

In the sample selection cases, $\mathcal{A}$ and $\mathcal{B}$ are disjoint, and thus $\sum_{c \in \mathcal{A} \cup \mathcal{B}}$ can be rewritten as $\sum_{c \in \mathcal{A} - \mathcal{B}}$. Therefore, we can reformulate $f(\mathcal{A} \cup \mathcal{B})$:

$$\begin{aligned} f(\mathcal{A} \cup \mathcal{B}) =& \lambda \sum_{i \in \mathcal{T}} \sum_{c \in \mathcal{A} \setminus \mathcal{B}} Sim(i, c) + \lambda \sum_{i \in \mathcal{T}} \sum_{b \in \mathcal{B}} Sim(i, b) \\ & - \sum_{b_1, b_2 \in \mathcal{B}} Sim(b_1, b_2) - \sum_{c_1, c_2 \in \mathcal{A} \setminus \mathcal{B}} Sim(c_1, c_2) \\ & - 2 \sum_{a' \in \mathcal{A} \setminus \mathcal{B}} \sum_{b \in \mathcal{B}} Sim(a', b). \end{aligned} \tag{18}$$

We can then formulate Eq. (16) as:

$$f(\mathcal{A}|\mathcal{B}) = f(\mathcal{A} \setminus \mathcal{B}) - 2 \sum_{a' \in \mathcal{A} \setminus \mathcal{B}} \sum_{b \in \mathcal{B}} Sim(a', b). \tag{19}$$

When $\mathcal{A}$ and $\mathcal{B}$ are disjoint, $\mathcal{A}$ is independent of $\mathcal{B}$ and we then simplify the conditional gain of graph cut function to:

$$f(\mathcal{A}|\mathcal{B}) = f(\mathcal{A}) - 2 \sum_{a' \in \mathcal{A} \setminus \mathcal{B}} \sum_{b \in \mathcal{B}} Sim(a', b). \tag{20}$$

A.2 Experimental Details

Datasets. We conduct experiments on multiple datasets:

- CIFAR10: an image dataset consists of 50,000 32×32 RGB images for training, and 10,000 images for testing. CIFAR10 contains 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck.
- CIFAR100: an image dataset that is similar to CIFAR100, but has 100 classes containing 600 images each.
- ImageNet subsets: high resolution image subsets of ImageNet [7] that contain all 128×128 RGB images and each contains 10 classes. ImageNet subsets include include ImageNette, ImageWoof, ImageFruit, ImageMeow, ImageSquawk, and ImageYellow.

Networks. For the experiments of low resolution datasets including CIFAR10 and CIFAR100, 3-layer convolutional neural networks (ConvNet) are employed and we follow the identical network structures to the previous works. Each convolution layer contains 128 3×3 filters, followed by an instance normalization layer, a ReLU, and an average pooling layer with 2×2 kernel and stride 2.

https://doi.org/10.52202/079017-1687

Algorithm 1: Algorithm for guided image selection with maximum information gain.

Input: Original Dataset $\mathcal{T}$; Number of classes N_{class}; Images per class *IPC*; Network θ
Output: Set of selected data samples $\mathcal{A}$ for each class

for $i \leftarrow 1$ **to** N_{class} **do**
  $X^i \leftarrow$ all images belong to class i from $\mathcal{T}$;
  $\mathcal{A}^i \leftarrow \{rand(X^i)\}$; `// Initialize the selected set with a random sample`
  for $j \leftarrow 2$ **to** *IPC* **do**
    $\mathcal{C} \leftarrow \mathcal{T}^i \setminus \mathcal{A}^i$;
    $c^* \leftarrow$ Eq. (11) ; `// Select the data sample with the highest gain`
    $\mathcal{A}^i \leftarrow \mathcal{A}^i \cup \{c^*\}$;

For the high resolution datasets such as ImageNet subsets, we use 5-layer ConvNets to perform the experiments. For the cross architecture experiments, we employ VGG11 [19], AlexNet [22], and RestNet18 [14] and follow the implementations of the previous DD works. The palette network consists of two convolution layers and one ReLU between them. Both two convolution layers have 1×1 kernels and the second layer has no bias.

Implementation details. While we primarily use Trajectory matching (TM) as the distillation objectives, our method can be seamlessly adapted into other DD frameworks for various downstream tasks and datasets. In Table 9 and 10, we provide the hyper-parameters settings in our work for both TM and DM on different datasets. Specifically, although our method is insensitive for most of the hyper-parameters, certain parameters including synthetic steps, the maximum starting epoch and the synthetic batch size should be carefully examined.

A.3 Algorithm for Sample Selection in Initialization

A.4 Cross Architecture Performance

One of the main concerns in data distillation arises from synthetic dataset overfitting to the training models, resulting in limited generalizability to be used by the other network architectures. Therefore, cross-architecture performance is crucial for evaluating the effectiveness of data distillation methods. To assess the generalized performance of our method, we employ CIFAR10 synthetic datasets trained on ConvNet to train various network structures including VGG11 [19], AlexNet [22] and ReNnet18 [14]. The results of cross-network architecture are presented in Table 6. We noticed that when IPC is relatively higher, our method outperforms the other baseline methods.

Table 6: Cross architecture performance of synthetic dataset that is optimized by ConvNet. Experiments are performed on CIFAR10, and VGG11, AlexNet and ResNet18 are used for evaluating the cross architecture performance.

	VGG11			AlexNet			ResNet18		
Method \ IPC	2	11	51	2	11	51	2	11	51
TM [3]	38.0 ± 1.2	50.5 ± 1.0	61.4 ± 0.3	26.1 ± 1.0	36.0 ± 1.5	49.2 ± 1.3	35.2 ± 1.0	45.1 ± 1.5	54.5 ± 1.0
IDC [20]	48.2 ± 1.2	52.7 ± 0.7	65.2 ± 0.6	32.5 ± 2.2	43.7 ± 3.0	54.9 ± 1.1	46.7 ± 0.9	50.2 ± 0.6	64.5 ± 1.2
HaBa [24]	48.3 ± 0.5	**60.5 ± 0.6**	67.5 ± 0.4	43.6 ± 1.5	49.0 ± 3.0	60.1 ± 1.4	47.4 ± 0.7	58.0 ± 0.9	64.4 ± 0.6
FReD [35]	**50.1 ± 0.8**	60.0 ± 0.6	69.9 ± 0.4	**44.1 ± 1.3**	**55.9** ± 0.8	65.9 ± 0.8	**53.9 ± 0.7**	64.4 ± 0.6	71.4 ± 0.7
AutoPalette	41.3 ± 1.1	57.6 ± 1.1	**70.3 ± 0.2**	36.7 ± 2.5	44.5 ± 1.1	**72.5 ± 0.2**	46.7 ± 1.2	**66.0 ± 1.3**	**75.8 ± 0.2**

A.5 Number of colors vs Performance

Our method still demonstrates competitive performance, even when synthetic images are quantized into extremely low bits, as shown in Table 7. The test results for CIFAR10 IPC=10 is provided, from which we can observe a slight performance gap when the number of colors used for quantized images decreases to merely 8 colors per pixel (3 bits) from the original 256 colors per pixel (8 bits). The results demonstrate that our color quantization method captures key features for synthetic data and appropriately condenses images into fewer colors.

Table 7: Model performance when evaluated on the synthetic dataset using at most different number of colors.

#colors	256 (full)	64	32	16	8	4	2
Accuracy	66.8	66.2	66.1	65.9	64.6	62.1	55.3

Table 8: Test accuracy (%) in comparison with the traditional color quantization methods. Quantization methods are applied to synthetic dataset for CIFAR10, when IPC=10 and 50.

IPC	10					
Scale	2	4	8	16	32	64
Median Cut	42.5	46.4	52.4	55.6	57.6	60.7
OCTree	23.7	30.7	41	53.1	59.6	60.8
Palette Network	**55.3**	**62.1**	**64.6**	**65.9**	**66.1**	**66.2**

Table 9: Hyperparameters for our method based on Distribution matching (DM) framework.

Dataset	IPC	Synthetic batch size	Palette network LR	Synthetic Image LR	ZCA
CIFAR10	1	-	0.05	1	True
	10	-	0.05	1	True
	50	-	0.05	10	True
CIFAR100	1	-	0.05	1	True
	10	-	0.05	1	True
	50	50	0.05	10	True

A.6 Traditional color Quantization Methods

Our palette network aims to identify the essential color characteristics of the synthetic dataset and represent it with fewer colors. To test the efficacy of the palette network, we compare it with other color quantization methods on DD tasks, including Median Cut [15] and OCTree [11]. We conduct the experiments on CIFAR10, and Table 8 shows that the palette network achieves the superior performances under different IPC settings.

Table 10: Hyperparameters for our method based on Trajectory matching (TM) framework.

Dataset	IPC	Synthetic batch size	Synthetic steps	Expert epochs	Max start epoch	Palette network LR	Synthetic Image LR	Step size LR	Teacher LR	ZCA
CIFAR10	1	-	80	2	15	0.05	500	10^{-7}	10^{-2}	True
	10	-	35	2	40	0.05	1000	10^{-5}	10^{-2}	True
	50	600	40	2	50	0.05	500	10^{-5}	10^{-2}	True
CIFAR100	1	-	60	2	20	0.05	1000	10^{-5}	10^{-2}	True
	10	600	35	2	70	0.05	1000	10^{-5}	10^{-2}	True
	50	200	60	2	70	0.05	1000	10^{-5}	10^{-2}	True
ImageNette	10	80	20	2	20	0.01	10000	10^{-4}	10^{-2}	False

https://doi.org/10.52202/079017-1687

A.7 Visualization of Distilled Images

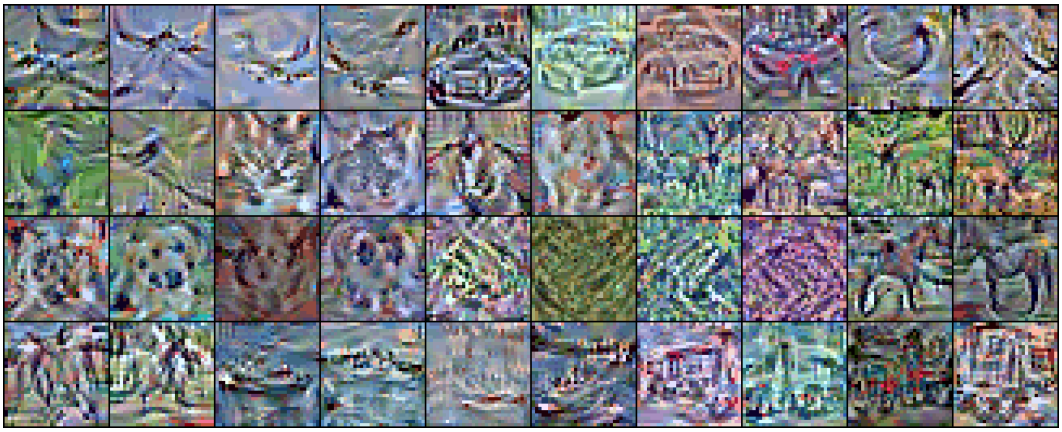

Figure 4: CIFAR10 color condensed synthetic images with ZCA whitening.

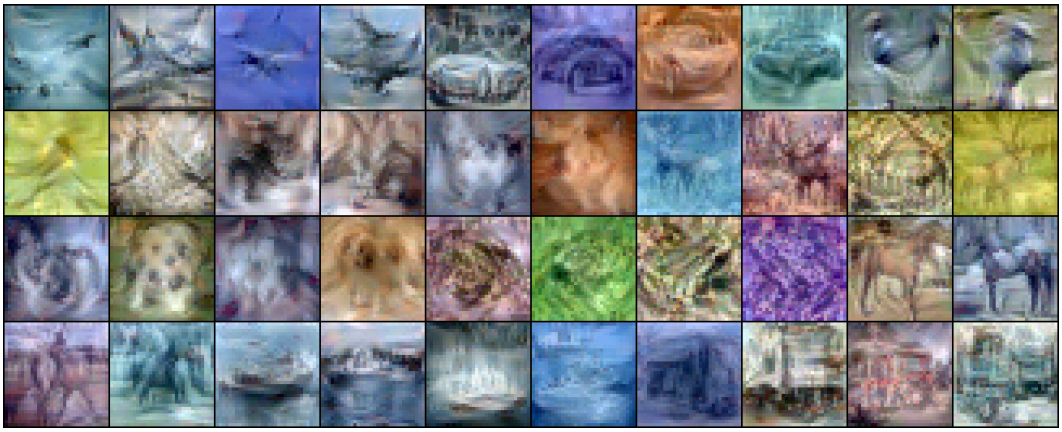

Figure 5: CIFAR10 color condensed synthetic images without ZCA whitening.

Figure 6: CIFAR10 synthetic images in 3-bit color depth

https://doi.org/10.52202/079017-1687

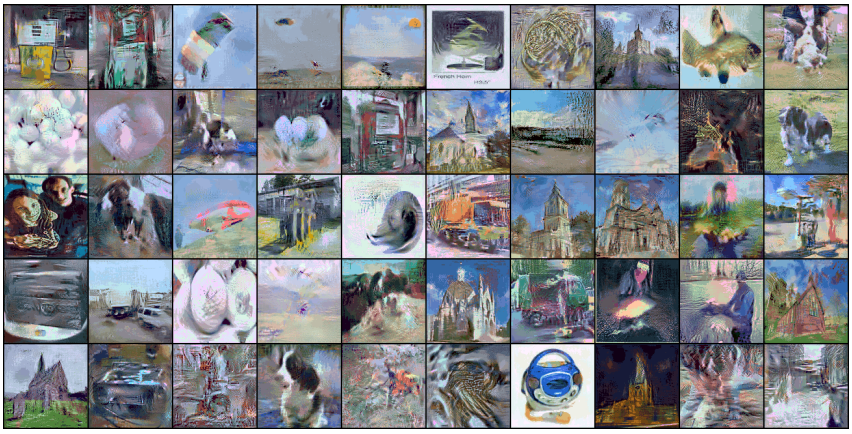

Figure 7: Color condensed synthetic images for ImageNette

Figure 8: Color condensed synthetic images for ImageWoof

Figure 9: Color condensed synthetic images for ImageFruit

https://doi.org/10.52202/079017-1687

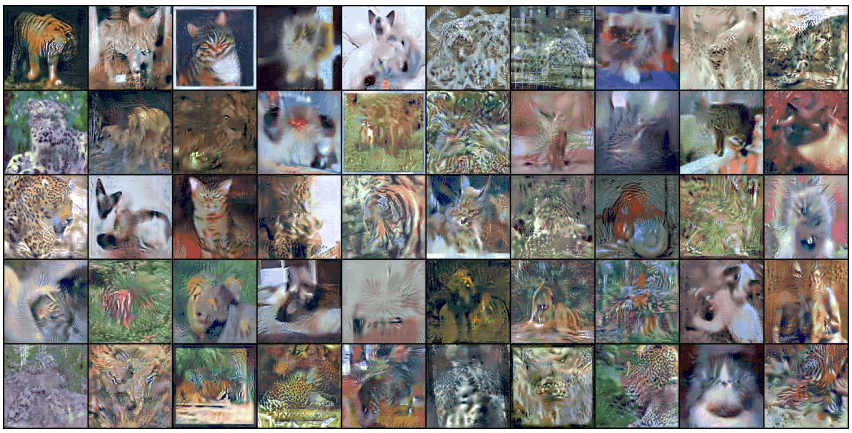

Figure 10: Color condensed synthetic images for ImageMeow

Figure 11: Color condensed synthetic images for ImageSquawk

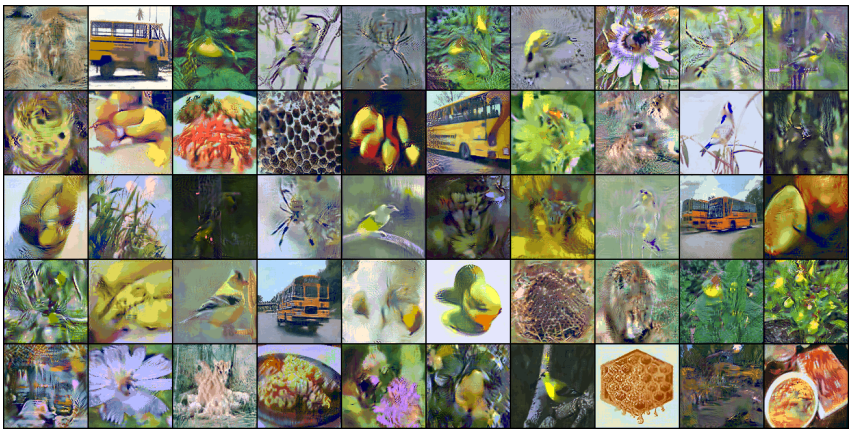

Figure 12: Color condensed synthetic images for ImageYellow

https://doi.org/10.52202/079017-1687

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main contributions of the work are clearly stated in the abstract and introductions, which is about exploring data distillation from color perspectives.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in the last section of the main paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

https://doi.org/10.52202/079017-1687

Justification: All assumptions and proof are clarified in the paper. For example, the proof of the conditional submodular gain function.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The experiment environments and hyper-parameters are included in the experiment section and appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All the datasets are open sourced, and the code is provided through the link in abstract.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: All experiment details are specified, and the choices of hyper-paramters are discussed.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Error bars are clearly reported. All experiments are conducted for multiple times and the mean accuracy and standard deviation are given.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

https://doi.org/10.52202/079017-1687

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Computation resources are included in the experiment sections.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work follows NeurIPS code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our work has no societal impacts, since it is not related to any social activities.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper has no such risks, since the data and models are all

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the existing works are properly referenced.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

https://doi.org/10.52202/079017-1687

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: New framework is well presented.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA] .

    Justification: No crowdsourcing is involved in this work.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA] .

    Justification: No human participant is involved in this work.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.