Fetch and Forge: Efficient Dataset Condensation for Object Detection

Ding Qi[1*] **Jian Li**[2†] **Jinlong Peng**[2] **Bo Zhao**[4]
Shuguang Dou[1] **Jialin Li**[2] **Jiangning Zhang**[2]
Yabiao Wang[3,2†] **Chengjie Wang**[2] **Cairong Zhao**[1†]
[1]Tongji University [2]Tencent YouTu Lab
[3]Zhejiang University [4]Shanghai Jiao Tong University

Abstract

Dataset condensation (DC) is an emerging technique capable of creating compact synthetic datasets from large originals while maintaining considerable performance. It is crucial for accelerating network training and reducing data storage requirements. However, current research on DC mainly focuses on image classification, with less exploration of object detection. This is primarily due to two challenges: (i) the multitasking nature of object detection complicates the condensation process, and (ii) Object detection datasets are characterized by large-scale and high-resolution data, which are difficult for existing DC methods to handle. As a remedy, we propose DCOD, the first dataset condensation framework for object detection. It operates in two stages: Fetch and Forge, initially storing key localization and classification information into model parameters, and then reconstructing synthetic images via model inversion. For the complex of multiple objects in an image, we propose Foreground Background Decoupling to centrally update the foreground of multiple instances and Incremental PatchExpand to further enhance the diversity of foregrounds. Extensive experiments on various detection datasets demonstrate the superiority of DCOD. Even at an extremely low compression rate of 1%, we achieve 46.4% and 24.7% AP_{50} on the VOC and COCO, respectively, significantly reducing detector training duration.

1 Introduction

In the past decade, the field of deep learning has witnessed the emergence of high-performance models, driven by the development of convolutional and Transformer networks [10, 24, 8, 5]. These models, while benefiting from large-scale datasets, also encounter significant challenges such as data storage and the demand for extensive training resources. Dataset condensation (or distillation) [27, 32, 2] has emerged in response, aiming to synthesize a small amount of data that approximates the training effectiveness of original datasets, thus offering hope to alleviate this dilemma.

Current dataset condensation (DC) research mainly focuses on image classification and is divided into two main frameworks: Meta-learning and Data-matching [11]. As shown in Figure 1 (a), the Meta-learning framework [27, 17, 34, 15] optimizes synthetic datasets by minimizing performance risks on the original dataset's validation set, using a bi-level optimization process. In contrast, Figure 1 (b) illustrates the Data-matching framework, which aligns gradients [32], feature distributions [31], or training trajectories [2] to simulate the original data's impact during different model training stages.

[*]Work done when Ding Qi is an intern in Tencent YouTu Lab, and the advisor is Jian Li.
[†] Corresponding author

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

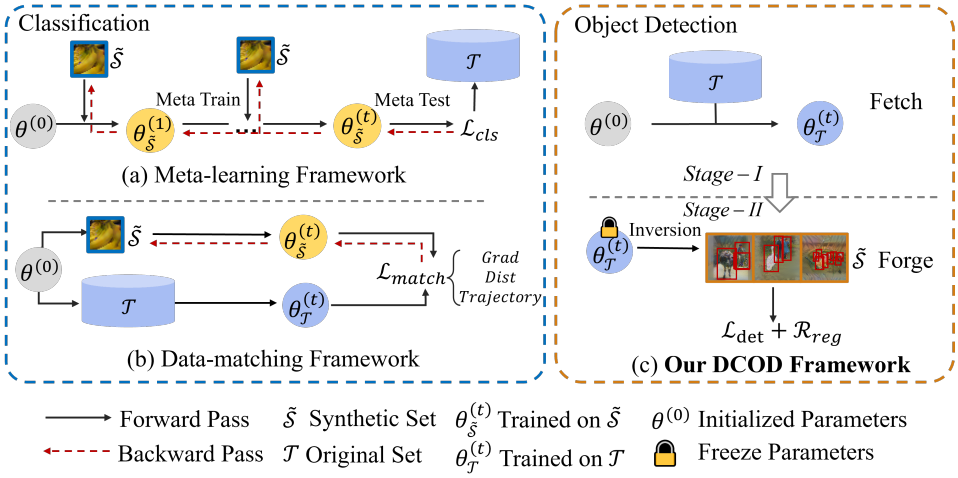

Figure 1: (a) Meta-learning treats synthetic set updates in DC as a meta-task; (b) Data-matching indirectly matches gradients, feature distributions, and training trajectories generated during network training; (c) Our proposed DCOD, the first for object detection, decouples the bi-level optimization of methods a and b, following a Fetch and Forge two-stage approach.

While both frameworks above make significant contributions to the field, they share common limitations. The complexity and computational expense of their bi-level optimization process restrict their application, typically confining them to smaller datasets (e.g., CIFAR10, CIFAR100, Tiny-ImageNet) and simpler network architectures (e.g., ConvNet-3, AlexNet, VGG-11, ResNet-18). This limitation is particularly pronounced in object detection tasks, which are inherently more complex than image classification. Specifically, i) the requirement in object detection for simultaneous localization and classification of multiple objects in an image considerably complicates the condensation process; and ii) the large-scale, high-resolution nature of detection dataset images, along with the complexity of detection network structures, poses significant challenges in extending existing DC methods.

In this study, we propose DCOD, **the first dataset condensation framework for object detection.** DCOD is distinct in its ability to flexibly generate foreground objects of varying sizes, shapes, and categories at any position within an image, as demonstrated in Figure 2. Additionally, DCOD streamlines the process by eliminating the complexities of traditional bi-level optimization, enhancing its compatibility with complex detection networks and large-scale, high-resolution data. **DCOD operates in two stages: Fetch and Forge.** As in Figure 1 (c), in the first stage Fetch, we train an object detector on the original dataset following a standard procedure, during which key information from the original data is stored within the detector. Then, in the second stage, we employ the model inversion to synthesize images from the trained detector. Specifically, we freeze the detector's weights and input initialization images and labels (including position, size, and category) randomly sampled from the original dataset to capture the instance distribution of real originals. During the inversion process, we propose the Foreground Background Decoupling ($\mathcal{F}_{BD}$) to enhance attention to the foreground areas through random erasure guided by a coarse mask and Incremental PatchExpand ($\mathcal{I}_{PE}$) expands a single image into multiple patches, each guided by different target labels, thus synthesizing a richer variety of instances. Finally, we utilize the detector's loss function as the guiding task loss for dataset condensation. To ensure the quality of the generated images, we implement two types of regularization: pixel-level alignment and feature-level alignment. Our method underwent rigorous evaluation on the MS COCO [13] and Pascal VOC [7, 6] datasets, achieving top-tier results that highlight its significant effectiveness. With some problems left open and a considerable improvement room existing, we hope this pilot study will attract more community interest.

The primary contributions of this paper can be summarized as follows:

- We propose DCOD, the first dataset condensation framework for object detection datasets. Utilizing a two-stage process of Fetch and Forge, it can simultaneously condense crucial localization and classification information from the original dataset.

- For the multi-object distribution characteristic of object detection datasets, we propose the Foreground Background Decoupling strategy and Incremental PatchExpand ($\mathcal{I}_{PE}$) which notably boost the diversity of multi-instances, all within a constrained storage budget.

(a) Single-object synthetic images

(b) Mutliple-object synthetic images

Figure 2: Visualization of images synthesized using DCOD. (a) A synthetic image contains only one category of foreground. (b) A synthetic image contains multiple foregrounds of different sizes, shapes, and categories.

- Extensive experiments on various detection datasets underscore the superiority of DCOD. Remarkably, even at an extremely low compression rate of 1%, we achieve 46.4% and 24.5% AP50 on the VOC and COCO datasets respectively, significantly reducing the training duration for object detection.

2 Related work

Dataset Condensation. Dataset condensation (or distillation) aims to compress large-scale original data into a small amount of synthetic data, accelerating network training while maintaining comparable performance. Current research primarily focuses on the field of image classification and can be divided into two frameworks: meta-learning and data matching. In the meta-learning framework, methods such as DD [27], KIP [17], RFAD [15], and FRePo [34] update model parameters in the inner loop and synthetic data in the outer loop. In the data matching framework, techniques like DC [32], DSA [30], and IDC [9] utilize gradient matching, comparing neural network weight gradients from training on both real and synthetic data. Distribution matching approaches, exemplified by DM [31], CAFE [26], DataDAM [22], and IDM [33], align real and synthetic data distributions using Maximum Mean Discrepancy (MMD), often through a single optimization process but potentially limiting the performance. Trajectory matching, such as MTT [2], optimizes synthetic data by pre-calculating and storing training trajectories of expert networks from the real dataset. A critical downside of MTT is the significant storage requirement for these trajectories.

Recent research suggests compressing original data into models rather than synthetic data. DiM [25] minimizes the difference in predicted logits between real and generated images, using a generator to store original dataset information. SRe^{2}L [29] indicates that key dataset information can be preserved in deep neural network training. Inspired by this, we argue that the localization and classification information of the detection data can be learned and preserved by the detector, allowing the key information to be recovered to reconstruct the synthetic image. Unlike SRe^{2}L, we focus on object detection datasets with multiple instances in a single image. We propose a foreground-background decoupling strategy and incremental PatchExpand to enhance the updating of synthesized multiple instances.

Diverging from the traditional focus on image classification, our study pioneers dataset condensation in object detection. We introduce a streamlined approach that separates the traditional bi-level optimization into a two-stage Fetch and Forge process. This strategy effectively realizes dataset condensation in object detection tasks with increased efficiency.

High Training Overhead in Object Detection. Object detection methods are broadly divided into two categories: one-stage and two-stage detectors. One-stage detectors, such as YOLO series [19, 20], SSD [14], DSFD [12], are valued for their speed and simplicity. Conversely, two-stage detectors like Faster R-CNN [21] prioritize accuracy. Training object detectors, particularly on a single GPU, demands substantial time. Training a complex model like Faster R-CNN on datasets such as COCO can take several days or even weeks, depending on the configuration and hardware. This extensive training time underscores the pressing need for more efficient training methodologies in object

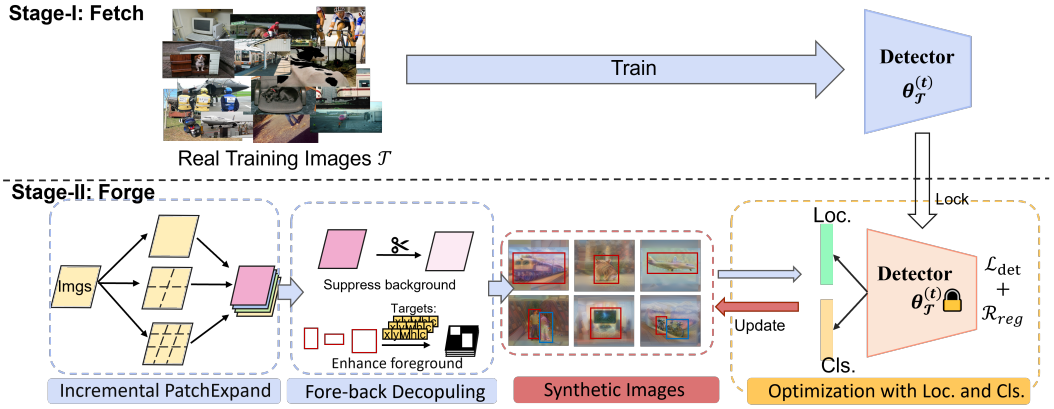

Figure 3: Overview of the DCOD framework: In the first stage, Fetch, a detector is trained on the original images, encapsulating key information from the original dataset. In the second stage, Forge, a randomly initialized synthetic set is enhanced through Foreground Background Decoupling and Incremental PatchExpand on the initial images, which are then input into the trained detector. Guided by targets, specific category targets are updated in the corresponding areas of the images. The loss of the detector serves as the task loss for condensation, while pixel-level and feature-level regularization ensure the quality of the generated images.

detection, highlighting the growing importance of dataset condensation advancements. To address this issue, we introduce the first dataset condensation framework for object detection.

3 Dataset Condensation for Object Detection

3.1 Preliminary

Dataset condensation reduces a large real dataset $\mathcal{T}$ into a smaller synthetic dataset $\widetilde{\mathcal{S}}$. We first extend the existing DC [27, 32] optimization framework to accommodate detection task. For object detection, each image is annotated with multiple bounding boxes and their associated class labels. The real dataset $\mathcal{T}$, containing N labeled images, is given by $\mathcal{T} = \{(X_i, A_i)\}_{i=1}^{N}$, where $A_i = \{(x_j, y_j, w_j, h_j, c_j)\}_{j=1}^{n_i}$, with x_j, y_j, w_j, h_j representing the coordinates and sizes of the j-th bounding box, and c_j being the class label. The synthetic dataset $\widetilde{\mathcal{S}}$, comprising M images where $M \ll N$, is formulated as $\widetilde{\mathcal{S}} = \{(\widetilde{X}_i, \widetilde{A}_i)\}_{i=1}^{M}$, where $\widetilde{A}_i = \{(\widetilde{x}_j, \widetilde{y}_j, \widetilde{w}_j, \widetilde{h}_j, \widetilde{c}_j)\}_{j=1}^{\widetilde{n}_i}$ represents the synthetic bounding boxes and their corresponding class labels. The optimization process for dataset condensation is twofold:

Model Parameters Update (Inner Loop): The model parameters θ are optimized over the synthetic dataset $\widetilde{\mathcal{S}}$ to minimize the loss function $L^{\widetilde{\mathcal{S}}}(\theta)$:

$$\theta_{\widetilde{\mathcal{S}}}(\widetilde{\mathcal{S}}) = \arg\min_{\theta} L^{\widetilde{\mathcal{S}}}(\theta). \tag{1}$$

Synthetic Dataset Update (Outer Loop): The synthetic dataset $\widetilde{\mathcal{S}}$ is refined to minimize the loss on the real dataset $\mathcal{T}$ using the model parameters $\theta_{\widetilde{\mathcal{S}}}(\widetilde{\mathcal{S}})$ obtained from the inner loop:

$$\widetilde{\mathcal{S}}^{*} = \arg\min_{\mathcal{S}} L^{\mathcal{T}}(\theta_{\widetilde{\mathcal{S}}}(\widetilde{\mathcal{S}})). \tag{2}$$

However, this optimization process underlying this procedure involves a complex bi-level optimization scheme, which leads to high computational costs. The challenge intensifies when dealing with intricate model structures in the inner-loop or large synthetic datasets in the outer-loop. In the context of object detection, where high-resolution images and complex network structures are the norm, employing such bi-level optimization becomes even more impractical. This emphasizes the need for more streamlined and practical methods in dataset condensation, especially for tasks like object detection that involve high computational demands.

3.2 Fetch and Forge: DCOD Framework

Pioneering the application of dataset condensation to object detection, we develop the Dataset Condensation for Object Detection (DCOD) framework, which is a novel two-stage process depicted in Figure 3. This method diverges from the traditional bi-level optimization that compresses information directly into synthetic data. Instead, we optimize the model and synthetic data separately. Subsequent sections will delve into the optimization objectives for the Fetch and Forge stages.

Stage-I: Fetch. For object detection tasks, the real dataset is given by $\mathcal{T} = \{(X_i, A_i)\}_{i=1}^{N}$, with A_i encompassing bounding boxes and class labels. By training an object detector ψ_θ parameterized with θ on the original dataset $\mathcal{T}$, we capture key information crucial for the tasks of localization and recognition within images. The optimization can be expressed as:

$$\theta_{\mathcal{T}} = \arg\min_{\theta} \mathcal{L}_{\text{det}}(\psi_\theta(X), A), \tag{3}$$

with $\mathcal{L}_{\text{det}}$ representing the composite loss function that combines localization loss $\mathcal{L}_{\text{loc}}$ and classification loss $\mathcal{L}_{\text{cls}}$:

$$\mathcal{L}_{\text{det}}(\psi_\theta(X), A) = \mathcal{L}_{\text{loc}}(\psi_\theta(X), A) + \mathcal{L}_{\text{cls}}(\psi_\theta(X), A), \tag{4}$$

minimizing $\mathcal{L}_{\text{det}}$ adjusts θ to detect objects with higher precision.

Stage-II: Forge. In the forge stage, we borrow principles from model inversion [16, 28, 3], utilizing a well-trained detection model to guide the embedding of essential information back into the synthetic images. To achieve this, it is essential to recognize the unique aspects of detection tasks: an image typically contains multiple foreground objects with different positions, sizes, and shapes. Therefore, we propose Foreground Background Decoupling ($\mathcal{F}_{BD}$) and Incremental PatchExpand ($\mathcal{I}_{PE}$). Finally, we use the detection loss as the task loss to guide the update of the targets' corresponding areas with the foregrounds of various categories.

Foreground-Background Decoupling. We initialize the synthetic set $\widetilde{\mathcal{S}} = \{(\widetilde{X}_i, \widetilde{A}_i)\}_{i=1}^{M}$ by randomly selecting a subset of images and their corresponding targets from the original dataset, which includes position coordinates and category labels. The primary objective of object detection is to distinguish the foreground from the background and then classify the foreground objects. Therefore, we prioritize synthesizing the foreground. Additionally, while the background is treated as a single category in detection tasks, we adopt a background suppression strategy to avoid blending different contextual semantics (e.g., integrating sky or ocean information into a grassy background).

To this end, we propose Foreground-Background Decoupling ($\mathcal{F}_{BD}$) to separately handle the updating of the foreground and background areas, as follows:

$$\mathcal{F}_{BD}(\widetilde{X} = \{x_{\text{back}}, x_{\text{fore}}\}) = \begin{cases} x_{\text{back}} \leftarrow \alpha x_{\text{back}} & (\alpha < 1) \\ \mathcal{R}_E(x_{\text{fore}}) \end{cases} \tag{5}$$

We first separate the foreground and background regions using a binary mask based on the bounding box coordinates from the original labels. For the background (x_{back}), we apply a suppression strategy controlled by a hyperparameter α, which limits updates to preserve contextual information. For the foreground (x_{fore}), we employ a random erasure method $\mathcal{R}_E$, ensuring the model focuses more on refining foreground pixels during the inversion phase. By strategically applying suppression and enhancement, this approach prioritizes important visual information, optimizing the model's ability to generate high-fidelity synthesized images.

Incremental PatchExpand. Unlike classification datasets that contain only one object, images in detection datasets sometimes contain multiple instances of several different categories. Inspired by curriculum learning, we propose Incremental PatchExpand ($\mathcal{I}_{PE}$) to learn more instances during training. By adopting this approach, we increase the number of divisions, thereby introducing complexity into the synthetic data. We first standardize all images to a uniform size via padding. Subsequently, we employ a procedure, as defined by the function:

$$\mathcal{I}_{PE}(\widetilde{X}) = \text{Expand}(\text{Patch}(\text{Padding}(\widetilde{X}), k), \widetilde{A}) \tag{6}$$

Algorithm 1 Dataset Condensation for Object Detection

Input: Training set $\mathcal{T}$

Required: Randomly initialized set of synthetic samples $\widetilde{\mathcal{S}}$, initialized detector ψ_θ parameterized with θ, training iterations K, learning rate η_s

Stage-I: Fetch
Train ψ_θ on $\mathcal{T}$: $\theta_\mathcal{T} = \arg\min_\theta \mathcal{L}_{\text{det}}(\psi_\theta(X), A)$
Stage-II: Forge
for $k = 0, \ldots, K-1$ **do**
    Apply $\mathcal{F}_{BD}$ and $\mathcal{I}_{PE}$ for synthetic set $\widetilde{\mathcal{S}}$=$\mathcal{F}_{BD}(\mathcal{I}_{PE}(\widetilde{\mathcal{S}}))$
    Compute task loss L_{det} using Equ. 4
    Compute R_{reg} using Equ. 8, where compute R_{pixel} and $R_{feature}$ using Equ. 9 and Equ. 11
    Calculate $L = L_{\text{det}}(\psi\theta_\mathcal{T}(\widetilde{X}), \widetilde{A}) + R_{reg}$
    Update $\widetilde{\mathcal{S}} \leftarrow \widetilde{\mathcal{S}} - \eta_s \nabla_s L$
end for

Output: Condensed synthetic dataset $\widetilde{\mathcal{S}}$

which divides the images into $k \times k$ patches and each patch is then optimized for different targets, enhancing the diversity in object position, size, and shape. The incremental introduction of more patches allows the model to gradually adapt to various complexities and learn effectively across a broader range of scenarios.

Optimization with Pixel and Feature Regularization. The synthetic images $\widetilde{X}$ are optimized by solving the following minimization problem:

$$\widetilde{X} = \underset{\widetilde{X}}{\text{argmin}}\ L_{\text{det}}(\psi_{\theta_\mathcal{T}}(\widetilde{X}), \widetilde{A}) + R_{reg}, \tag{7}$$

where L_{det} denotes the detection loss and θ signifies the parameters of the model applied to $\widetilde{X}$. Here, $\widetilde{A}$ represents the set of annotations for the targets present in $\widetilde{X}$. The term R_{reg}, a regularization term, is crucial for preserving the intrinsic qualities of the original dataset within the synthetic images and is defined as the sum of pixel and feature regularization:

$$R_{reg} = R_{pixel} + R_{feature}, \tag{8}$$

The pixel-level regularization R_{pixel} comprises two terms:

$$R_{pixel}(\widetilde{X}) = \alpha_{TV} R_{TV}(\widetilde{X}) + \alpha_{l_2} R_{l_2}(\widetilde{X}). \tag{9}$$

with R_{TV} promoting spatial smoothness through the Total Variation regularization:

$$R_{TV}(\widetilde{X}) = \sum_{i,j} \sqrt{(\widetilde{X}_{i,j} - \widetilde{X}_{i+1,j})^2 + (\widetilde{X}_{i,j} - \widetilde{X}_{i,j+1})^2}, \tag{10}$$

and R_{l_2} minimizing the squared Euclidean norm of pixel values. The coefficients α_{TV} and α_{l_2} are hyperparameters that balance the Total Variation and L_2 terms, respectively.

Feature-level regularization $R_{feature}$ aims to align the feature statistics of the synthetic image $\widetilde{X}$ with the batch normalization (BN) layers of the pre-trained model, as follows:

$$R_{feature}(\widetilde{X}) = \sum_l \Big(\|\mu_l(\widetilde{X}) - \mu_l^{BN}\|_2^2 + \|\sigma_l^2(\widetilde{X}) - \sigma_l^{BN}\|_2^2 \Big), \tag{11}$$

where $\mu_l(\widetilde{X})$ and $\sigma_l^2(\widetilde{X})$ represent the mean and variance of the activations at the l-th BN layer of $\widetilde{X}$, while μ_l^{BN} and σ_l^{BN} correspond to the statistics derived from the pre-trained model. This feature regularization is fundamental for the consistency of the feature distribution and the utility of $\widetilde{X}$ in training object detectors. At last, we summarize the complete two-stage process in Algorithm 1.

Table 1: The Performance comparison results on Pascal VOC, with the compression ratios of 0.5%, 1%, and 2%. The methods compared include Random [18], Uniform, K-Center [23] and Herding [1, 4]. The models used in both the compression and evaluation phases are YOLOv3-SPP. Ratio (%): the ratio of condensed images to the whole training set.

Methods	mAP (%)	$AP_{50}(\%)$	$AP_{75}(\%)$	$AP_s(\%)$	$AP_m(\%)$	$AP_l(\%)$
Ratio 0.5 (%)						
Random	4.9±0.2	15.8±0.7	1.4±0.1	0.2±0.1	1.4±0.1	6.7±0.4
Uniform	5.0±0.2	15.8±0.5	1.5±0.1	0.1±0.1	1.5±0.1	6.7±0.3
K-Center	3.8±0.3	14.5±0.6	0.9±0.1	0.01±0.0	0.7±0.12	5.1±0.4
Herding	3.6±0.0	12.6±0.2	0.9±0.01	0.1±0.1	0.9±0.1	4.6±0.1
Ours	**14.2±0.5**	**37.9±0.9**	**6.6±0.4**	**2.4±0.3**	**6.6±0.4**	**18.1±0.7**
Ratio 1 (%)						
Random	8.6±0.5	25.5±0.7	3.0±0.2	0.4±0.1	3.1±0.1	12.6±0.6
Uniform	8.8±0.4	25.7±0.6	3.2±0.3	0.4±0.1	3.2±0.2	12.8±0.4
K-Center	6.1±0.2	21.9±0.9	1.4±0.1	0.1±0.0	1.3±0.2	8.3±0.3
Herding	5.5±0.2	19.3±0.5	1.3±0.1	0.2±0.0	1.3±0.1	7.9±0.3
Ours	**19.8±0.4**	**46.4±0.4**	**13.0±0.6**	**4.4±0.2**	**10.5±0.3**	**24.9±0.5**
Ratio 2 (%)						
Random	15.8±0.5	40.5±0.7	8.2±0.7	1.4±0.6	5.8±0.4	20.5±0.4
Uniform	15.9±0.3	40.6±0.5	8.2±0.5	1.5±0.3	5.8±0.3	20.6±0.3
K-Center	9.21±0.2	31.3±0.5	2.4±0.1	0.4±0.1	2.5±0.1	12.4±0.3
Herding	8.5±0.4	28.1±0.8	2.6±0.1	0.5±0.1	2.4±0.2	12.3±0.2
Ours	**23.7±0.6**	**50.7±0.6**	**18.7±0.5**	**5.6±0.7**	**13.6±0.5**	**29.9±0.7**
Whole Dataset	46.5±0.5	76.4±0.4	46.7±0.3	11.7±0.6	28.2±0.5	52.2±0.4

Table 2: The Performance comparison results on MS COCO, with the compression ratios of 0.25%, 0.5%, and 1%. The models used in both the compression and evaluation phases are YOLOv3-SPP.

Methods	mAP (%)	$AP_{50}(\%)$	$AP_{75}(\%)$
Ratio 0.25 (%)			
Random	3.5±0.1	9.7±0.1	1.6±0.1
Uniform	3.6±0.1	9.8±0.2	1.6±0.1
K-Center	1.7±0.1	6.3±0.0	0.4±0.0
Herding	1.7±0.1	5.8±0.1	0.5±0.1
Ours	**7.2±0.1**	**17.2±0.2**	**4.8±0.2**
Ratio 0.5 (%)			
Random	5.5±0.1	14.2±0.1	2.9±0.1
Uniform	5.6±0.1	14.3±0.1	2.9±0.1
K-Center	2.8±0.0	8.9±0.1	0.7±0.1
Herding	2.6±0.1	8.8±0.1	0.8±0.1
Ours	**10.0±0.1**	**21.5±0.2**	**8.0±0.1**
Ratio 1 (%)			
Random	8.3±0.0	19.7±0.1	5.3±0.1
Uniform	8.4±0.1	19.7±0.2	5.4±0.1
K-Center	4.0±0.1	12.9±0.1	1.2±0.1
Herding	4.1±0.1	12.5±0.2	1.3±0.2
Ours	**12.1±0.1**	**24.7±0.2**	**10.4±0.1**
Whole Dataset	36.1±0.2	63.6±0.3	36.5±0.2

4 Experiment

4.1 Experiment Setup

The standard evaluation of dataset condensation consists of two steps: first, condensing the original training set into a smaller synthetic dataset; then, training an initialized model with this synthetic data and evaluating it on the original dataset's test set to assess the synthetic data's effectiveness in model training.

Datasets and Metrics. Our DCOD method is evaluated on Pascal VOC [7, 6] and MS COCO [13] benchmarks, with image resolution set to 512x512. For Pascal VOC, we merge the trainval sets of VOC2007 and VOC2012 into a single training set with 16,551 images, using the VOC2007 test set for evaluation. MS COCO comprises 80 categories with 118,287 training and 5,000 test images. Both datasets are assessed using standard COCO metrics: mAP (mean Average Precision), AP_{50} (0.5 IoU threshold), AP_{75} (0.75 IoU threshold), and size-specific AP_s, AP_m, AP_l for performance evaluation. We train the network from scratch 5 times on the distilled dataset and evaluate them on the original test dataset to get the $\bar{x} \pm std$.

Implementation Details. The initialization of the synthetic set involves random sampling from the original images, adhering to the specified compression ratio. The compression rate in classification tasks is typically set to around 1%. In our task, for VOC, we evaluate at compression rates of 0.5%, 1%, and 2%, while for COCO, we report at 0.25%, 0.5%, and 1%. During the standard image synthesis process, we set the image learning rate at 0.002. The weight for the task loss is set at 1, while $R_{feature}$is assigned a weight of 0.1. The α_{TV} and α_{l2} is established at 1 and 0.001, respectively. All experiments were performed on a single V100 gpu.

4.2 Experimental Results

Counterpart Methods. As we are the first dataset condensation methods for object detection, we refer to the comparison settings of general DC in classification [27, 32] and choose three core-set selection methods: Random [18], K-center [23], and Herding [1, 4]. The Random method randomly selects real images from the original training set to form a subset. To adapt the K-center and Herding for object detection tasks, we implement a straightforward modification by using a pre-trained feature extractor (ResNet50) to process the features of entire images and employing the L2 norm to measure distances. Additionally, considering class balance, we also employ the Uniform, which is commonly used in practical applications within the field of object detection.

Table 3: Ablation Study on the components of Forge, $\mathcal{F}_{BD}$, and $\mathcal{I}_{PE}$ at 1% compression rate on Pascal VOC, with the model using YOLOv3-SPP.

Methods	Ratio 1 (%)		
	mAP(%)	AP_{50}(%)	AP_{75}(%)
baseline	8.3±0.6	25.9±0.5	2.9±0.3
baseline+$\mathcal{F}_{BD}$	10.0±0.3	27.9±0.4	3.4±0.2
baseline+$\mathcal{I}_{PE}$	17.7±0.4	42.6±0.5	11.6±0.4
baseline+$\mathcal{F}_{BD}$+$\mathcal{I}_{PE}$	19.8±0.4	46.4±0.4	13.0±0.6

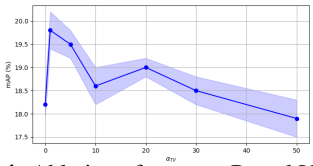

Figure 4: Ablation of α_{TV} on Pascal VOC, with a compression ratio of 1%, using YOLOv3-SPP.

Table 4: Cross-architecture performance on Pascal VOC and MS COCO. We use the one-stage detector YOLOv3-SPP and the two-stage detector Faster RCNN for the evaluation phase, respectively.

Ratio (%)	Detector	VOC		COCO	
		mAP(%)	AP_{50}(%)	mAP(%)	AP_{50}(%)
0.5	YOLOv3-SPP	14.2±0.5	37.9±0.9	10.0±0.1	21.5±0.2
	Faster-RCNN	6.3±0.4	18.5±0.7	3.7±0.3	8.4±0.2
1	YOLOv3-SPP	19.8±0.4	46.4±0.4	12.1±0.1	24.7±0.2
	Faster-RCNN	13.8±0.5	33.2±0.6	6.1±0.4	13.5±0.3

Table 5: Comparison of different initialization on Pascal VOC with a compression ratio of 0.5% using YOLOv3-SPP.

Initial Methods	mAP(%)	AP_{50}(%)
Noise	6.9±0.7	23.2±0.5
Random	14.2±0.5	37.9±0.9
K-center	11.9±0.6	34.2±0.4
Herding	11.5±0.7	33.8±0.6

Comparison Results. Table 1 and 2 provide the comparative results of our dataset condensation method on the VOC and COCO datasets, underlining its superior performance at all compression rates. According to the official ultralytics/yolov3 implementation, the detector achieves an mAP of 46.5% and AP_{50} of 76.4% on VOC when trained with the full dataset, and 35.6% mAP on COCO, which serves as an approximate upper limit of performance. On VOC, using a 2% compression rate, our method still reaches an AP_{50} of 50.7%. On COCO, at a 1% compression rate, the mAP and AP_{50} reach 12.1% and 24.7% respectively. Compared to random sampling, at the lowest compression rate of 0.5% on VOC, our method increases mAP and AP_{50} by 9.3% and 22.1% respectively, and at COCO's lowest compression rate of 0.25%, our method improves mAP and AP_{50} by 3.7% and 7.5% respectively. Moreover, while the Uniform method shows slight improvement over Random, it still falls short of the performance achieved by our approach. The Coreset method, focusing only on the overall image features and neglecting instance-level details, shows poor performance. Our method also demonstrates comprehensive advantages in multi-size object detection metrics like AP_s, AP_m, AP_l, indicating that our DCOD significantly enhances the diversity of foreground positions, sizes, and shapes.

4.3 Ablation Study

Effectiveness of Each Component. Table 3 shows that using model Inversion from Forge as our baseline, our proposed augmentations $\mathcal{I}_{PE}$ and $\mathcal{F}_{BD}$ improve performance by 1.7% and 9.4% respectively. Combined, they yield an 11.5% overall improvement. Notably, the baseline alone underperforms compared to random sampling. We found that without these augmentations, the detection loss quickly converges in early iterations, missing key positional and classification details. Thus, incorporating both $\mathcal{I}_{PE}$ and $\mathcal{F}_{BD}$ is crucial to enhance the diversity and effectiveness of the synthetic images.

Cross-architecture Generalization. We verify the performance of condensed data on new architectures, and in Table 4, our approach is able to maintain good generalization at 1% compression. However, we observe a larger performance drop at lower compression rates. This is due to the fact that the one-stage detector and two-stage detector architectures differ significantly, preserving limited information when too few images are synthesized.

Sensitivity of Hyper-Parameters. Figure 4 shows how different α_{TV} regularization weights affect our method's performance, using the VOC dataset with a 1% compression rate. The results indicate that an optimal regularization weight enhances performance, while no regularization or too much regularization reduces it. This happens because lack of regularization introduces excessive noise, and too much regularization causes excessive smoothing in the synthetic images, both of which degrade the quality of the images.

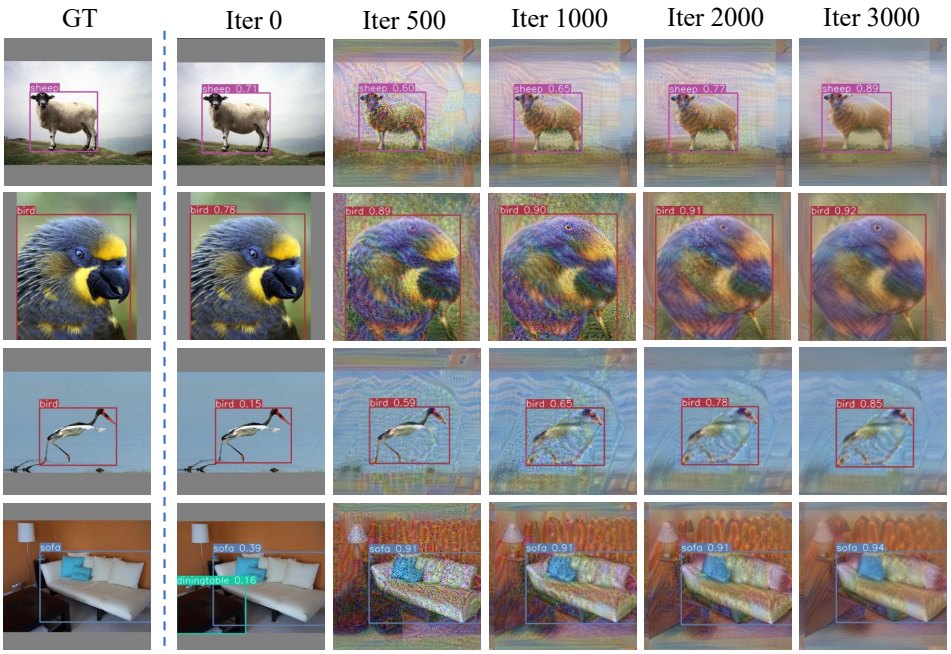

Figure 5: Visualization of different iteration steps during the condensed phase. Scores are assigned using the trained YOLOv3-spp model, based on IOU@0.5. The "GT" column represents the true labels of images. "iter0" shows the initial image scores, followed by scores of synthetic images and their bounding box at iterations 500, 1000, 2000, and 3000.

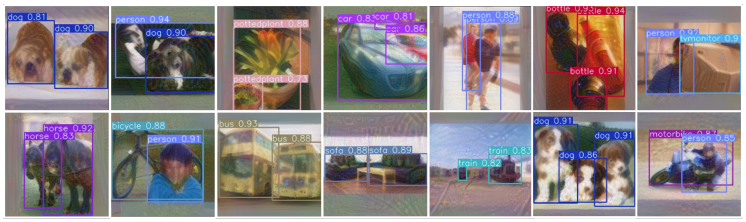

Figure 6: Visualization of the multi-instance synthesized images. Scores are assigned using the trained YOLOv3-SPP model, based on IOU@0.5.

Performance upper bound As shown in Figure 7, we compare the performance variations of the Random baseline method and the DCOD method as the compression ratio increases. The performance of the full dataset, marked by a gray line, serves as the theoretical upper bound. When the ratio is below 5%, DCOD shows a significant advantage over the random method, while as the ratio exceeds 20%, the performance of both methods converge.

Discussion on Initialization We discuss the impact of initialization sampling on the performance of the synthetic dataset in this section. As shown in Table 5, the Random method, which is the primary choice for the experiments in this paper, is simple and performs well. Initialization strategies based on K-center and Herding lead to a decline in performance. This is these core-set methods fail to account for the complex factors in detection datasets, such as class distribution and object size distribution, resulting in significant differences between the synthetic and original datasets. Using noise as the initialization method, due to the absence of any prior information from the original dataset, causes a notable drop in the effectiveness of the synthetic dataset.

4.4 Visualization

We present a comparative analysis of synthetic images across various iterations, as shown in Figure 5. Notably, there is a progressive enhancement in the foreground scores with advancing iterations, indicative of the incremental integration of category-specific beneficial information into the images. Specifically, for sheep in the first row, the scores at iter500 and iter1000 start lower than the initial

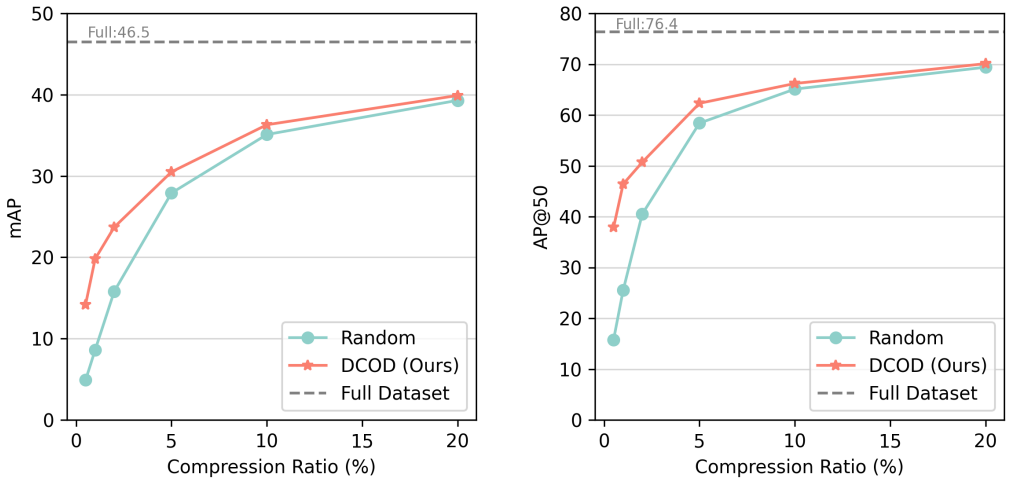

Figure 7: Visualization of performance curves on Pascal VOC using YOLOv3-SPP as the compression ratio increases.

image but surpass initial scores after iter2000. This trend implies that early images are initially disrupted by noise from the update process, but as iterations progress, pixel, and feature regularization help maintain image clarity.

Additionally, we also showcase the visualization results of synthetic images containing multiple instances, as shown in Figure 6, demonstrating that DCOD effectively supports the generation of multiple instances required in detection tasks.

5 Conclusion, Limitations and Future Work

In this study, we introduce the first dataset condensation framework for object detection, DCOD. It is a two-stage framework where, in the first stage, Fetch stores information vital for detection from the original dataset into model parameters. In the second stage, Forge, the images are then reconstructed through model Inversion. We propose foreground background decoupling and incremental PatchExpand, to further improve the efficiency and diversity of the synthetic images. Extensive experiments are conducted on two standard detection benchmarks, Pascal VOC and MS COCO, to demonstrate that DCOD can significantly maintain superior performance even at an extremely low compression rate.

We acknowledge the limitations of our work from two perspectives. First, due to the significant differences among object detectors, we did not extend our approach to more complex detectors like DETR, as this might require more specialized designs to accommodate their structures. Second, the performance across different architectures is still insufficient and remains an area that needs improvement.

6 Acknowledgments and Disclosure of Funding

This work was supported by National Natural Science Fund of China (62076184 , 62473286) in part by Shanghai Natural Science Foundation (22ZR1466700).

References

[1] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 233–248, 2018.

[2] 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.

[3] Akshay Chawla, Hongxu Yin, Pavlo Molchanov, and Jose Alvarez. Data-free knowledge distillation for object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3289–3298, 2021.

[4] Yutian Chen, Max Welling, and Alex Smola. Super-samples from kernel herding. *arXiv preprint arXiv:1203.3472*, 2012.

[5] 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. *arXiv preprint arXiv:2010.11929*, 2020.

[6] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111:98–136, 2015.

[7] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010.

[8] 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.

[9] 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.

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

[11] Shiye Lei and Dacheng Tao. A comprehensive survey to dataset distillation. *arXiv preprint arXiv:2301.05603*, 2023.

[12] Jian Li, Yabiao Wang, Changan Wang, Ying Tai, Jianjun Qian, Jian Yang, Chengjie Wang, Jilin Li, and Feiyue Huang. Dsfd: dual shot face detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5060–5069, 2019.

[13] 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.

[14] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[15] Noel Loo, Ramin Hasani, Alexander Amini, and Daniela Rus. Efficient dataset distillation using random feature approximation. *Advances in Neural Information Processing Systems*, 35:13877–13891, 2022.

[16] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks. 2015.

[17] Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. In *International Conference on Learning Representations*, 2020.

[18] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[19] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[20] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[22] 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.

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

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

[25] 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.

[26] 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.

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

[28] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020.

[29] Zeyuan Yin, Eric Xing, and Zhiqiang Shen. Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective. *arXiv preprint arXiv:2306.13092*, 2023.

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

[31] 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.

[32] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *International Conference on Learning Representations*, 2020.

[33] 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.

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

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 abstract and introduction clearly outline the primary contributions and scope of the paper.

   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: We describe it in Section 5.

   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]

Justification:

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: We describe it in Section 4.1.

   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: [No]

Justification: We commit to releasing the complete code later.

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, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We describe it in Section 4.1.

   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: In all exp table we consider error bars.

   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.
   - 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: We use a singlev100 gpu in section4.1.

   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:

   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: [Yes]

    Justification: The potential positive societal impacts are described in Abstract and Introduction. In our view, this work has no potential negative social impact.

    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:

    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: [NA]

    Justification:

    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.
    - 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: [NA]

    Justification:

    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: The paper does not involve crowdsourcing nor research with human subjects.

    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: The paper does not involve crowdsourcing nor research with human subjects.

    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.