LOSSLESS DATASET COMPRESSION VIA DATASET QUANTIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The power of state-of-the-art deep learning models heavily depends on large amounts (millions or even billions) of training data, which hinders researchers having limited resources from conducting relevant researches and causes heavy CO_2 emission. Dataset distillation methods are thus developed to compress large datasets into smaller ones to reduce model training cost, by synthesizing samples to match the original ones w.r.t. certain metrics like the training loss. However, existing methods generally suffer poor scalability (not applicable to compressing large-scale datasets such as ImageNet), and limited generalizability for training other model architectures. We empirically observe the reason is that the condensed datasets have lost the sample diversity of the original datasets. Driven by this, we study dataset compression from a new perspective—what is the minimum number of pixels necessary to represent the whole dataset without losing its diversity?—and develop a new dataset quantization (DQ) framework. DQ conducts compression at two levels: the sample level and the pixel level. It introduces a sample-level quantizer to find a compact set of samples to better represent distribution of the full dataset and a pixel-level quantizer to find the minimum number of pixels to describe every single image. Combining these two quantizers, DQ achieves new state-of-the-art dataset lossless compression ratio and provides compressed datasets practical for training models with a large variety of architectures. Specifically, for image classification, it successfully removes 40% data with only 0.4% top-5 accuracy drop on ImageNet and almost zero accuracy drop on CIFAR-10. We further verify that the model weights pre-trained on the 40% compressed dataset only lose 0.2% mAP on COCO dataset for object detection and 0.3% mIoU on ADE20k for segmentation. Code will be made public.

1 INTRODUCTION

Deep learning models have shown prominent performance in a wide range of fields such as computer vision (He et al., 2016; 2017; Dosovitskiy et al., 2020) and natural language processing (Devlin et al., 2018; Bao et al., 2021). However, state-of-the-art deep learning models typically require to be pre-trained on a huge amount of data (Kolesnikov et al., 2020; Wang et al., 2022b; Dean, 2021), which is hardly affordable for the researchers having limited computational resources. Recent dataset distillation methods target at reducing the dataset size by learning a smaller set of synthesized samples (Kim et al., 2022a; Jiang et al., 2022) to match the original samples w.r.t. certain training related metrics (*e.g.*, the training loss). In this manner, each synthetic sample is expected to contain more information than the natural samples and provides higher performance when used for model training, thus reducing the training cost.

However, these methods heavily rely on matching the loss gradients or sample features between the synthetic and real samples (Zhao et al., 2021; Wang et al., 2018), thus their generated condensed datasets are inevitably biased by the specific model architecture used for computing the matching metrics. To demonstrate this, we run a set of experiments on CIFAR-10 at 10% data keep ratio, from which we observe their compressed datasets generalize poorly for training different model architectures: the synthesized dataset can only be used for training the same model architecture synthesizing the samples; when applied for training a different model architecture, the validation accuracy drops by as high as 50%, compared with using the natural samples, as shown in Fig. 1(b).

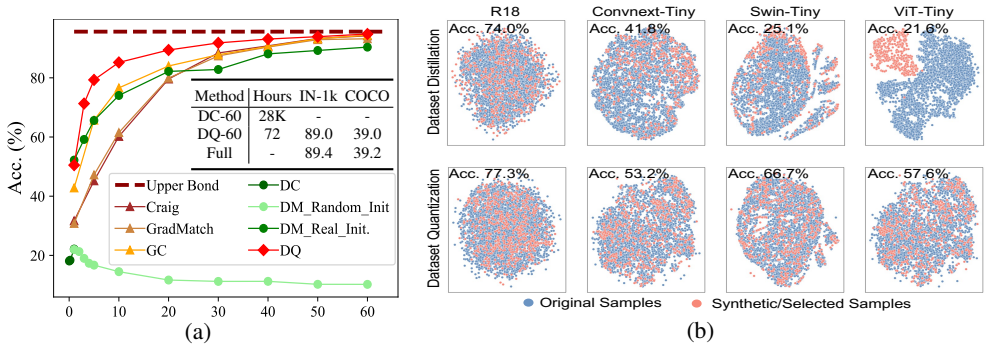


Figure 1: **Our proposed dataset quantization outperforms existing dataset distillation and coreset selection methods significantly**. (a) Model training accuracy from DD, coreset selection, and our proposed DQ method across different data keep ratios. 'Hours' denotes the time for compressing 40% of the ImageNet dataset. (b) Cross-architecture visualization of the feature distributions among the dataset generated by Distribution Matching (DM) and DQ on ResNet-18 on CIFAR-10 bird class. Compared with DM, our proposed DQ effectively captures the whole dataset distribution for all the architectures, thus generalizing better.

We investigate the reason behind this issue by visualizing features of the original and synthesized samples via tSNE. As shown in the first row of Fig. 1(b), the features of the synthesized samples are well diversified only for ResNet18 model, the model used for synthesis. When applied on other model architectures however, such as vision transformer (Dosovitskiy et al., 2020) and ConvNeXt (Liu et al., 2022), the features are distributed with significantly smaller diversity and away from the original samples, leading to the performance drop of these new models when trained on synthetic samples. As the model architecture used for sample synthesis are typically small, such constraint severely limits existing dataset distillation methods for general applications.

Different from pursuing matching-based sample synthesis, in this work, we propose to look at the dataset compression from a new perspective: since an image dataset is a collection of pixel values; then what is the minimum amount of pixels that are necessary to describe the full dataset? From this perspective, we transform the dataset compression into a dataset quantization problem. We look at this problem from two hierarchies: the sample level and the pixel level, and develop a bi-level quantization method accordingly.

More specifically, to find the minimum number of samples to describe the full dataset distribution, we propose to quantize the full dataset into a set of non-overlapping bins via a novel recursive sampler, termed as the sample-level quantizer. Each bin contains representative samples which are selected to maximize the sample diversity to capture the full dataset distribution. As for the pixel-level quantization, motivated by recent popular patch-based image representation (Dosovitskiy et al., 2020; He et al., 2022; Zhou et al., 2022), we measure the importance score of pixels at a patch level and use the most important ones to represent the images. Combining these quantization methods at two levels yields our proposed dataset quantization (DQ) method, which can generate a compact set of image patches to describe the full dataset. During training, a small portion of samples are selected uniformly within each bin to form the new dataset based on the pre-defined data keep ratio.

As shown in the second row in Fig. 1(b), with the proposed dataset quantization method, the synthesized dataset maintains a large diversity across different model architectures. Their validation accuracy is also significantly higher than those models trained with DC algorithms (*e.g.*, 34.4% higher for ViT-Tiny than Distribution Matching (DM) (Zhao & Bilen, 2021b)). Besides, all matching based image synthesis methods need to instantiate a large learnable tensor, leading to explosion of both optimization difficulty and computational cost as the number of synthesized samples increase. As a result, the model training performance saturates fast at an accuracy significantly lower than the upper bound, as shown in Fig. 1(a). In contrast, dataset quantization can adjust the size of the quantized dataset at any size and achieves lossless compression with significantly lighter computational cost. For example, even with an efficiency improved method (Zhao & Bilen, 2021b), synthesis based

methods requires 28,000 GPU hours for generating 60% of ImageNet data[1]. In contrast, with dataset quantization, it only takes 72 GPU hours, which is 388 $\times$ faster than synthesis based methods.

We conduct comprehensive experiments and show that the proposed dataset quantization method enables lossless dataset compression. Specifically, on CIFAR-10 and ImageNet, only 60% of the data are used to train the models to achieve a comparable model performance as those trained with full dataset. We further verify that the model weights pre-trained on the quantized dataset does not affect the fine-tuning performance on downstream tasks. As shown in Fig. 1(a), the ResNet-50 (He et al., 2016) model pre-trained on 60% ImageNet also achieves negligible performance drop when finetuned on COCO dataset (Lin et al., 2014) (39.0% vs 39.2%).

Our main contributions are summarized as:

- We propose a new concept, termed as Dataset Quantization (DQ), to compress the dataset with the state-of-the-art generalization capability. With DQ, the compressed dataset can be used to train any models, in contrast to the previous synthetic dataset compression methods where the compressed dataset cannot be generalized to different model architectures.
- We empirically observed a limitation of the current synthesis based dataset compression algorithms: the current dataset distillation algorithms are poorly scalable. As the data keep ratio increases, the model training performance saturates fast at a low accuracy. We conduct detailed analysis and show that the proposed dataset quantization algorithm could be scaled to achieve lossless dataset compression with the new SOTA compression ratio.
- We show that the proposed dataset quantization algorithm is significantly more computationally efficient than the conventional dataset distillation method. The traditional dataset distillation method cannot be applied to large dataset such as ImageNet due to the heavy computation cost. In contrast, dataset quantization is significantly more computation efficient and able to compress large dataset such as ImageNet. Besides, the quantization brings little performance drop when transferred to downstream tasks such as object detection and semantic segmentation on COCO and ADE20k respectively.

2 Limitations of previous dataset compression algorithms

2.1 Dataset distillation

Literature review. Dataset distillation (DD) (Wang et al., 2018) is the first method that proposes to synthesize a small-scale informative samples from a large training dataset. Specifically, they introduce a model to generate synthetic samples and optimize by minimizing the training loss between the original training data and the synthetic data. After that, a series of works based on the image synthesis technique has been proposed such as Dataset condensation (DC) (Zhao et al., 2021), DSA (Zhao & Bilen, 2021a) and IDC (Kim et al., 2022b). All those methods propose to match the gradient calculated from the original dataset and the synthetic data to avoid the nested-loop optimization. CAFE (Wang et al., 2022a) and DM (Zhao & Bilen, 2021b) introduce a feature matching strategy to reduce the influence of large-gradient samples. (Cazenavette et al., 2022) try to minimize the differences of training trajectories between original and synthetic samples.

Limitations and analysis. Almost all these methods can not be applied on large datasets, such as ImageNet, mainly due to the following limitations. (i) As shown in Fig. 1(b), image synthesis based methods can only perform well on the same model architecture synthesizing the samples; while it fails training on other model architectures. This is denoted as architecture generalization (AG) in Tab. 1. (ii) The scalability of these methods are poor. As the yellow line shows in Fig. 1(a), they saturate fast as the data keep ratio increases and can never reach the performance of the original dataset. (iii) The computation cost of these method are extremely heavy when applied on large datasets. As shown in the mini table in Fig. 1(a), condensing the whole ImageNet into 60% subset requires 28K GPU hours in total. This is denoted as time efficiency (TE) in Tab. 1.

[1]The GPU hours are estimated from the computational complexity as it is not feasible to run the algorithm.

Method	AG	TE	S	DE
Dataset distillation	✗	✗	✗	✓
Coreset selection	✓	✓	✗	✗
Dataset quantization (ours)	✓	✓	✓	✓

Table 1: Comparisons of three methods. AG, TE, S, and DE denote architecture generalization, time efficiency, stability, and data efficiency.

2.2 CORESET SELECTION

Literature review. Coreset selection is a well explored research field for compressing the dataset. The key idea is to select a subset of most representative samples out of a target dataset. The previous methods can be divided into seven categories based on the selection cateria: the geometry based methods (Chen et al., 2010; Agarwal et al., 2020; Sener & Savarese, 2018; Sinha et al., 2020), the uncertainty based method (Coleman et al., 2019), the error based methods (Toneva et al., 2018; Paul et al., 2021), the decision boundary based methods (Ducoffe & Precioso, 2018; Margatina et al., 2021), the gradient matching methods (Mirzasoleiman et al., 2020; Killamsetty et al., 2021a), the bilevel optimization method (Killamsetty et al., 2021b), and the submodularity based method (Iyer et al., 2021). The Contextual Diversity (CD) (Agarwal et al., 2020), Herding (Welling, 2009), and k-Center Greedy (Sener & Savarese, 2018) try to remove the redundant samples based on their similarity to the rest of the data samples. Margin (Coleman et al., 2019) assumes that the lower confident samples include more key information than the high confident samples, and should be added into the coreset. GradNd (Paul et al., 2021) tries to select coreset based on each samples' contribution to the loss function. Cal (Margatina et al., 2021) and Deepfool (Ducoffe & Precioso, 2018) argue that the coreset should be selected based on their difficulties for learning. Craig (Mirzasoleiman et al., 2020) and GradMatch (Killamsetty et al., 2021a) try to find an optimal coreset that has the similar gradient values with the whole dataset when training them on a network. Glister (Killamsetty et al., 2021b) introduce a validation set to maximize the log-likelihood with the whole dataset, where involves a time-consuming bilevel optimization. FL (Iyer et al., 2021) and Graph Cut (GC) (Iyer et al., 2021) consider the diversity and information simultaneously by maximizing submodular function.

Limitations and analysis. Coreset selection methods usually perform worse than DD methods when the data keep ratio is low (Zhao et al., 2021), such as only keeping 1, 10, 50 samples per category, as shown in Fig. 1(a) and Fig. 3(a) DD methods distill knowledge from the whole dataset into the synthesized samples, while coreset methods can only see the limited images. Besides, many coreset selection methods only select once from the dataset, and intend to select from high-density regions of the sample distribution. Training on these coresets leads to unstable performances on validation set, which is represented as stability (S) in Tab. 1. To better understand the differences among DD, coreset selection and DQ, we show the comparisons of these methods from four aspects in Tab. 1, for which our proposed dataset quantization consistently performs well.

3 METHOD

3.1 OVERVIEW

Dataset quantization (DQ) is proposed to compress a given dataset into a smaller one that maximally preserves the transfer-ability with flexible scalability to trade-off between the computation cost and the performance drop. To this end, DQ introduces two coherent components: a sample-level quantizer (SQ) and a pixel-level quantizer (PQ). SQ is designed to divide the target dataset distribution into several representative subsets that are called *dataset bin* in the DQ framework; and PQ is then applied on all the images within each bin to remove the non-informative and redundant pixels.

Specifically, as shown in Fig. 2, given a dataset $\mathbf{D} = \{(x_i, y_i)\}_{i=1}^{M}$ of M labeled samples, SQ is applied to sample small informative bins from $\mathbf{D}$ *recursively* with a pre-defined bin size K, yielding a set of small bins $[\mathbf{B}_1, \ldots, \mathbf{B}_N]$ with $N = M/K$. Each bin $\mathbf{B}_n = \{(x_j^{(n)}, y_j^{(n)})\}_{j=1}^{K} \subset \mathbf{D}$ is constrained to contain the samples that are most diverse among all the remained samples during the recursive selection, aiming for sufficient informativeness to capture the disturbing of the full dataset $\mathbf{D}$. Then, an pixel-level quantizer is applied on all the images within each bin to remove the redundant information by dropping non-informative pixels. After that, a small portion of the samples are uniformly selected from each bin and those samples are combined and returned as the

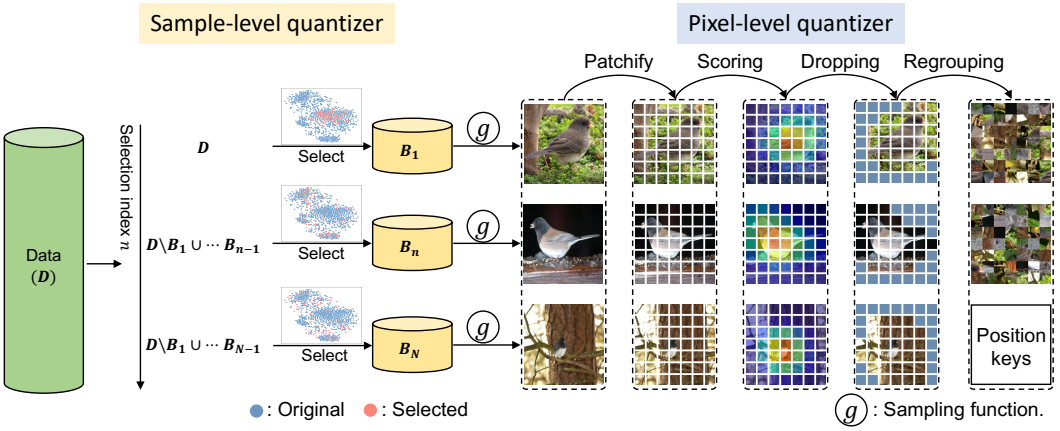


Figure 2: An overview of the proposed DQ framework. We first divide the whole dataset $\mathbf{D}$ into N non-overlapping bins $\mathbf{B}_n$ using sample-level quantizer (SQ). Then, the $\mathbf{B}^*$ is aggregated from N bins by a sampling function. A pixel-level quantizer (PQ) is designed for further reducing the redundancy from each image. PQ drops a fraction of patches with the lowest information for classification and then re-groups the remained patches to generate privacy-protected data.

compressed dataset that is ready for model training. The details of SQ and PQ are further explained in Sec. 3.2 and Sec. 3.3 respectively.

3.2 Sample-level Quantizer

As aforementioned in Sec. 2.2, previous coreset based works (Chen, 2009; Shim et al., 2021; Huang et al., 2020) can also be scaled to large datasets. However, those methods only select single subset from the full dataset in one stop manner. We empirically observe that this scheme inevitably introduces severe *selection bias*—the samples lying in the high-density regions of the dataset distribution are more often selected than others—and lead to large variance of the selection results and hence the model's performance. As a result, the quality of the selected subset is not stable and can even be worse than the subset from random selection.

To alleviate the selection bias and reduce the variance, we exploit a divide-then-aggregate strategy. We first divide the full dataset into a set of non-overlapping bins. The sample selection is then conducted within each bin and thus the variance can be effectively reduced, with a flexible scalability over different compression ratios, even with a simple uniform sampling strategy over the bins. Following this idea, we design a sample-level quantizer (SQ) for the bin creation and sample selection, with the target of maximizing the sample diversity for each bin.

Dataset bin generation. Given a dataset $\mathbf{D}$, the n-th bin $\mathbf{B}_n$ is selected by optimizing the following diversity function, which has submodularity (Iyer et al., 2021) and enjoys optimization simplicity:

$$\max_{\mathbf{B}_n \subset \mathbf{D}} I(\mathbf{B}_n; \mathbf{D} \backslash \mathbf{B}_1 \cup \cdots \cup \mathbf{B}_{n-1}) \triangleq H(\mathbf{B}_n) + H(\mathbf{D} \backslash \mathbf{B}_1 \cup \cdots \cup \mathbf{B}_{n-1}) - H(\mathbf{D}), \tag{1}$$

where $H(\cdot)$ is an entropy function as defined in (Thomas & Joy, 2006), and $\mathbf{D} \backslash \mathbf{B}_1 \cup \cdots \cup \mathbf{B}_{n-1}$ denotes the rest of the data in the dataset after selecting bin $\mathbf{B}_{n-1}$, simplified as $\mathcal{B}_{n-1}$ in the follows.

Intuitively, Eqn. (1) is selecting the subset such that the entropy to the remained samples of the dataset is maximized. Optimizing a submodular function is a typically NP-hard problem but fortunately can approximately solved by greedy algorithms (Minoux, 1978; Nemhauser et al., 1978) in practice, with almost linear time complexity w.r.t. the sample number. Following the implementation of Guo et al. (2022), we employ a Graph Cut function (Iyer et al., 2021) to calculate the diversity gains of selecting sample x into the current bin $\mathbf{B}_n^{k-1}$ with $k-1$ samples:

$$f(x, \mathbf{B}_n^{k-1}) = \sum_{x_b \in \mathbf{D} \backslash \mathcal{B}_{n-1} \cup \mathbf{B}_n^{k-1}} s(x, x_b) - \sum_{x_b \in \mathbf{D} \backslash \mathcal{B}_{n-1}} s(x, x_b), \tag{2}$$

where $s(\cdot,\cdot)$ computes the cosine feature similarity between two samples. A feature extractor is utilized to obtain the gradient embeddings (Ash et al., 2019) of $\mathbf{D}$ for similarity calculation. We iteratively select the x with the largest diversity gains to form bin $\mathbf{B}_n$, as detailed in Algorithm 1.

Algorithm 1 Sample-level dataset quantizer.

Input: original dataset $\mathbf{D}$, bin number N, bin size K, the score function $f(\cdot,\cdot)$.
For $n = 1, \ldots, N-1$ ▷ Indices of sequentially selection
  $\mathbf{B}_n^1 \leftarrow \emptyset$ ▷ Initialization of $\mathbf{B}_n$
  For $k = 1, \ldots, K$ ▷ Find K most informative samples for $\mathbf{B}_n$
    For $x \in \mathbf{D} \backslash \mathcal{B}_{n-1} \cup \mathbf{B}_n^{k-1}$, score $\Delta_x \leftarrow f(x, \mathbf{B}_n^{k-1})$
      $x^* \leftarrow \arg\max_{x \in \mathbf{D} \backslash \mathcal{B}_{n-1} \cup \mathbf{B}_n^{k-1}} \Delta_x$
      $\mathbf{B}_n^k \leftarrow \mathbf{B}_n^{k-1} \cup x^*$
Output: N dataset bins $\mathbf{B}_1, \ldots \mathbf{B}_n, \ldots \mathbf{B}_N$.

Bin sampling. After generating the dataset bins, a sampler $\mathbf{g}(\cdot,\cdot)$ is used to sample a small portion from each bin and the sampled images are then combined and used as the input to the pixel-level quantizer. The process is formally defined as:

$$\mathbf{B}^* = g(\mathbf{B}_1, \omega_1) \cup \cdots \cup g(\mathbf{B}_n, \omega_n) \cup \cdots \cup g(\mathbf{B}_N, \omega_N), \tag{3}$$

where ω_n denotes the sampling ratio. We set $\mathbf{g}(\cdot,\cdot)$ as the uniform sampler by default.

3.3 Pixel-level quantizer

The above SQ reduces the information required for describing the dataset in the sample level. However, are all pixels necessary for representing every single image? As pointed out in Masked Auto-Encoder (MAE) (He et al., 2022), with a pre-trained decoder, some image patches can be dropped without affecting the reconstruction quality of the image. Motivated by it, we present a novel Pixel Quantizer (PQ) to reduce the number of pixels utilized for describing each image. Specifically, as shown in Fig. 2, given an image x, we first feed it into a pretrained feature extractor (ResNet-18 (He et al., 2016)) to obtain the last feature map $\mathcal{M}$ and a prediction score y^c of the image class c. A group of attention scores is then calculated with the gradient values of each pixel in the last feature map following GradCAM++ (Aditya et al., 2017):

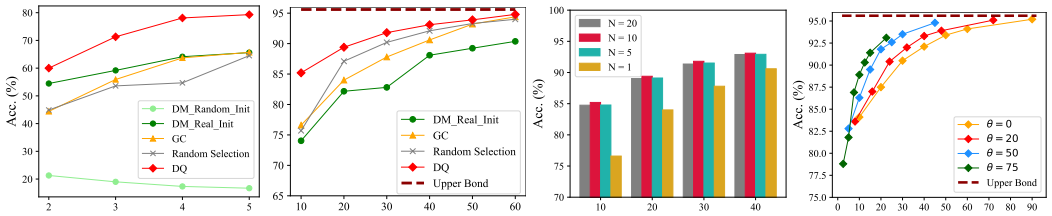
$$a^c = \sum_{i,j} \left[\frac{\frac{\partial^2 y^c}{(\partial \mathcal{M}_{ij})^2}}{2\frac{\partial^2 y^c}{(\partial \mathcal{M}_{ij})^2} + \sum_{m,n} \mathcal{M}_{mn} \{ \frac{\partial^3 y^c}{(\partial \mathcal{M}_{ij})^3} \}} \right] \mathrm{ReLU}\left(\frac{\partial y^c}{\partial \mathcal{M}_{ij}} \right), \tag{4}$$

where a^c is the attention scores for each pixel w.r.t. class c, ReLU is the Rectified Linear Unit activation function, and $(i,\ j)$ and $(m,\ n)$ are iterators over the feature map A. The pixel-wise attention score a^c is upsampled to fully cover the original input image. In order to integrate the attention information into image patches, we unify the attention scores of the corresponding pixels of a patch by their average value to generate the patch-wise importance scores $p_\cdot^c$ as follows,

$$p_k^c = \frac{1}{hw} \sum_{i=h_k}^{h_k+h} \sum_{j=w_k}^{w_k+w} a^c(i,j), \tag{5}$$

where h_k and w_k are the coordinates of the upper left corner of the patch k, and h and w are the height and width of image patches. According to the patch-wise attention scores, we drop a percentage of θ non-informative patches with smallest attention scores, and regroup the remained informative patches into new images.

Data privacy protection. Our DQ method offers data privacy protection for free. During the above patch regrouping, we further apply a shuffling operation to break their spatial order while recording their original spatial position as separate position keys, and merge the patches from different images to constitute a single image. In this way, the content of each image is largely visually distorted, which can be further encrypted by a vision encoder, and thus the privacy can be well protected. This further extends the applicability of our proposed DQ method to the fields concerning data privacy. When needing the original data, the shuffled patches are first reordered with the position keys reserved in the shuffling operation. Then we employ a strong pre-trained MAE (He et al., 2022) decoder to reconstruct the dropped patches and the original images.



(a) Low Ratio (b) High Ratio (c) Ablation on N (d) Ablation on θ (%)

Figure 3: **Testing performance of DC, random selection, GC and DQ on CIFAR-10** at (a) low and (b) high data keep ratio; and sensitiveness of DQ performance w.r.t. (c) the bin number N and (d) patch drop ratio θ across varying data keep ratios. All results are averaged over three runs.

4 Experiments

4.1 Datasets and Implementation Details

Datasets We mainly evaluate the proposed dataset quantization method on image classification tasks on CIFAR-10 (Krizhevsky et al., 2009) and ImageNet (Deng et al., 2009). To better evaluate the transferability of the pre-trained weights on the compressed dataset from DQ, we also conduct experiments on downstreaming tasks including semantic segmentation and object detection on ADE20K (Zhou et al., 2019) and COCO (Lin et al., 2014).

Implementation details Following the previous works (Kim et al., 2022a; Zhao et al., 2021), we mainly use ResNet-18 (He et al., 2016) as the model architecture for the ablation studies, unless specified otherwise. When verifying the generalization capability of the compressed dataset, we use ResNet-18 as the feature extractor during data compression and use the compressed dataset to train representative transformer and CNN architectures, including ViT (Dosovitskiy et al., 2020), Swin transformer (Liu et al., 2021), ConvNeXt (Liu et al., 2022) and MobilenetV2 (Sandler et al., 2018) models with their official training recipes. For experiments of sample-level quantizer, we use ResNet-18 and Vision Transformer (ViT-Base) models to extract features of CIFAR-10 and ImageNet-1K, respectively. The models are pre-trained on the corresponding full dataset with 10 epochs. The number of dataset bins N is set to 10 by default. We use pytorch-cifar[2] and timm library (Wightman, 2019) for model training on CIFAR-10 and ImageNet-1K datasets. We train 200 epochs for CIFAR-10 with a batch size of 128 and a cosine-annealed learning rate of 0.1. We train ImageNet in DDP manner with the default scripts of different architectures from timm. For downstream tasks and robustness experiments, we follow the default setting of mmdetection (Chen et al., 2019). We choose dataset condensation (DC) (Zhao & Bilen, 2021b) and graph cut (GC) (Iyer et al., 2021) as two strong baselines, as well as other well-established dataset compression methods.

4.2 Analysis

In this section, we investigate the effects of different components of DQ and provide apple-to-apple comparisons among DQ, DM (Zhao & Bilen, 2021b) and GC (Iyer et al., 2021).

Hyper-parameter tuning. There are two hyper-parameters that need to be pre-defined for DQ: the number of bins for the sample-level quantizer and the drop ratio for the pixel-level quantizer. We run the experiments with four different values of the bin number: 1, 5, 10 and 20. As shown in Fig. 3(c), when the number of bins is large enough, the performance does not vary significantly. However, when the bin number is set to be 1, the performance drops significantly. This is the same case of coreset selection where the dataset distribution is not quantized. This gap comes from the fact a one time subset selection could have a large selection bias. θ is the patch drop ratio in pixel-level quantizer. With a fixed dataset bin number defined ($N = 10$), we vary the drop ratio of the pixel-level quantizer and the results are shown in Fig. 3(d). It is observed that a large drop ratio improves the model training performance at small compression ratio but the performance drops significantly at

[2]https://github.com/kuangliu/pytorch-cifar

high data compression ratio. We empirically observe that the combination of $N = 10$ and $\theta = 20\%$ give the best trade-off.

Generalizability of the compressed datasets. We investigate the generalizability of the compressed datasets for training different models. Fig. 1(a) has demonstrated DQ can well preserve the dataset distribution for various architectures. We further look into the impact on the model's performance. We use DQ and DM to compress the dataset by 90%, 80% and 70% respectively and use the generated dataset to train the selected models as detailed in Sec. 4.1. The results are shown in Tab. 2. As observed, under all compression ratios, the dataset generated by DM suffers a significant performance drop when used for model training. The drop is the smallest on CNN models and the largest on transformer based models. Surprisingly, when used for training the ViT and Swin models, the performance drops by up to 70 points with DM generated dataset. In contrast, DQ compressed dataset offers much better model performance, higher than them by 30 to 40 points. The same conclusion is supported at all data compression ratios as shown in Tab. 2.

Table 2: Comparisons of cross-architecture generalization of DC and DQ on CIFAR-10. The R18 (first row) is the source architecture used to obtain condensed data or $\mathbf{B}^*$. All architectures are trained from scratch.

(a) DM on CIFAR-10.

ρ (%)	10	20	30
R18	74.0	82.2	82.8
R50	35.0	36.2	43.9
ViT	21.6	25.5	23.1
Swin	25.1	30.1	27.3
ConvNext	41.8	48.3	47.9

(b) DQ on CIFAR-10.

ρ (%)	10	20	30	100
R18	84.1	87.6	91.0	95.6
R50	82.7	88.1	90.8	95.5
ViT	58.4	66.8	72.0	80.2
Swin	69.2	79.1	84.4	90.3
ConvNext	52.8	61.8	64.2	73.0

(c) DQ on ImageNet-1K.

ρ (%)	60	80	100
R18	69.4/89.0	70.2/89.5	70.4/89.4
R50	77.1/93.4	78.6/94.2	79.2/94.4
ViT	77.0/93.5	79.0/94.6	79.7/94.9
Swin	78.6/94.1	80.6/95.0	81.4/95.5
ConvNext	80.8/95.4	81.3/95.5	82.3/96.0

Compression scalability. We investigate how the performance of different compression methods scales when the compression ratio decreases. We use DQ, DC and GC to compress the CIFAR-10 dataset to the same ratio and then use the compressed dataset to train a ResNet-18 model from scratch. The results are shown in Fig. 3. It is clearly observed that when the compression ratio is extremely high (e.g. 1%), the coreset based algorithm GC gives the lowest accuracy. Under low data compression ratio, the knowledge distillation based method DM saturates quickly and the final accuracy is 5% lower than the random sampling baseline. Under both cases, DQ achieves the highest accuracy when used for model training, demonstrating outperforming scalability.

Table 3: Evaluation of dropping patches randomly and ours with the drop ratio $\theta = 20\%$. **Bold entries** are best results. Evaluation of the GPU hours of DC and DQ.

(a) Dropping strategy comparison.

ρ (%)	1	3	5	10	30	50
Random Acc. (%)	41.1	68.6	76.8	83.5	90.1	93.1
Ours Acc. (%)	**41.8**	**69.8**	**77.3**	**83.6**	**90.4**	**93.3**

(b) GPU hours comparison.

ρ (%)	Bin creation	10	20	30	40	50	60	Total
DM	N.A.	7	14	22	29	35	41	148
DQ	1	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	1

Impact of the image patch attention. As mentioned above, we calculate a patch importance score to drop some patches to further improve data efficiency in PQ. Removing these patches randomly is a straightforward baseline. We compare the performances of randomly and GradCAM-based drops. As shown in Tab. 3(a), our method performs better than the random strategy at all data keep ratios.

Computational cost analysis. Due to the synthesizing strategy used in DM, a large tensor needs to be defined. As a result, both the computational cost and memory consumption increase linearly to the size of the dataset. We directly measure the GPU hours needed for synthesizing the dataset and the results are shown in Tab. 3(a). For each dataset compression ratio, the whole process need to be repeated. In contrast, DQ only need to calculate a sample wise similarity matrix during bin generation. The following sampling steps takes negligible GPU computations.

4.3 Comparison with state-of-the-art methods

We compare our method to previous state-of-the-art methods on both CIFAR-10 and ImageNet. The results are shown in Fig. 4. We would like to highlight that the knowledge distillation based methods are only shown on CIFAR-10 dataset as it is not feasible to verify those methods on ImageNet dataset

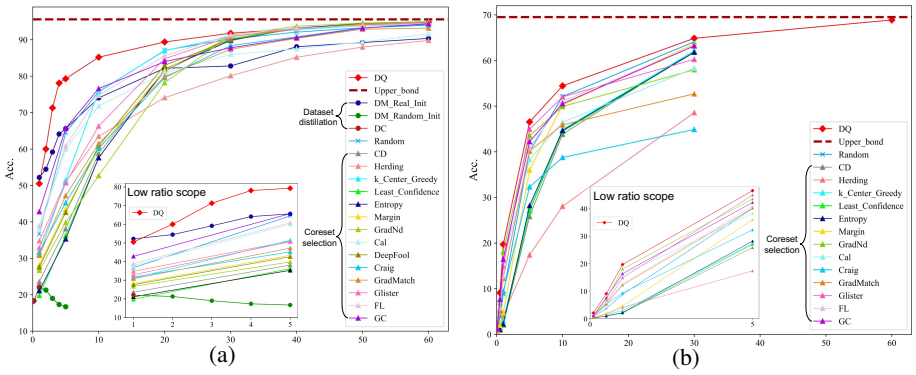
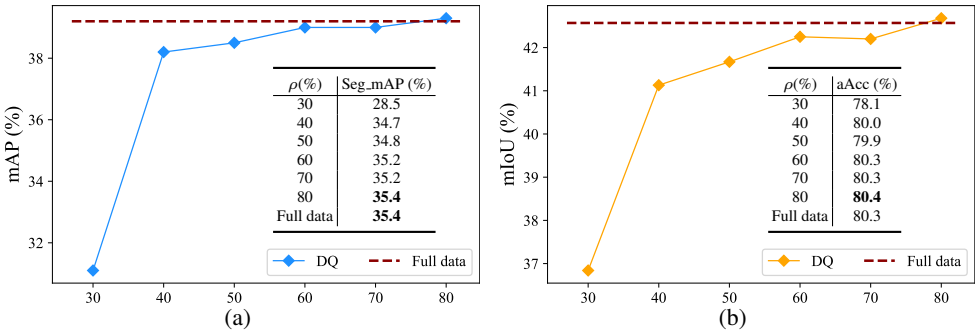


Figure 4: **Comparison of DQ with previous state-of-the-arts** with different data keep ratios on (a) CIFAR-10 and (b) ImageNet.

intuitively due to the extreme large computational cost. Our method is based on GC algorithm, while outperforms GC with a large margin on all the data keep ratio. On both CIFAR-10 and ImageNet-1K, we obtain lossless results when using only 60% data, setting a new state-of-art for dataset compression. Actually, DQ works as a play-and-plug module that could be combined with most coreset selection methods.

4.4 Performance on downstream tasks

To evaluate data efficiency of DQ on downstream tasks, we finetune the pretrained models with different data keep ratios (from 30% to 80%) on COCO and ADE20K datasets. As shown in Fig. 5, our proposed DQ achieves comparable mAP and mIOU results as training on full data when the data keep ratio is 60%. Setting the data keep ratio as 80% can achieve lossless results, which indicates the samples selected by DQ are informative. We would like to highlight that this is not feasible for DM due to the unaffordable computation cost to compress ImageNet and obtain the pre-trained model as mentioned in Sec. 4.2.



ρ(%)	Seg_mAP (%)
30	28.5
40	34.7
50	34.8
60	35.2
70	35.2
80	**35.4**
Full data	**35.4**

ρ(%)	aAcc (%)
30	78.1
40	80.0
50	79.9
60	80.3
70	80.3
80	**80.4**
Full data	80.3

Figure 5: **Performance for downstream tasks** on (a) COCO and (b) ADE20K.

5 Conclusion

In this work, we proposed a new perspective on dataset compression by treating it as a distribution quantization problem. Then we develop a new dataset quantization method, which conducts quantization in a hierarchical manner. It first quantizes the dataset on the sample level by optimizing a submodular diversity objective and then quantizes the dataset on the image level based on the image patch informativeness. Experiment results show that our proposed dataset quantization has two desirable characteristics that are not equipped by previous methods: it is transferrable to new architectures and scalable to large dataset such as ImageNet. Under all compression ratios, it achieves the new state-of-the-art performance and outperforms both sample synthesise based methods and the coreset based methods. Besides, it also offers data privacy protection as a by-product.

REFERENCES

C Aditya, S Anirban, D Abhishek, and H Prantik. Grad-cam++: Improved visual explanations for deep convolutional networks. *arXiv preprint arXiv:1710.11063*, 2017.

Sharat Agarwal, Himanshu Arora, Saket Anand, and Chetan Arora. Contextual diversity for active learning. In *ECCV*, pp. 137–153. Springer, 2020.

Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *ICLR*, 2019.

Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

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*, 2022.

Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv:1906.07155*, 2019.

Ke Chen. On coresets for k-median and k-means clustering in metric and euclidean spaces and their applications. *SIAM Journal on Computing*, 39(3):923–947, 2009.

Yutian Chen, Max Welling, and Alex Smola. Super-samples from kernel herding. *The Twenty-Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence*, 2010.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *ICLR*, 2019.

Jeff Dean. Introducing pathways: A nextgeneration ai architecture. *Google Blog*, 2021.

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*, pp. 248–255. Ieee, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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.

Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach. *arXiv preprint arXiv:1802.09841*, 2018.

Chengcheng Guo, Bo Zhao, and Yanbing Bai. Deepcore: A comprehensive library for coreset selection in deep learning. *arXiv preprint arXiv:2204.08499*, 2022.

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*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Lingxiao Huang, K Sudhir, and Nisheeth Vishnoi. Coresets for regressions with panel data. *Advances in Neural Information Processing Systems*, 33:325–337, 2020.

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

Zixuan Jiang, Jiaqi Gu, Mingjie Liu, and David Z Pan. Delving into effective gradient matching for dataset condensation. *arXiv preprint arXiv:2208.00311*, 2022.

Krishnateja Killamsetty, S Durga, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *ICML*, pp. 5464–5474, 2021a.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. Glister: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021b.

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. *arXiv preprint arXiv:2205.14959*, 2022a.

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 (ICML)*, 2022b.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *European conference on computer vision*, pp. 491–507. Springer, 2020.

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

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 *European conference on computer vision*, pp. 740–755. Springer, 2014.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986, 2022.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. Active learning by acquiring contrastive examples. *arXiv preprint arXiv:2109.03764*, 2021.

Michel Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization techniques*, pp. 234–243. Springer, 1978.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *ICML*. PMLR, 2020.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *arXiv preprint arXiv:2107.07075*, 2021.

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

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *ICLR*, 2018.

Jae-hun Shim, Kyeongbo Kong, and Suk-Ju Kang. Core-set sampling for efficient neural architecture search. *arXiv preprint arXiv:2107.06869*, 2021.

Samarth Sinha, Han Zhang, Anirudh Goyal, Yoshua Bengio, Hugo Larochelle, and Augustus Odena. Small-gan: Speeding up gan training using core-sets. In *ICML*. PMLR, 2020.

MTCAJ Thomas and A Thomas Joy. *Elements of information theory*. Wiley-Interscience, 2006.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. In *ICLR*, 2018.

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*, pp. 12196–12205, 2022a.

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

Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022b.

Max Welling. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1121–1128, 2009.

Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

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

Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. *arXiv*, 1(2):3, 2021b.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. *ICLR*, 1(2):3, 2021.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019.

Daquan Zhou, Zhiding Yu, Enze Xie, Chaowei Xiao, Animashree Anandkumar, Jiashi Feng, and Jose M Alvarez. Understanding the robustness in vision transformers. In *International Conference on Machine Learning*, pp. 27378–27394. PMLR, 2022.

A APPENDIX

We present more explanations of the proposed dataset quantization, experiment results and visualizations in this section.

A.1 SOURCE CODE

We have submitted the source code as the supplementary materials in a zipped file named as 'DQ.zip' for reproduction. A README file is also included for the instructions fo running the code. We will make it public after the submission period.

A.2 DIFFERENCES BETWEEN CORESET SELECTION AND DATASET QUANTIZATION

Coreset VS DQ We here give more detailed explanations on the difference between the coreset selection methods and our proposed dataset quantization. As shown in Fig. 6, the coreset selection only select one subset from the full data distribution. This practice will suffer from a selection bias. Besides, when the the size of the selected subset is small, it will suffer a large variance suffer the selection process. Differently, dataset quantization first quantize the full distribution into non-overlapping bins and then sampling from each bin uniformaly. As a result, the sampled data could maximally preserve the original data distribution. To verify this, we use GraphCut (Iyer et al., 2021) as a representation of the coreset based method and 10% and 20% data from ImageNet dataset and compare the results with the data distribution sampled with dataset quantization. We use a pre-trained ResNet-18 model to extract the features of the data and then visualize the extracted data via t-SNE. The results are shown in Fig. 7. It is clearly observed that the data sampled via dataset quantization do capture a more diverse distribution.

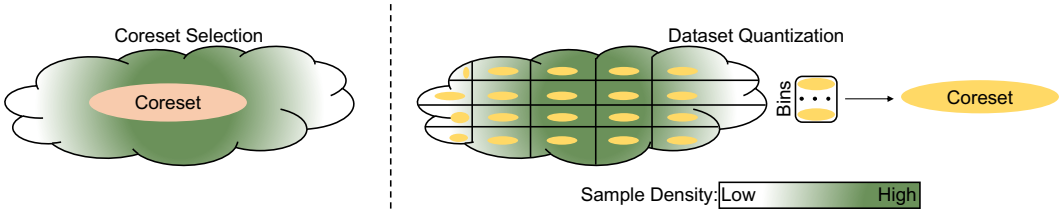


Figure 6: Differences between coreset selection methods and our dataset quantization.

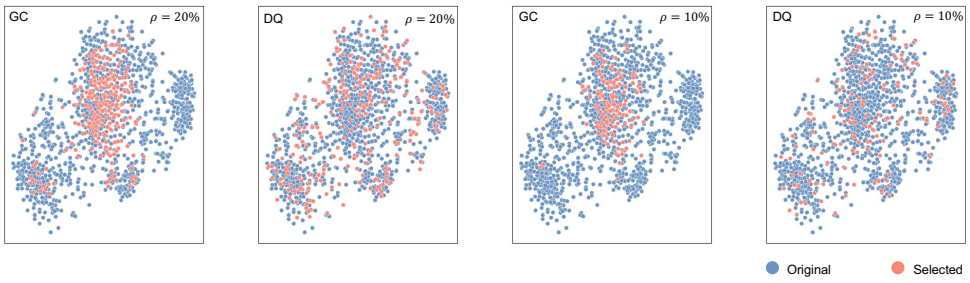


Figure 7: Visualization of the feature distributions among data selected by GraphCut and SQ.

Bin diversity of DQ To dig deeper for the reason why DQ can better preserved the data distribution. We use the same visualization method as aforementioned for the data contained within each bin. The results are shown in Fig. 8. Each bin contains 20% of the total data in the left column and 10% data in the right column. As shown, different bins are capturing different distributions. As a results, after sampling uniformly from each bin, the combined dataset enjoys a large diversity.

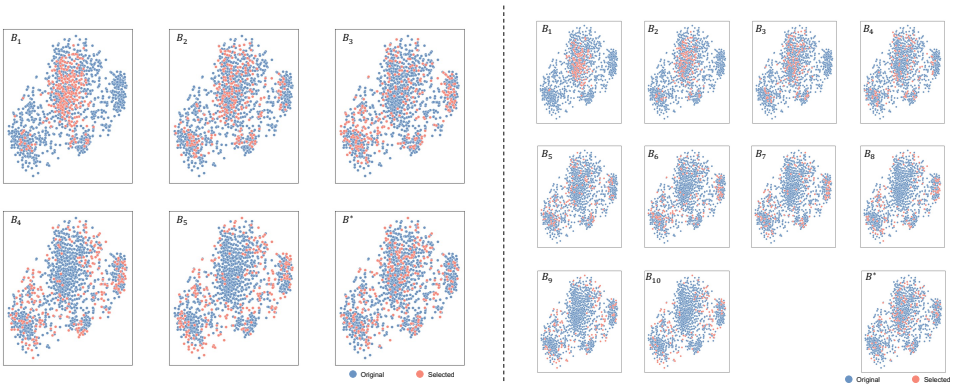


Figure 8: Visualization of the feature distributions among data selected in each bin and the final output of SQ on ImageNet dataset tench class. The bin number N and the data keep ratio ρ are set as (5, 20), (10, 10), respectively for the left and right column.

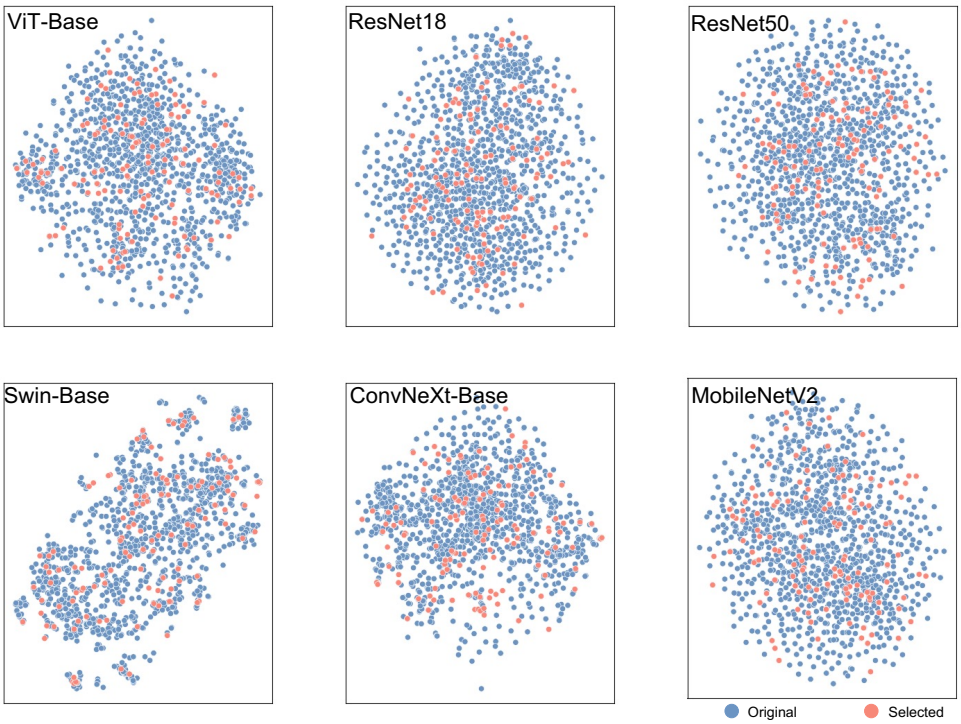


Figure 9: Cross-architecture visualiztaion of the feature distributions among the dataset generated by DQ on ViT-Base on ImageNet dataset tench class.