

From Fewer Samples to Fewer Bits: Reframing Dataset Distillation as Joint Optimization of Precision and Compactness

My H. Dinh^{1*} Aditya Sant^{1†} Akshay Malhotra¹ Keya Patani¹ Shahab Hamidi-Rad¹

¹InterDigital Communications, Inc.

{first.last}@interdigital.com

Abstract

Dataset Distillation (DD) compresses large datasets into compact synthetic ones that maintain training performance. However, current methods mainly target sample reduction, with limited consideration of data precision and its impact on efficiency. We propose Quantization-aware Dataset Distillation (QuADD), a unified framework that jointly optimizes dataset compactness and precision under fixed bit budgets. QuADD integrates a differentiable quantization module within the distillation loop, enabling end-to-end co-optimization of synthetic samples and quantization parameters. Guided by the rate-distortion perspective, we empirically analyze how bit allocation between sample count and precision influences learning performance. Our framework supports both uniform and adaptive non-uniform quantization, where the latter learns quantization levels from data to represent information-dense regions better. Experiments on image classification and 3GPP beam management tasks show that QuADD surpasses existing DD and post-quantized baselines in accuracy per bit, establishing a new standard for information-efficient dataset distillation.

1. Introduction

In the era of big data, information is continuously generated from diverse sources such as smart devices, sensors, and autonomous systems. Sharing and leveraging diverse data is crucial for training robust deep learning models, as sample diversity promotes better generalization. However, the ever-increasing data volume introduces severe computational, storage, and communication bottlenecks, particularly in distributed or resource-constrained environments. A natural approach to mitigating these costs is constructing a compact dataset that preserves the essential information of

the original data. Traditional methods such as coreset selection [18] identify representative subsets of real samples but rely on the assumption that sufficiently informative samples already exist in the dataset. Dataset Distillation (DD) [23] offers a flexible approach by synthesizing a compact set of informative samples that capture the knowledge of the full dataset. Models trained on the distilled datasets achieve comparable performance to those trained on the original data while requiring far less storage and computation.

Existing DD methods treat synthetic samples as learnable parameters, optimized to reduce the discrepancy between models trained on real and synthetic data. These methods improve dataset compactness by reducing training samples. Recent parameterization-based DD methods extend this idea by representing synthetic data in lower-dimensional spaces that exploit shared structure among samples, yielding further gains in storage efficiency [2, 10, 20, 24].

Despite these advances, most DD work overlooks a broader notion of data efficiency. A dataset is fundamentally an information representation, and its true cost depends not only on the number and dimensionality of samples but on the total bits required to store or transmit it. This becomes critical in practical scenarios like distributed learning, which relies on bandwidth-limited links, and in IoT or edge environments with strict storage constraints. These considerations motivate viewing DD through an information-centric lens, shifting the goal from reducing sample count to minimizing total information content: *from fewer samples to fewer bits*. Quantization is a natural solution to control precision. Although widely studied in model and image compression, its integration into DD remains largely unexplored. Recent work [26] introduced a color-quantization approach that learns a palette-based codebook to merge similar colors, reducing bit depth while preserving salient features like edges and shapes. However, this approach is restricted to images, adds training complexity, and scales poorly to other modalities. These limitations underscore the need for a general framework to jointly optimize dataset compactness and precision under a unified, domain-agnostic objective.

*Work completed during a Summer Internship at the AI Lab, InterDigital, Los Altos, CA. Now at Walmart Inc (email: my.dinh@walmart.com).

†Corresponding author.

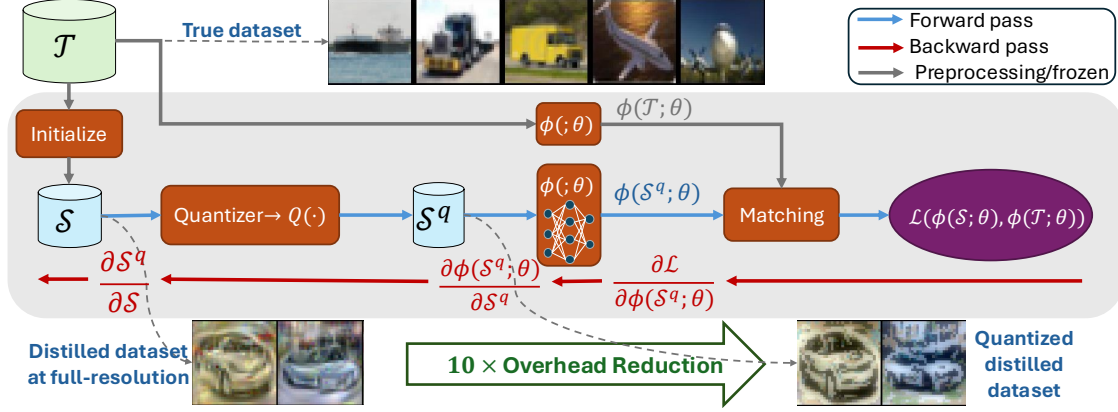


Figure 1. Illustration of the quantized dataset distillation (QuADD) framework, applied to a general DD algorithm.

Integrating quantization into DD introduces several challenges. Quantization typically involves two operations: (i) *clipping*, which constrains values to a predefined range by setting any out-of-bound values to the range boundary, and (ii) *projection*, which maps each clipped value to the nearest discrete quantization level. Their combination hinders direct optimization via standard gradient descent. Moreover, quantization introduces information loss that can amplify the reduction already imposed by DD. Thus, an effective quantization scheme must minimize distortion while adapting to the evolving data distribution during distillation. Additionally, it should also remain simple and modality-agnostic to ensure applicability across diverse data types. To address these challenges, we propose Quantization-aware Dataset Distillation (QuADD), shown in Fig. 1, a unified framework that integrates a differentiable quantization layer within the distillation loop to *jointly optimize* synthetic data and quantizer parameters end-to-end. The training becomes explicitly aware of precision: synthetic samples adapt to quantization-induced loss, while the quantizer adapts to the distilled data. By coupling quantization with distillation, QuADD shifts the goal from reducing samples to a broader objective of information-efficient learning.

Technical contributions of this work:

1. *Quantization-aware distillation.* **QuADD** embeds a differentiable quantizer in the distillation loop, jointly optimizing quantization parameters and synthetic data. Inspired by the rate distortion perspective with explicit bit accounting, we empirically analyze the trade-off between sample count and precision, revealing a “sweet spot” between fewer high-precision samples and many low-precision ones.
2. *Differentiable and adaptive quantization module.* We provide a simple, differentiable quantization layer that supports both uniform and non-uniform techniques. Our primary focus is on the adaptive non-uniform variant, which learns quantization parameters from data to allo-

cate finer bins to information-dense regions, improving task fidelity under the same bit budget.

3. *Cross-domain validation.* We evaluate our framework across diverse image datasets and a tabular wireless-communication benchmark, where data must be transmitted efficiently over a network and training occurs under resource-constrained conditions, demonstrating that QuADD generalizes beyond visual modalities.

2. Related Work

Dataset Distillation: Early DD unrolled training for performance matching [12, 21, 23], later replaced by efficient surrogates such as Gradient [29], Distribution [28, 30], and Trajectory Matching [1] aligning gradients, features, or optimization paths, respectively.

Redundancy reduction in parameterized DD: Beyond reducing sample count, parameterization-based approaches improve storage efficiency by exploiting shared information or structure. IDC [10] increases information density via deterministic multi-format transformation. HaBa [14], RTP [2], and SPEED [24] share structure across images through bases, memory dictionaries, or sparse epitomic tokens, respectively, while FreD [20] retains only informative frequency bands. These methods compact structural dimension but still rely on full-precision data.

AutoPalette [26] addresses bit-level redundancy through palette-based quantization, encoding images as color index maps and codebooks. However, its color-specific design limits applicability beyond visual data. We extend this direction by introducing a differentiable quantization module that generalizes beyond color-based schemes.

Quantization: Quantization reduces numerical precision by mapping continuous values to discrete levels, a core principle in signal compression and modern neural network optimization. Classical scalar and vector quantizers [4] balance the rate–distortion trade-off, where using fewer bits

lowers storage and transmission costs at the expense of fidelity. Uniform quantization employs fixed step sizes for simplicity but performs poorly on long-tailed distributions. Non-uniform quantization adapts resolution to dense regions, enhancing reconstruction under low-bit constraints. Modern approaches learn quantization mappings from data, including LQ-Net [27], LSQ [3], QIL [9], and adaptive log-based methods [16]. Beyond static designs, differentiable quantization enables end-to-end learning of discrete mappings via gradient approximations such as the Straight-Through Estimator (STE) [8] or soft relaxations [5, 22].

3. Problem Set-Up

Let $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^N$ denote the large-scale training dataset and $\mathcal{S} = \{(\tilde{x}_j, \tilde{y}_j)\}_{j=1}^M$ the compact synthetic dataset, where $M \ll N$. DD aims to learn $\mathcal{S}$ such that a model trained on $\mathcal{S}$ performs comparably to one trained on $\mathcal{T}$. This can be formulated as a bi-level optimization:

$$\begin{aligned} \mathcal{S}^* &= \arg \min_{\mathcal{S}} \mathbb{E}_{\theta \sim \Theta} [\mathcal{L}(\mathcal{T}; \theta_{\mathcal{S}}^*)], \\ \text{where } \theta_{\mathcal{S}}^* &= \arg \min_{\theta} \mathcal{L}(\mathcal{S}, \theta) \end{aligned} \quad (1)$$

where θ are the model parameters and $\mathcal{L}(\cdot; \theta)$ is the task loss. The inner loop optimizes on synthetic data, and the outer loop evaluates on real data.

To reduce the heavy computation in eq. (1), most DD methods adopt a one-step surrogate objective that matches model responses between real and synthetic data [1, 28, 29]:

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

where $\phi(\cdot; \theta)$ denotes intermediate features, gradients, or trajectories depending on the surrogate choice.

Conventional DD improves data efficiency by reducing either the number of synthetic samples M or the data dimension D (e.g., via low-resolution or latent-space parameterizations). However, each element of $\mathcal{S}$ is stored at fixed precision—typically 8 or 32-bit, leaving storage efficiency partly unexploited. We define the overall bit budget as

$$\text{Budget} = M \times D \times b, \quad (3)$$

where b is the bit precision per data element.

Optimizing M or D alone thus neglects precision as a controllable degree of freedom. Post-quantization of distilled data is a simple workaround, but it often degrades accuracy since synthetic samples are not optimized for low precision. To address this, we integrate a differentiable quantizer directly into the DD pipeline, enabling the synthetic data and quantization parameters to co-adapt during training. This *quantization-aware distillation* produces datasets that remain informative even under tight bit budgets.

Finally, this formulation can be viewed through a *rate–distortion* lens, where the bit budget is the “rate” and the performance drop the “distortion.” Although QuADD does not explicitly optimize this function, we empirically study how different (M, b) combinations affect performance under fixed storage budgets.

4. Quantization-Aware Dataset Distillation

This section details the Quantization-Aware Dataset Distillation (QuADD) framework. We first introduce the training objective and gradient flow through the quantizer, then describe the differentiable quantization framework for both uniform and adaptive non-uniform settings.

4.1. Training Objectives

To incorporate quantization into DD loop, QuADD augments the standard DD loop with a differentiable quantizer $Q(\cdot)$. Given synthetic samples $\tilde{x}_j \in \mathcal{S}$, the quantized representations are

$$\tilde{x}_j^q = Q(\tilde{x}_j), \quad \mathcal{S}^q = \{(\tilde{x}_j^q, \tilde{y}_j)\}_{j=1}^M. \quad (4)$$

Instead of matching real and synthetic data in full precision, QuADD enforces performance equivalence between real data $\mathcal{T}$ and quantized synthetic data $\mathcal{S}^q$ as

$$\mathcal{S}^* = \arg \min_{\mathcal{S}} \mathbb{E}_{\theta \sim \Theta} [\mathcal{L}(\phi(\mathcal{T}; \theta), \phi(\mathcal{S}^q; \theta))], \quad (5)$$

where the bit precision b of $Q(\cdot)$ controls the storage budget.

Gradient propagation: Quantization introduces nondifferentiable operations (clipping and rounding), requiring special handling for backpropagation. Loss gradients with respect to the synthetic data are computed via the chain rule:

$$\frac{\partial \mathcal{L}}{\partial \mathcal{S}} = \frac{\partial \mathcal{L}}{\partial \phi} \frac{\partial \phi}{\partial \mathcal{S}^q} \frac{\partial \mathcal{S}^q}{\partial \mathcal{S}}, \quad (6)$$

where the last term $\partial \mathcal{S}^q / \partial \mathcal{S}$ involves differentiation through the quantizer and is approximated using straight-through or smooth surrogates described next.

4.2. Differentiable Quantization Layer

The quantization layer transforms full-precision synthetic data into quantized data while remaining differentiable, enabling end-to-end optimization with the DD objective.

Quantization is composed of clipping and projection, as

$$Q(x; \alpha, b) = \Pi_{Q(\alpha, b)} [x; \alpha] \quad (7)$$

where α is the clipping threshold, $[\cdot, \alpha]$ clips inputs to $[-\alpha, \alpha]$, and $Q(\alpha, b)$ is the b -bit codebook. $\Pi_{Q(\alpha, b)}(\cdot)$ projects each value to its nearest level. The $\text{round}(\cdot)$ operation within the projection is non-differentiable, thus requires special handling to enable gradient training.

Forward pass: We explored two different forward modes of the round operator:

- Hard rounding: $\text{round}(\cdot)$ to the nearest codebook value; applicable to both uniform and non-uniform quantizers.
- Soft rounding: a continuous relaxation that approximates rounding [15, 17]; used only for the uniform case.

Backward pass: During backpropagation, gradients are estimated as:

- Straight-through estimator (STE): for hard rounding, treat Q as identity within the clipping range:

$$\frac{\partial x^q}{\partial x} \approx \mathbf{1}(|x| \leq \alpha).$$

- Analytic surrogate soft quantizers use the derivative of the smooth approximation of rounding (see Appendix A in the supplementary material for this work).

4.2.1. Baseline: Uniform quantizer

When $Q(\alpha, b)$ is uniformly spaced, this reduces to the standard uniform quantizer, widely used in model compression and image coding, given as:

$$Q^u(\alpha, b) = \alpha \times \left\{ -1, \frac{\pm 1}{2^{b-1} - 1}, \dots, 1 \right\}. \quad (8)$$

To reduce gradient discontinuities near the clipping boundary, *companding functions* (e.g., tanh, sigmoid, or CDF-based) offer smooth alternatives to hard clipping, with α fixed by the chosen function (e.g., $\alpha = 1$ for tanh).

4.2.2. Adaptive Non-uniform Quantizer

The adaptive quantizer adopts the **Additive Powers-of-Two (APoT)** scheme [13], which expresses each quantized value as a sum of scaled powers of two:

$$Q^a(\alpha, b) = \gamma \times \left\{ \sum_{i=0}^{n-1} p_i \mid p_i \in \left\{ 0, \frac{1}{2^i}, \dots, \frac{1}{2^{i+(2^k-2)n}} \right\} \right\} \quad (9)$$

where γ scales the dynamic range to $[-\alpha, \alpha]$, k is the *base bit-width* for each additive term, and $n = b/k$ denotes the number of additive bases. This representation achieves non-uniform level spacing, allocating more quantization density to dense regions of the data distribution.

Forward pass: Here, the forward pass uses hard rounding to project values onto additive combinations of PoT bases.

Backward pass: The clipping threshold α is learnable and updated automatically through back-propagation, allowing the quantizer to adapt to the evolving data distribution. Because the standard STE updates α only from out-of-range values, we adopt the Reparameterized Clipping Function (RCF) [13], which rescales inputs to a normalized range before projection and restores their scale afterward. This allows α to receive gradients from all samples:

$$\frac{\partial x^q}{\partial \alpha} = \begin{cases} \text{sign}(x), & \text{if } |x| > \alpha, \\ \Pi_{Q(1,b)}\left(\frac{x}{\alpha}\right) - \frac{x}{\alpha}, & \text{if } |x| \leq \alpha. \end{cases}$$

Algorithm 1: General Quantization-Aware Dataset Distillation

Input: Dataset $\mathcal{T}$, quantizer $Q(\cdot; \alpha, b)$, DD framework $\phi(\cdot)$ and loss $\mathcal{L}_{\text{distill}}(\cdot, \cdot)$

Output: Quantized distilled dataset $S^q = Q^*(S^*)$

- 1 Initialize S^q using (10)-(11) and quantizer parameters (α, b)
 - 2 **for each** DD iteration **do**
 - 3 Sample mini-batch $\mathcal{B}_{\mathcal{T}} \sim \mathcal{T}$
 - 4 Quantize synthetic data: $S^q = Q(S)$
 - 5 Compute loss $\mathcal{L}_{\text{distill}}(\phi(\mathcal{B}_{\mathcal{T}}), \phi(S^q))$
 - 6 Backpropagate through both S and Q using (6)
 - 7 Update S and Q via gradient descent
-

Remarks. While APoT was originally designed for model quantization, its adaptation to dataset distillation requires careful initialization and normalization for stable convergence (see Appendix B in the supplementary material for this work). In QuADD, it is computationally efficient, with only one learnable parameter (α), and achieves higher fidelity than uniform quantization, making it the default choice in all experiments.

4.3. Algorithm Implementation for QuADD

Quantization-guided initialization: QuADD initializes synthetic data following the quantization-guided selection strategy of [26], adapted from color to scalar quantization. The real dataset $\mathcal{T}$ is first quantized using a uniform quantizer to obtain $\mathcal{T}^q = Q(\mathcal{T})$. Let $A \subset \mathcal{T}^q$ and $C = \mathcal{T}^q \setminus A$ the remaining candidates. Representative samples are iteratively chosen to maximize the condition defined by a generalized graph-cut criterion:

$$G^*(A|C) = \sum_{i \in C} \sum_{j \in A} \text{Sim}(i, j) - \sum_{j_1, j_2 \in A} \text{Sim}(j_1, j_2) \quad (10)$$

It measures how much new information a candidate image adds to the selected set. Sample similarity $\text{Sim}(i, j)$ is defined as the cosine similarity of last-layer gradients.

$$\text{Sim}(i, j) = \cos(\nabla_{\theta} L_{\text{CE}}(x_i^q, \theta), \nabla_{\theta} L_{\text{CE}}(x_j^q, \theta)) \quad (11)$$

and $\nabla_{\theta} L_{\text{CE}}$ denotes the gradient of the cross-entropy loss with respect to model parameters.

QuADD implementation: Algorithm 1 outlines the end-to-end procedure for QuADD. At each iteration, real and synthetic batches are processed through the differentiable quantizer, and the DD loss is computed on quantized data. Gradients flow through the synthetic samples and quantizer, enabling their co-optimization.

By varying sample count and bit precision, QuADD empirically traces the rate-distortion performance (see Sec. 5.2.1). The general quantizer implementation integrates with different DD objectives (see Sec. 5.2.4), as well as generalizes across data domains (see Sec. 5.3).

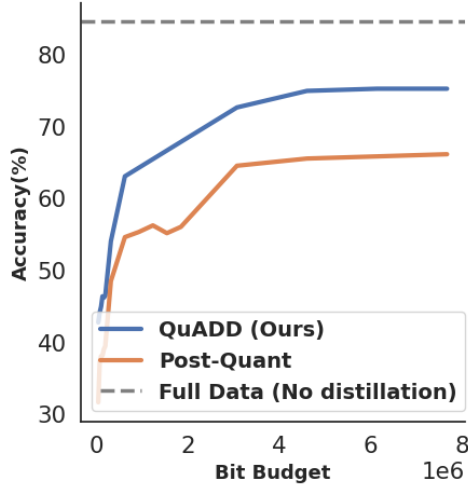


Figure 2. Rate–distortion curve. Each point represents the best achievable accuracy under a fixed transmission budget (in bits).

5. Experiments

5.1. Experimental Setup

We evaluate QuADD across both vision and tabular modalities to demonstrate its generality under fixed storage budgets, using CIFAR-10/100 [11], ImageNette [7], and a 3GPP beam management dataset (see Appendix E in the supplementary material for details). All models are trained under a fixed storage constraint as defined in eq. (3). Unless stated otherwise, full-precision baselines use 32-bit data, while QuADD employs 9-bit precision for image data (3 bits per RGB channel) and 8-bit for tabular data to ensure comparable bit budgets.

5.1.1. Benchmarks and Baselines

Image classification: We compare QuADD against: (i) full-precision dataset distillation methods trained without quantization, including DD [23], DM [28], and TM [1]; (ii) AutoPalette [26], a color-quantization approach that reduces bit depth b ; and (iii) FreD [20], a parameterization-based DD method that compresses feature dimension D .

3GPP beam management: For the tabular domain, we evaluate QuADD on a 3GPP beam management dataset, where the goal is to predict the optimal beam index given a *subset* of beam-level measurements. We compare QuADD against: (i) the full-precision original dataset, (ii) a distilled dataset trained by DATM [6] without quantization, and (iii) a coreset-selection baseline using a conditional information-gain objective (see Sec. 4.3).

5.1.2. Implementation Details

For image benchmarks, we use 1, 10, and 50 images per class (IPC) following standard DD settings. ConvNetD3

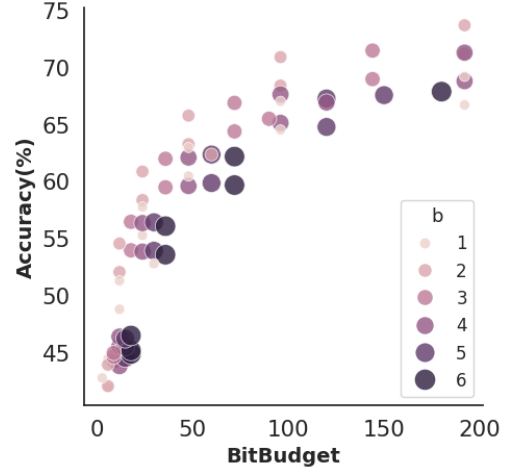


Figure 3. QuADD performance on CIFAR-10 under varying (M, b) . Lighter & smaller markers indicate lower-bit precision. The storage budget scales as $b \times 3 \times D \times M$ (constants omitted).

and ConvNetD5 serve as student and evaluation networks for CIFAR and ImageNette, respectively. We adopt TM based DD from DATM [6] without soft-label initialization. Results are averaged over five runs (see Appendix C of the supplementary material for all training details).

5.2. Results on Image Benchmarks

5.2.1. Rate-distortion Analysis

Figure 2 illustrates the rate–distortion relationship between total transmitted bits and test accuracy. The bit rate follows eq. (3), combining both the number of synthetic samples and their quantization precision, and each point reflects the best-performing (M, D) configuration under a fixed budget. Higher bit budgets naturally yield better accuracy, but QuADD delivers substantially stronger rate–distortion efficiency than post-training quantization. Across multiple budgets, QuADD improves accuracy by at least 10%, showing that quantization-aware distillation enables synthetic data to adapt to precision limits and preserve more task-relevant information.

Figure 3 illustrates the performance of QuADD on CIFAR-10 as a function of the number of samples M and bit precision b . The x-axis represents the total bit budget, and points that align vertically correspond to different (M, b) configurations that yield the same total number of transmitted bits. For each such budget, multiple feasible pairs (M, b) exist—for example, a configuration with more samples at lower precision or fewer samples at higher precision. Across these vertical groups, we observe a consistent trend: For a fixed bit budget, lower-bit quantization combined with more samples often achieves higher accuracy than using fewer high-precision samples. A clear pattern emerges: for nearly every budget, using lower-bit precision to allow more

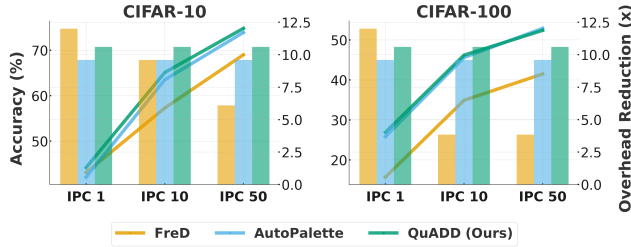


Figure 4. Accuracy and overhead reduction on CIFAR-10/100. Lines show test accuracy (left y -axis); bars show compression ratio (right y -axis) relative to the unquantized baseline.

synthetic samples yields higher accuracy than allocating the budget to high-precision samples. In contrast, using the budget for higher precision limits how many samples can be transmitted, which often yields less effective use of the available bits. Notably, most of the best results come from 2–3 bits per sub-pixel, showing that aggressively reducing precision not only preserves accuracy but often improves the rate–distortion trade-off. This highlights an advantage of QuADD: precision reduction is an effective way to maximize the informational value of transmitted synthetic data

5.2.2. Main Comparison under Fixed Storage Budget

Table 1 summarizes results across standard vision benchmarks. Coreset and unquantized DD methods, as well as FreD [20], output 32-bit full-precision values, while AutoPalette [26] and our QuADD compress data into lower-bit representations under equivalent configurations. To ensure fair comparison, all distillation methods use the same number of synthetic samples per class, i.e., the same IPC/SPC, without increasing sample count to offset quantization—as done in certain storage-budget analyses by AutoPalette and FreD. The “Overhead Reduction” section denotes the compression ratio relative to the unquantized or coreset baseline, which serves as the full-precision reference. We adjust quantization resolution such that the effective storage budgets are closely matched across methods, allowing accuracy to be compared under comparable conditions.

QuADD consistently achieves nearly the same accuracy—typically within 1%—as the full-precision distilled baseline while using less than one-tenth of the storage. For example, on CIFAR-10 with IPC=10, QuADD attains 65.1% accuracy at $10.6\times$ lower storage, closely matching the 65.5% accuracy of the unquantized baseline. Similar trends hold for CIFAR-100 and Imagenette, confirming that quantization-aware training effectively compensates for bit-level precision loss and preserves task-relevant information even under low-resolution encoding.

Figure 4 visualizes these results on CIFAR-10 and CIFAR-100. Line plots represent test accuracy (left y -axis), while bar plots show overhead reduction ratios (right y -axis). Across all IPC settings, QuADD consistently achieves

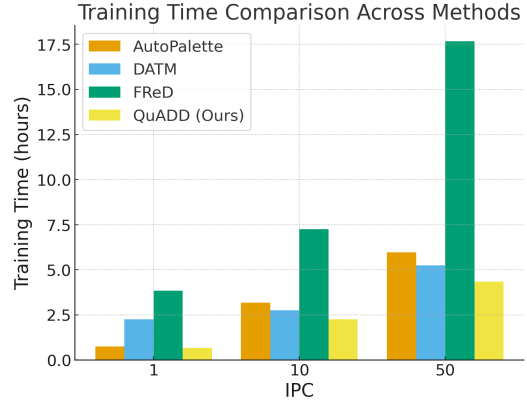


Figure 5. Training time on CIFAR-10 for DD baselines (DATM, FreD, AutoPalette, QuADD) at IPC = 1, 10, 50. Lower is better.

higher accuracy under comparable or greater compression—improving average accuracy by 1.5–3.0% at roughly $10\times$ smaller storage. Across various datasets, QuADD outperforms both AutoPalette and FreD, barring the IPC=1 instance, which highlights its enhanced storage efficiency at higher capacities. Moreover, while AutoPalette and FreD are primarily tailored for image data, the QuADD framework is modality-agnostic; this is further demonstrated by the subsequent results. in Sec. 5.3.

5.2.3. Training Efficiency

Figure 5 demonstrates that QuADD maintains competitive or superior training efficiency compared to existing distillation methods. Across all IPCs, QuADD either matches or outperforms the unquantized baseline (DATM), achieving up to 25–30% reduction in training time at medium and high IPC. In contrast, AutoPalette is slower at all scales, reflecting the additional overhead of its color-space optimization. FreD exhibits the highest cost, with training time growing rapidly with IPC, highlighting poor scalability. Overall, this confirms that QuADD’s quantization layer is simple, lightweight, and adds no extra computation.

5.2.4. Ablation and Analysis

1) Cross architecture performance: We evaluate whether QuADD’s distilled datasets transfer beyond the ConvNetD3 teacher by training AlexNet, VGG, and ResNet on the same synthetic data. As shown in Table 2, QuADD maintains strong performance across backbones and IPC settings, outperforming DATM and AutoPalette and remaining competitive with FreD (e.g., 51.8% on ResNet at IPC=10 vs. 49.0% for DATM and 50.1% for AutoPalette).

The higher accuracy gains observed by FreD are attributed to the alignment of frequency-domain representations with architectural biases and their capacity to preserve high-frequency details. However, these methods involve certain trade-offs: frequency-domain training is slower, optimiza-

Table 1. Comparison of dataset distillation on **CIFAR-10**, **CIFAR-100**, and **Imagenette** under different IPC budgets. Best result for Parameterized DD in each column is **bolded**. For AutoPalette, we quantize images to **7-bit**; for QuADD, to **9-bit (3 bits/channel)**; for FreD, we use their best reported hyperparameters. Higher value indicates better performance.

		CIFAR-10			CIFAR-100			Imagenette
		IPC 1	IPC 10	IPC 50	IPC 1	IPC 10	IPC 50	IPC 10
Total storage		983K	9.8M	49.1M	9.8M	98.3M	491M	157.2M
Coreset	Random	14.4±0.2	26.0±1.2	43.4±1.0	4.2±0.3	14.6±0.5	30.0±0.4	–
	Herding [25]	21.5±1.3	31.6±0.7	40.4±0.6	8.4±0.3	17.3±0.3	33.7±0.5	–
	K-Center [19]	23.3±0.9	36.4±0.6	48.7±0.3	8.6±0.3	20.7±0.2	33.6±0.4	–
Unquantized DD	DD [23]	–	36.8±1.2	–	–	–	–	–
	DM [29]	26.0±0.8	48.9±0.6	63.0±0.4	11.4±0.3	29.7±0.3	43.6±0.4	–
	DC [28]	28.3±0.5	44.9±0.5	53.9±0.5	12.8±0.3	25.2±0.3	–	–
	TM [1]	45.8±0.8	65.1±0.7	71.6±0.2	24.3±0.3	40.1±0.4	47.7±0.2	67.3±0.3
	DATM [6]	46.1±0.5	65.7±0.2	75.1±0.6	27.1±0.2	47.3±0.4	53.9±0.2	67.8±0.3
Parameterized DD	FreD [20]	43.1±1.5	57.3±1.1	69.0±0.8	15.7±0.8	34.9±0.6	41.5±0.5	66.2±0.3
	AutoPalette [26]	42.1±0.8	63.5±0.4	73.9±0.3	25.8±0.3	45.6±0.4	52.9±0.3	63.0±0.7
	QuADD (Ours)	44.9±0.4	65.4±0.2	74.8±0.2	26.9±0.3	46.2±0.1	52.4±0.2	67.0±0.4
Entire dataset		84.8±0.1			56.2±0.3			87.4±0.3
Overhead reduction (relative to original dataset)								
FreD		12 ×	9.6×	6.1×	12 ×	3.8×	3.8×	12×
AutoPalette		9.6×	9.6×	9.6×	9.6×	9.6×	9.6×	13.3 ×
QuADD (Ours)		10.6×	10.6 ×	10.6 ×	10.6×	10.6 ×	10.6 ×	10.6×

tion is less stable, and DCT-based parameterizations do not easily generalize to all data types.

2) Cross distillation performance: To evaluate the generality of the QuADD framework, we implement it on additional DD backbones, as shown in Table 3. Under TM, QuADD maintains high accuracy across all IPC budgets, remaining within 1–2 points of the full-precision baseline and surpassing FreD and AutoPalette at IPC1 and IPC10. In the more challenging DM setting, QuADD significantly improves over AutoPalette at low IPC and remains competitive with FreD, narrowing the gap as IPC increases

5.3. Cross-Domain Evaluation: Tabular 3GPP Data

To assess QuADD’s generality beyond vision tasks, we evaluate it on a beam management problem inspired by 3GPP wireless systems. The goal is to predict the optimal transmission beam index from a partially observed set of channel measurements representing user-side signal information like received power. These scalar, unstructured features serve to validate QuADD’s multi-domain capabilities.

Problem Description. The task involves identifying the transmit–receive beam pair with the highest received power

from a masked set of beam measurements (see Appendix E of the supplementary material for details). The received measurements are organized as a 2-D matrix, where masking prevents observation of all values. The model takes this masked matrix as input interpolates the missing values to predict the beam with the maximum reconstructed power, analogous to a matrix completion problem.

DD Setup. The true data contains 3×10^4 measurements generated from a simulated mmWave link setup. A three-layer CNN is trained on distilled datasets with M samples per class and bit precision $B \in \{4, 8, 32\}$. All methods are evaluated under a fixed storage budget ($M \times B \times D$ bits) to ensure fair comparison across full-precision, quantized-only, and quantization-aware distilled data.

Results. Figure 6 shows the trade-off between compression and accuracy on the beam management task. Bars indicate compression ratios, and the dashed line shows accuracy. A model trained on the full dataset reaches 89% accuracy. Quantizing the full dataset (8-bit) yields $4\times$ reduction with accuracy 87%, while unquantized DD gets 77% accuracy at $46\times$ reduction. QuADD achieves 81.9% at $36\times$ reduction, and 77.5% accuracy at a substantial $183\times$ reduction.

Method	AlexNet			VGG			ResNet		
	IPC1	IPC10	IPC50	IPC1	IPC10	IPC50	IPC1	IPC10	IPC50
Unquantized Baseline (DATM)	15.6±1.0	25.5±1.5	54.1±1.3	29.8±0.9	39.5±1.0	63.1±0.3	30.1±1.0	49.0±1.5	65.2±1.0
FreD	26.8±1.2	35.3±1.5	56.9±1.3	34.7±0.8	43.6±0.8	55.8±0.6	34.9±0.7	43.1±0.6	62.0±0.7
AutoPalette	19.4±2.5	32.0±1.1	56.8±0.7	28.3±1.1	39.2±1.1	61.7±0.5	29.7±2.1	50.1±1.3	61.3±0.5
QuADD (Ours)	21.4±0.7	29.7±0.7	50.3±0.8	29.8±1.0	39.6±0.7	63.5±0.5	30.7±1.2	51.8±0.5	66.6±0.4

Table 2. Accuracy comparison of different methods across AlexNet, VGG, and ResNet under varying IPC values on CIFAR-10. Best results in each column are highlighted in bold.

Method	DM			TM		
	IPC1	IPC10	IPC50	IPC1	IPC10	IPC50
<i>Vanilla (unquantized)</i>	26.0 ± 0.8	48.3 ± 0.6	63.0 ± 0.4	45.7 ± 0.8	65.2 ± 0.7	71.4 ± 0.2
FreD	33.6± 0.4	48.1± 0.8	62.0 ± 0.6	43.1± 0.3	57.3 ± 0.5	69.0 ± 0.2
AutoPalette	12.0 ± 0.4	15.3 ± 0.2	48.5 ± 0.4	42.0 ± 1.1	61.2 ± 0.2	69.5 ± 0.2
QuADD (Ours)	17.5 ± 0.5	45.1 ± 0.3	62.4± 0.3	44.7± 0.5	63.2± 0.2	70.5± 0.2

Table 3. Performance comparison across distillation methods on DM and TM datasets under varying IPC values on CIFAR-10. The unquantized baseline serves as a reference for all methods. Best results in each column are highlighted in bold.

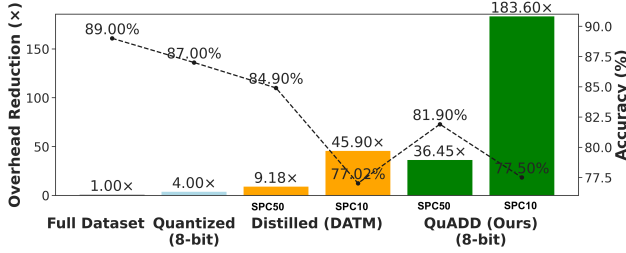


Figure 6. Overhead reduction and prediction accuracy on beam management problem. Higher is better.

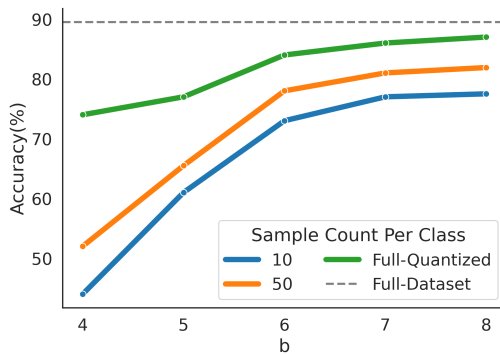


Figure 7. Accuracy-rate trade-off on the 3GPP beam management dataset for different sample counts (SPC = 10, 50, Full).

Figure 7 illustrates QuADD’s rate–distortion trade-off on the beam management dataset at different SPCs (10, 50, and full). Accuracy rises with bit precision across all scales but plateaus beyond 6 bits, indicating that moderate precision suffices to capture most task-relevant information.

6. Conclusions

This work redefines DD through the lens of information efficiency, shifting the goal *from fewer samples to fewer bits*. Unlike prior approaches that minimize sample count, we show that jointly optimizing sample count and precision yields more compact yet equally informative distilled data. We introduced **Quantization-aware Dataset Distillation (QuADD)**, a framework that integrates quantization into the distillation loop. By jointly adapting synthetic data and quantization parameters end-to-end, QuADD produces low-precision datasets that retain true data information. Its adaptive non-uniform quantization layer further aligns quantization density with data, improving representational efficiency under stringent bit budget constraints.

Experiments on image benchmarks and a 3GPP task show that QuADD delivers state-of-the-art performance under fixed bit budgets. Notably, QuADD preserves near-baseline accuracy while compressing data by over 10× on image datasets and more than 180× on 3GPP data, highlighting its effectiveness and generality beyond visual domains.

References

- [1] 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. 2, 3, 5, 7
- [2] Zhiwei Deng and Olga Russakovsky. Remember the past: Distilling datasets into addressable memories for neural networks, 2022. 1, 2
- [3] Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned step size quantization, 2020. 3
- [4] Allen Gersho and Robert M. Gray. *Vector Quantization and Signal Compression*. Springer New York, New York, NY, USA, 1 edition, 1992. 2
- [5] Ruihao Gong, Xianglong Liu, Shenghu Jiang, Tianxiang Li, Peng Hu, Jiazhen Lin, Fengwei Yu, and Junjie Yan. Differentiable soft quantization: Bridging full-precision and low-bit neural networks, 2019. 3
- [6] Ziyao Guo, Kai Wang, George Cazenavette, Hui Li, Kaipeng Zhang, and Yang You. Towards loss-less dataset distillation via difficulty-aligned trajectory matching. In *The Twelfth International Conference on Learning Representations*, 2024. 5, 7
- [7] Jeremy Howard. Imagenette: A smaller subset of 10 easily classified classes from imagenet and a little more french. <https://github.com/fastai/imagenette>, 2019. 5
- [8] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2016. 3
- [9] Sungsoo Ahn Jung, Chungman Son, Young-Joon Lee, Gunhee Choi, and Changkyu Kim. Learning to quantize deep networks by optimizing quantization intervals with task loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4350–4359, 2019. 3
- [10] 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)*, 2022. 1, 2
- [11] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Master’s thesis, Department of Computer Science, University of Toronto, 2009. 5
- [12] Shiyi Lei and Dacheng Tao. A comprehensive survey of dataset distillation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):17–32, 2024. 2
- [13] Yuhang Li, Xin Dong, and Wei Wang. Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks. In *International Conference on Learning Representations*, 2020. 4
- [14] Songhua Liu, Kai Wang, Xingyi Yang, Jingwen Ye, and Xinchao Wang. Dataset distillation via factorization. In *Advances in Neural Information Processing Systems*, pages 1100–1113, 2022. 2
- [15] Fabian Mentzer, David Minnen, Eiríkur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *The Twelfth International Conference on Learning Representations*, 2024. 4
- [16] Markus Nagel, Rana Ali Amjad, Mart van Baalen, Tijmen Blankevoort, and Max Welling. A white paper on neural network quantization. In *Proceedings of the International Conference on Machine Learning (ICML) Workshops*, 2021. 3
- [17] Julian D Parker, Anton Smirnov, Jordi Pons, CJ Carr, Zack Zukowski, Zach Evans, and Xubo Liu. Scaling transformers for low-bitrate high-quality speech coding. In *The Thirteenth International Conference on Learning Representations*, 2025. 4
- [18] Jeff M. Phillips. Coresets and sketches, 2016. 1
- [19] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach, 2018. 7
- [20] Donghyeok Shin, Seungjae Shin, and Il-chul Moon. Frequency domain-based dataset distillation. In *Advances in Neural Information Processing Systems*, pages 70033–70044. Curran Associates, Inc., 2023. 1, 2, 5, 6, 7
- [21] Iliia Sucholutsky and Matthias Schonlau. Soft-label dataset distillation and text dataset distillation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021. 2
- [22] Stefan Uhlich, Lukas Mauch, Fabien Cardinaux, Kazuki Yoshiyama, Javier Alonso Garcia, Stephen Tiedemann, Thomas Kemp, and Akira Nakamura. Mixed precision dnns: All you need is a good parametrization, 2020. 3
- [23] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. Dataset distillation, 2020. 1, 2, 5, 7
- [24] Xing Wei, Anjia Cao, Funing Yang, and Zhiheng Ma. Sparse parameterization for epitomic dataset distillation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 1, 2
- [25] Max Welling. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, page 1121–1128, New York, NY, USA, 2009. Association for Computing Machinery. 7
- [26] Bowen Yuan, Zijian Wang, Mahsa Baktashmotlagh, Yadan Luo, and Zi Huang. Color-oriented redundancy

- reduction in dataset distillation. In *Advances in Neural Information Processing Systems*, pages 53237–53260. Curran Associates, Inc., 2024. [1](#), [2](#), [4](#), [5](#), [6](#), [7](#)
- [27] Dongqing Zhang, Jia Yang, Guo-Jun Ye, Shuang Hua, and Yonggang Xu. Lq-nets: Learned quantization for highly accurate and compact deep neural networks. In *International Conference on Learning Representations (ICLR)*, 2018. [3](#)
- [28] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6514–6523, 2023. [2](#), [3](#), [5](#), [7](#)
- [29] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *International Conference on Learning Representations*, 2021. [2](#), [3](#), [7](#)
- [30] Yongchao Zhou, Ehsan Nezhadarya, and Jimmy Ba. Dataset distillation using neural feature regression. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2022. [2](#)