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

Supplementary Material

A. Baseline: Uniform Quantization

A.1. Overview

Uniform quantization serves as a baseline implementation of the quantization layer. Given bit precision b and clipping threshold α, the uniform quantizer applies evenly spaced levels after clipping:

$$Q^u(x;\alpha,b) = \text{round}\left(\frac{\lfloor x,\alpha\rceil}{\Delta}\right)\Delta, \quad \Delta = \frac{2\alpha}{2^b-1}. \quad (12)$$

Uniform quantization is widely used in deep image compression due to its simplicity and computational efficiency [24]. However, because the rounding operator is nondifferentiable with zero gradients almost everywhere, several surrogate gradient techniques are used during training.

Let $z = \lfloor x,\alpha\rceil$. The following formulations are the most common.

1. Straight-Through Estimator (STE):

Forward:

$$Q_{\text{STE}}(z) = \text{round}(z/\Delta)\Delta.$$

Backward:

$$\frac{\partial Q_{\text{STE}}}{\partial z} \approx 1$$

.

This treats quantization as the identity function during backpropagation. STE is simple but introduces an entropy estimation gap because it does not model quantization noise during training.

2. Additive Uniform Noise (AUN):

Forward:

$$Q_{\text{AUN}}(z) = z + u, \qquad u \sim \mathcal{U}\left[-\tfrac{\Delta}{2},\tfrac{\Delta}{2}\right].$$

Backward: Same as STE (gradient ≈ 1).

AUN provides a continuous relaxation of rounding and is widely used in compression models, but it introduces a discrete gap since no actual rounding occurs during training. We do not use this approach in our experiments.

3. Soft / Differentiable Rounding: Soft rounding smooths the hard rounding transition using a continuous k-shaped function, reducing gradient mismatch and improving stability [5, 16]:

$$Q_{\text{soft}}(z) \approx \text{softround}_k(z)$$

where the backward pass is governed by the derivative of the underlying hyperbolic functions (e.g., tanh). This reduces the gradient mismatch inherent to STE by smoothing the rounding boundary.

FSQ-Style Symmetric Uniform Quantizer In our experiments, we adopt a modified FSQ scalar quantizer [18] designed to maintain symmetry around the origin for any number of quantization levels $L = 2^b$.

Forward: For a scalar input x, the forward quantization operator is:

$$Q_L(x) = \frac{2}{L-1}\left\lfloor \frac{(L-1)\tanh(x)+1}{2} + \frac{1}{2}\right\rfloor - 1. \quad (13)$$

This maps $\tanh(x) \in (-1,1)$ onto a set of L uniformly spaced levels in $[-1,1]$.

Backward: A hybrid training strategy combining additive noise and STE backpropagation is used:

- Noise-based approximation (50% of training steps):

$$Q_L(x) \approx \tanh(x) + \frac{U\{-1,1\}}{L-1},$$

  where $U\{-1,1\}$ samples -1 or $+1$ uniformly.
- Straight-Through Estimator (remaining 50%): In the other half of training steps, we use the hard quantizer from Eq. 13 in the forward pass and the STE in the backward pass.

A.2. Benchmark Comparision:

Figure 8 compares several uniform-quantization baselines against our non-uniform APoT quantizer on CIFAR-10 for IPC 1 and IPC 10 across bit-widths from 1 to 6. Across both IPC settings, all uniform methods improve as bit-precision increases, but their behaviors differ substantially. Post-training quantization performs worst, particularly at low precision, highlighting the mismatch created when synthetic data are optimized in full precision and later quantized. STE improves over post-quantization but remains sensitive to rounding noise. The FSQ-based differentiable uniform quantizer achieves consistently higher accuracy than other uniform variants, confirming the value of soft rounding and hybrid training for stabilizing optimization.

APoT (non-uniform), however, achieves the strongest results overall—especially at the most aggressive precision levels—reflecting its ability to allocate more resolution near high-density regions of the synthetic data. Notably, APoT reaches or surpasses the unquantized DATM baseline with only 3–4 bits per channel, whereas uniform methods require higher precision to approach similar performance.

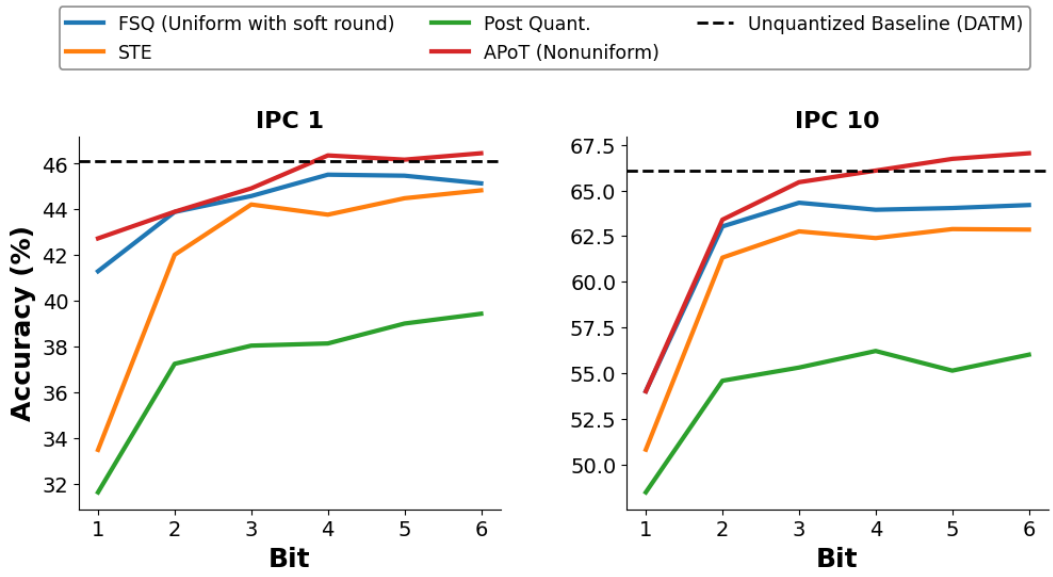

Figure 8. Benchmark comparison between uniform quantization variants and non-uniform (APoT) quantization on CIFAR-10 for IPC 1 and IPC 10. Each curve evaluates synthetic data quantized from 1 to 6 bits per channel (resulting in $b \times 3$ bits per RGB pixel). The unquantized baseline (DATM) is stored at standard 32-bit precision.

B. Nonuniform Quantization

Normalization: APoT was originally designed for model quantization, where weight tensors are approximately Gaussian. Accordingly, [14] normalizes weights to zero mean and unit variance before each clipping–projection step to stabilize training. However, this assumption does not directly transfer to dataset distillation, where the data distribution depends on the task and modality rather than following a near-Gaussian weight prior.

In image classification, synthetic images retain channel-wise structure, so we apply per-channel batch normalization before quantization to standardize the activation range and improve stability. For wireless tabular data, feature distributions are heterogeneous and often non-stationary, making unified normalization counterproductive; in such cases, we skip normalization.

Initialization of the Clipping Threshold α: The clipping threshold α determines the dynamic range of the quantizer and strongly affects quantization error. Too large α widens the quantization grid and increases projection error in the high-density central region, while too small α causes excessive clipping of informative values. Learning α directly is unstable, as noted in [14], because the data distribution shifts constantly throughout training.

To ensure faster and more reliable convergence, we initialize α using a robust percentile-based range:

$$\alpha_0 = \left|\mathrm{P}_{99}(x) - \mathrm{P}_1(x)\right|, \tag{14}$$

which removes extreme outliers while retaining most of the useful signal.

For settings that apply DCT-based transformations to the data, as in [21], we follow a two-band frequency strategy: low-frequency components and high-frequency components each receive their own learnable clipping threshold. We initialize these as:

$$\begin{aligned} \alpha_0^{\text{low}} &= \left|\mathrm{P}_{99}(x_{\text{low}}) - \mathrm{P}_1(x_{\text{low}})\right|, \\ \alpha_0^{\text{high}} &= \left|\mathrm{P}_{99}(x_{\text{high}}) - \mathrm{P}_1(x_{\text{high}})\right| \end{aligned} \tag{15}$$

reflecting the different dynamic ranges across frequency bands.

C. Experiments Details

Datasets: We conduct experiments across both image and tabular modalities:

- CIFAR-10: A 10-class image dataset containing 50,000 training and 10,000 test RGB images of size 32×32. Classes include airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.
- CIFAR-100: Similar in structure to CIFAR-10 but with 100 classes and 600 images per class (500 train, 100 test). Image resolution is also in 32×32.
- ImageNette Subset: 10-class high-resolution subsets of ImageNet ,containing 128×128 RGB images.
- 3GPP Beam Management (Wireless Tabular): A tabular dataset for wireless signal prediction, consisting of 29,784 samples split into 80% training and 20% testing. Each sample has 64 features and one of 64 beam classes.

Unlike the image datasets, the class distribution is highly imbalanced, with class sizes ranging from 14 to 284 samples.

Networks For low-resolution datasets (CIFAR-10/100), we adopt the standard ConvNetD3 architecture used in prior distillation works. The network contains 3 convolutional blocks, each with 128 filters of size 3×3, followed by an instance normalization layer, a ReLU, and an average pooling layer with 2×2 kernel and stride 2. For ImageNet subsets, we follow prior practice and use ConvNet-D5, a deeper 5-layer variant with the same block structure.

For cross-architecture evaluation, we train distilled datasets on three unseen networks: VGG11[22], AlexNet[12], and ResNet18[7]. All architectures follow their standard implementations.

For the palette network used in [29], we follow their architecture and employ a single 1×1 convolutional layer. The palette network is first warmed up for 2 epochs before being jointly optimized within the DD pipeline.

Implementation Details For APoT quantization layer, the module consists of a learnable clipping threshold α, a b-bit precision parameter, and an optional batch-normalization step applied before clipping when enabled.

QuADD is trained using Trajectory Matching (TM) by default, though it is fully compatible with other DD objectives such as Distribution Matching (DM). The hyperparameter settings for the non-uniform quantization case, as well as the configurations used for benchmark methods (AutoPalette, FReD, and DATM), are provided in Tables 4 and 5 for each dataset and framework.

For FReD [21], we used the hyperparameter settings reported in the paper.

We observe that QuADD is generally robust across hyperparameters, though several settings still require careful tuning. Many of these align with prior DD work—including the number of synthetic steps, the maximum start epoch, and the synthetic batch size.

QuADD's quantization layer itself introduces only a small number of hyperparameters, and we find that it requires minimal tuning in practice:

- Uniform quantization: choice of companding function (e.g., `tanh`, Laplace CDF) or an optional linear transform to reshape the data distribution before quantization.
- Non-uniform quantization: number of learnable clipping thresholds α, which is domain-specific. For images, we use either a single α shared across RGB channels or one per channel; for tabular data, α may be per feature or shared globally. Additional α parameters yield only marginal gains (typically $\leq 1\%$), so we adopt the minimal setting unless noted otherwise.
- Batch normalization: applied before quantization. For uniform quantization, we find it beneficial primarily when the data is not preprocessed with ZCA. For nonuniform quantization, it is applied by default in the image case.

All experiments use the SGD optimizer with standard DD training schedules. Unless otherwise specified:

- CIFAR experiments use 32×32 synthetic images at 2–6 bits per sub-pixel.
- ImageNet subset experiments use 128×128 synthetic images at 3–5 bits.
- Wireless experiments use 4-8-bit quantization.

Training was performed on 4 $\times$ NVIDIA V100 (32GB) or 2$\times$ NVIDIA H100 (80GB) GPUs.

D. QuADD Implementation for DATM

We integrate our proposed Quantization-aware Dataset Distillation (QuADD) framework with the Difficulty-Aligned Trajectory Matching (DATM) method [6], forming a quantization-aware variant referred to as **QuADD-DATM**. This integration enables simultaneous optimization of synthetic data, model parameters, and quantizer precision under fixed bit budgets.

DATM Objective. DATM optimizes a synthetic dataset $\mathcal{S}$ such that model parameters θ trained on $\mathcal{S}$ follow a trajectory that closely matches the training trajectory on the real dataset $\mathcal{T}$. The DATM loss for a model parameterized by θ_t at iteration t is expressed as

$$\begin{aligned} \mathcal{L}_{\text{DATM}}(\mathcal{S}; \theta_t) &= \sum_{t=1}^{T} \|\theta_{t+1}^{\mathcal{T}} - \theta_{t+1}^{\mathcal{S}}\|_2^2, \\ \theta_{t+1}^{\mathcal{S}} &= \theta_t - \eta \nabla_{\theta_t} \mathcal{L}_{\text{CE}}(\mathcal{S}, \theta_t), \end{aligned} \tag{16}$$

where η is the learning rate, and $\mathcal{L}_{\text{CE}}$ denotes the cross-entropy loss.

Quantization-Aware Formulation. In QuADD-DATM, we introduce a differentiable quantization layer $Q(\cdot; \alpha, b)$ into the synthetic data path, such that the quantized synthetic dataset is

$$\mathcal{S}_q = Q(\mathcal{S}; \alpha, b), \tag{17}$$

where α is the clipping threshold and b is the bit precision. The quantizer used is the APoT quantization layer, with the implementation of the forward and backward pass detailed in Sec. 4.2.2.

The QuADD-DATM objective is therefore modified as

$$\begin{aligned} \mathcal{L}_{\text{QuADD-DATM}} &= \sum_{t=1}^{T} \|\theta_{t+1}^{\mathcal{T}} - \theta_{t+1}^{\mathcal{S}_q}\|_2^2, \\ \theta_{t+1}^{\mathcal{S}_q} &= \theta_t - \eta \nabla_{\theta_t} \mathcal{L}_{\text{CE}}(\mathcal{S}_q, \theta_t). \end{aligned} \tag{18}$$

Table 4. Hyperparameters for all methods based on Trajectory Matching (TM) framework.

Dataset	IPC	Synth. BS	Synth. steps	Expert ep.	Max start ep.	Quant. LR	Img. LR	Step LR	Teacher LR	ZCA
CIFAR-10	1	200	80	2	15	0.1	500	10^{-7}	10^{-2}	True
	10	400	35	2	40	0.1	1000	10^{-5}	10^{-2}	True
	50	200	60	2	40	0.1	500	10^{-5}	10^{-2}	True
CIFAR-100	1	200	60	2	25	0.1	1000	10^{-5}	10^{-2}	True
	10	500	50	2	70	0.1	1000	10^{-5}	10^{-2}	True
	50	800	80	2	70	0.1	1000	10^{-5}	10^{-2}	True
ImageNette	10	60	40	2	20	0.1	1000	10^{-5}	10^{-2}	False

Table 5. Hyperparameters for our method based on the Distribution Matching (DM) framework.

Dataset	IPC	Synthetic batch size	LR (Quantizer)	LR (Synthetic Image)	ZCA
CIFAR-10	1	–	0.1	1	True
	10	–	0.1	1	True
	50	–	0.1	10	True
CIFAR-100	1	–	0.1	1	True
	10	–	0.1	1	True
	50	50	0.1	10	True

Algorithm 2: QuADD Implementation for DATM with APoT

Input: Real dataset $\mathcal{T}$, synthetic dataset $\mathcal{S}$, APoT quantizer $Q_{\text{APoT}}(\cdot;\alpha,b,k,n,\gamma)$, trajectory mapping $\phi(\cdot;\theta)$, learning rates $\eta_S,\eta_\alpha,\eta_\gamma,\eta_\theta$

Output: Quantized distilled dataset $\mathcal{S}^{q^*} = Q^*_{\text{APoT}}(\mathcal{S}^*)$

Initialize $\mathcal{S}$ and APoT parameters (α,γ) as in Sec. 4.2.2; fix (b,k,n)

for *each distillation iteration* $t = 1,\dots,T$ **do**

`// Mini-batch sampling`

Sample $\mathcal{B}_\mathcal{T} \sim \mathcal{T}$ and $\mathcal{B}_\mathcal{S} \sim \mathcal{S}$

`// Quantize with APoT (Sec. 4.2.2)`

$\mathcal{S}^q \leftarrow Q_{\text{APoT}}(\mathcal{B}_\mathcal{S})$

`// DATM inner loop (one step, referenced)`

Compute per-sample CE losses and logits on $\mathcal{B}_\mathcal{T}$ and $\mathcal{S}^q$ with θ_t in (16)

`// Trajectory alignment on quantized data`

Evaluate $\mathcal{L}_{\text{QuADD-DATM}}$ using (18)

`// Backprop through APoT quantizer`

Backpropagate using (6)

`// Parameter updates (keep (b,k,n) fixed)`

$\mathcal{S} \leftarrow \mathcal{S} - \eta_S \frac{\partial \mathcal{L}}{\partial \mathcal{S}}$; $\alpha \leftarrow \alpha - \eta_\alpha \frac{\partial \mathcal{L}}{\partial \alpha}$; $\gamma \leftarrow \gamma - \eta_\gamma \frac{\partial \mathcal{L}}{\partial \gamma}$

Algorithm Overview. Algorithm 2 summarizes the QuADD-DATM procedure. At each iteration, real and synthetic mini-batches are sampled, quantized through $Q(\cdot)$, and used to compute the trajectory-matching loss. Both $\mathcal{S}$ and the quantizer parameters are updated jointly, enabling adaptation to precision-induced distortion.

E. Beam-management problem for 3GPP wireless communication systems

E.1. Overview

Role of AI/ML in 3GPP wireless systems Artificial Intelligence and Machine Learning (AI/ML) have become integral components of modern wireless systems, especially within the 3GPP standardization framework. Starting from Release 17 and continuing through Releases 18–20, AI/ML methods are being systematically explored for radio access network (RAN) optimization, air-interface design, and system-level management. The goal is to improve network adaptability and efficiency under diverse environmental conditions and operational constraints. Within 3GPP, AI/ML techniques are primarily leveraged to:

- Enhance network automation via data-driven models that can dynamically learn from radio measurements and user contexts.
- Optimize signal processing tasks such as channel estimation, link adaptation, and beam selection.
- Reduce signaling and feedback overhead, by learning compact representations of high-dimensional wireless environments.
- Enable cross-layer intelligence, where data from physical, MAC, and higher layers are fused for predictive or prescriptive control.

Beam Management Problem in 3GPP Wireless Systems Beam management is a cornerstone of millimeter-wave (mmWave) and sub-THz communications, where highly di-

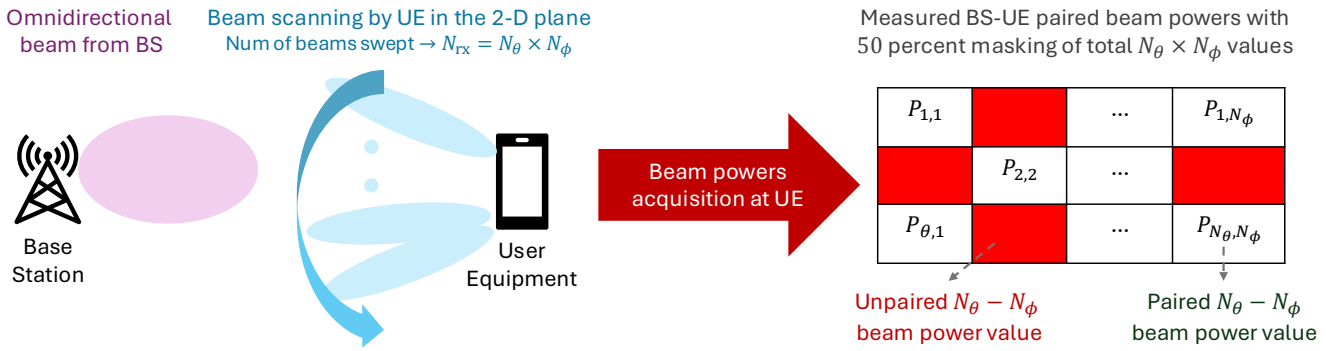

Figure 9. **Illustration of the 3GPP beam management problem under one-sided beam scanning.** The base station (BS) transmits an omnidirectional or fixed reference beam, while the user equipment (UE) performs directional scanning across azimuth (θ) and elevation (ϕ) angles. The resulting $N_\theta \times N_\phi$ grid of received beam powers represents the UE's spatial response pattern. Only a subset of these directions—here, 50%—is measured, simulating limited probing due to time or energy constraints. The masked and observed beam powers form a tabular dataset used as input to the AI/ML model, with missing entries representing unmeasured spatial directions.

rectional transmissions are necessary to overcome path loss and fading[1][2][3]. In the 3GPP framework, beam management encompasses the procedures for beam sweeping, measurement, reporting, and selection between the base station (BS) and user equipment (UE) [4][5][6].

The goal of beam management is to identify the optimal transmit–receive (Tx–Rx) beam pair that maximizes the received power or signal-to-noise ratio (SNR) at the UE. However, this search space grows quadratically with the number of Tx and Rx beams—e.g., with N_{tx} transmit beams and N_{rx} receive beams, the system must evaluate $N_{\text{tx}} \times N_{\text{rx}}$ possible beam pairs. Measuring all combinations is often infeasible due to time and energy constraints. The AI/ML formulation seeks to predict the best beam pair from a partially observed subset of measurements—thereby reducing beam sweeping overhead while maintaining near-optimal link quality.

The BM framework is portrayed in Fig. 9. Although the general beam management problem in 3GPP systems involves identifying the optimal transmit–receive (Tx–Rx) beam pair, we illustrate here a simplified one-sided scanning scenario, where the base station (BS) transmits an omnidirectional or fixed reference beam, and the user equipment (UE) performs directional scanning in both azimuth and elevation planes. The UE thus collects a set of received beam power measurements corresponding to $N_\theta \times N_\phi$, representing its local angular response to the BS transmission. Only a subset of these measurements—here, 50 % of the total entries—are available due to masking, emulating practical limitations in beam probing caused by time and energy constraints. The measured powers $P_{i,j}$ are organized in a 2-D tabular form indexed by azimuth (θ) and elevation (ϕ), where missing entries denote unobserved directions.

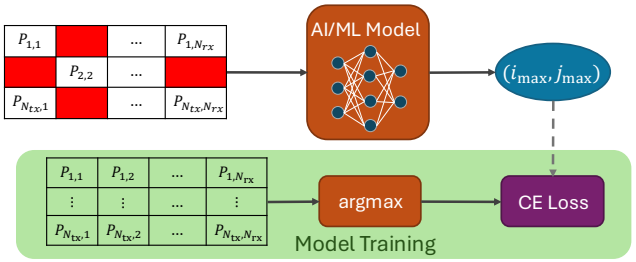

Figure 10. **AI/ML-based inference for optimal beam prediction in the 3GPP beam management problem.** The masked beam-power matrix from Fig. 9 is processed by an AI/ML model trained to nonlinearly interpolate missing measurements and predict the beam direction corresponding to the maximum reconstructed power. The model outputs the estimated index $(i_{\max}, j_{\max})$ of the optimal beam, trained using a cross-entropy loss over all possible beam indices.

An AI/ML model is used for the extraction of the best beam pairs from the observed measurements, as shown in Fig. 10. This setup can be viewed as a matrix completion

[1] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*, Cambridge University Press, 2005.

[2] H. L. Van Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*, John Wiley & Sons, 2002.

[3] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *Foundations and Trends in Signal Processing*, vol. 11, no. 3–4, pp. 154–655, 2017.

[4] 3rd Generation Partnership Project (3GPP), "Study on Artificial Intelligence (AI)/Machine Learning (ML) for NR air interface," Technical Report (TR) 38.843, V18.0.0, Dec. 2023. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/38_series/38.843/

[5] 3rd Generation Partnership Project (3GPP), "NR; Medium Access Control (MAC) protocol specification," Technical Specification (TS) 38.321, V17.7.0, Dec. 2023. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/38_series/38.321/

[6] 3rd Generation Partnership Project (3GPP), "Study on New Radio Access Technology; Physical Layer Aspects," Technical Report (TR) 38.802, V14.2.0, Sept. 2017. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/38_series/38.802/

plus classification task, where the model must infer latent spatial correlations between adjacent beams to accurately estimate the unobserved entries. Such correlations often capture the angular continuity of propagation paths, reflecting realistic 3GPP channel models.

- The AI/ML model receives the masked beam-power matrix as input.
- The network learns to nonlinearly interpolate the missing values and predict the beam index $(i_{\max}, j_{\max})$ corresponding to the maximum received power.
- The model is trained using cross-entropy loss (CE Loss) to classify the correct optimal beam among all candidates. The training data consists of complete tabular measurements with all $N_{\text{tx}} \times N_{\text{rx}}$ possible beam pairs.

Dataset distillation for beam management in 3GPP systems In wireless communication systems, dataset distillation (DD) serves a particularly vital role owing to the distributed nature of data acquisition and model training across multiple base station (BS) and user equipment (UE) nodes. Each node observes channel conditions unique to its spatial, temporal, and environmental context, leading to inherently non-identically distributed (non-IID) data. Transferring large-scale, full-precision datasets from multiple nodes to a centralized learning server—or between cooperating edge nodes—imposes substantial communication and storage overhead under constrained fronthaul and backhaul bandwidths. DD mitigates these challenges by synthesizing compact yet information-preserving datasets that capture the key statistical and structural features of local measurements. When shared across nodes, these distilled datasets significantly reduce the bit-level transfer cost while maintaining the fidelity required for downstream model training. Hence, in wireless systems, DD extends beyond mere data compression—it functions as a foundational enabler of efficient multi-node learning and collaboration, aligning with emerging 3GPP initiatives on AI/ML-driven air-interface optimization and distributed RAN intelligence.

The beam management dataset serves as a non-visual tabular benchmark for validating the generality of the proposed Quantization-aware Dataset Distillation (QuADD) framework. While visual datasets like CIFAR or ImageNette test perceptual fidelity, the 3GPP beam management data evaluates QuADD's performance in a domain characterized by structured sparsity and physical constraints. The distilled and quantized datasets aim to preserve predictive power while drastically reducing bit-level storage—reflecting practical needs in bandwidth-limited network environments.

E.2. Ablation Study: Imbalanced Dataset

Figure 11 presents an interesting case study on the imbalanced 3GPP wireless dataset—reflective of many real-world scenarios where class frequencies vary widely. At IPC,50, both sampling strategies improve with larger bit budgets, but sampling without replacement yields slightly higher Accuracy and F1-score, especially at 6–8 bits. One intuition is that sampling with replacement forces minority classes to be duplicated, which can overemphasize a few low-variation or noisy examples and reduce the effective diversity of the initialization. In contrast, using each available sample once preserves the natural variability of rare classes and avoids reinforcing redundant gradients, leading to more stable distillation under imbalance at higher fidelity.

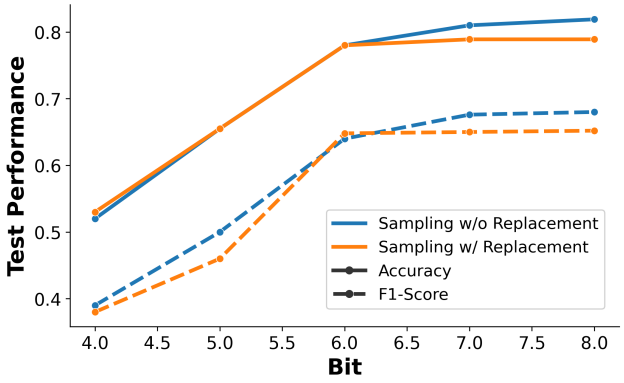

Figure 11. Effect of sampling strategy on the imbalanced 3GPP wireless dataset at IPC 50. For classes with fewer than 50 real samples, we initialize $\mathcal{S}$ either by (i) using all available samples (sampling *without* replacement), or (ii) duplicating samples to reach 50 via sampling *with* replacement.