FIDELITY FOR REHEARSAL, COVERAGE FOR CALIBRATION: ROLE-DECOUPLED MEMORY FOR CLASS-INCREMENTAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

In Class-Incremental Learning (CIL), limited replay can preserve separable features for historical classes, yet provide insufficient distribution coverage to align the classifier with the historical class distributions in feature space. We observe that representation rehearsal and classifier calibration have different input fidelity requirements: with backbone frozen, calibration can trade input detail for broader historical coverage, provided that compression retains information relevant to classification in feature space. To this end, we propose a role-decoupled memory framework: using high-fidelity exemplars for representation learning, and compressed low-fidelity samples exclusively for subsequent classifier calibration. Drawing on the concept of Average Gradient Outer Product (AGOP), we propose two novel compression schemes, AGOP-PCA and AGOP-guided transform coding, to achieve broad feature coverage and preserve information useful for classification. Experiments across diverse CIL benchmarks show consistent improvements over strong replay, compressed replay and classifier-alignment baselines.

1 INTRODUCTION

Continual learning (CL) studies how a model can acquire knowledge from a sequence of evolving data distributions without losing what it has learned before (De Lange et al., 2022; Wang et al., 2024). We focus on class-incremental learning (CIL), where new classes arrive over time and the model must ultimately discriminate among all classes seen so far without access to task identity at test time (Rebuffi et al., 2017; Zhou et al., 2024). A simple and consistently effective strategy is *replay*: retain a small subset of historical samples and interleave them with current data during subsequent training (Chaudhry et al., 2019; Rolnick et al., 2019). Replay supplies direct supervision for previous classes and substantially mitigates catastrophic forgetting, but practical memory budgets allow only a tiny fraction of the historical data to be stored. The same limited set of exemplars is therefore asked to support both continued representation learning and classification over an increasingly diverse set of past classes. This raises a basic question: *what historical information should a limited memory preserve when these two roles impose different demands on the stored data?*

Limited replay need not support representation preservation and classifier estimation equally well. Recent evidence shows that old-class features can remain substantially discriminative even when the classifier is poorly aligned with them (Lanzillotta et al., 2026), with recency bias toward newly introduced classes being a common manifestation of such classifier misalignment (Wu et al., 2019; Zhao et al., 2020). With few exemplars per class, the classifier receives only sparse support for historical class locations and within-class variation in the current feature space. Fitting those exemplars alone therefore need not determine the classifier's behavior over the full historical population. We thus view limited replay partly as a *distribution coverage* problem for classifier alignment, motivating separate historical support for representation rehearsal and classifier calibration.

To test whether broader low-fidelity coverage benefits these two roles equally, we fix the ER memory and spend the same extra-byte budget on either raw or JPEG-compressed samples, using them either in full-model training or frozen-backbone classifier fine-tuning (Figure 1). Increasing coverage through compression raises classifier-only gains from roughly 3 pp to about 15 pp, whereas full-model gains are much smaller and linear-probe gains remain around 0–1 pp. This controlled

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

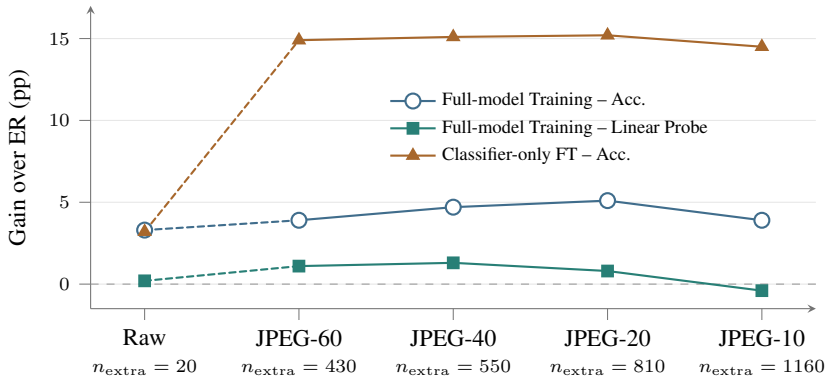

Figure 1: Effect of raw versus compressed additional replay under a fixed byte budget. The ER memory is fixed at 200 samples and the extra budget is 10% of ER storage. Raw or JPEG-compressed samples are used in full-model training or reserved for classifier-only fine-tuning. n_{extra} denotes the number of extra samples fitting the budget.

result suggests that approximate historical coverage is substantially more useful for classifier estimation than for representation learning. Rehearsal samples directly affect backbone optimization and therefore benefit from high fidelity, whereas frozen-backbone calibration can trade input detail for broader historical coverage as long as compression preserves classification-relevant information. Together, these observations motivate a role-dependent use of replay memory: we retain high-fidelity exemplars for representation rehearsal, and reserve additional compressed samples for classifier calibration.

Generic compression objectives determine what information is preserved through their distortion measures, but the resulting priorities need not match what matters for classifier calibration. For example, minimizing input-space reconstruction error can assign similar importance to directions that contribute very differently to class discrimination. Prior CL methods have addressed a related mismatch by preferentially retaining discriminative or salient image content when compressing replay memories (Luo et al., 2023; Saha & Roy, 2023). In the calibration setting considered here, we define relevance more specifically through the effect of reconstruction perturbations on separation from competing alternatives in feature space. We derive input sensitivities from prototype margins and aggregate their gradient outer products in the spirit of the Average Gradient Outer Product (AGOP) (Radhakrishnan et al., 2024). This sensitivity provides a common principle for prioritizing classifier-relevant information when allocating compression capacity, while allowing the concrete encoding mechanism to vary with the data domain and memory regime.

Our role-decoupled memory combines (i) a high-fidelity replay memory for continued representation learning and (ii) a compressed calibration memory dedicated to classifier calibration and alignment. At each stage, reconstructed calibration samples are re-embedded by the current frozen backbone before classifier-only calibration. We realize classifier-sensitive compression in two complementary ways: *AGOP-PCA*, a general low-dimensional representation with sensitivity-guided scaling and refinement, and *AGOP-guided transform coding*, which avoids AGOP-PCA's persistent decoder-state overhead by keeping sensitivity at the encoder while retaining a standard decoder. We evaluate JPEG2000 as the concrete transform-codec realization.

Our contributions are threefold:

- We identify a role-dependent fidelity–coverage trade-off in replay: representation rehearsal places stronger demands on input fidelity, whereas classifier calibration can trade fidelity for broader historical coverage.
- We introduce a role-decoupled memory framework that combines a high-fidelity replay memory with a compressed calibration-only memory, enabling historical inputs to be re-embedded under the current frozen backbone for historical feature-distribution recovery and classifier calibration.

- We develop classifier-sensitive compression based on prototype-margin input sensitivity and realize it via two complementary encoders: a general-purpose AGOP-PCA representation and an image-specific transform-coding scheme with AGOP-guided rate allocation.

2 Related Work

Compression-Based and Content-Aware Replay. Exemplar replay methods such as iCaRL and ER retain limited historical data under a fixed memory budget (Rebuffi et al., 2017; Rolnick et al., 2019). Compressed replay reduces per-sample storage to increase quantity. MRDC studies this fidelity–quantity trade-off (Wang et al., 2022). Later methods preserve discriminative content through adaptive compression, region packing/completion, or frequency-aware coding (Luo et al., 2023; Saha & Roy, 2023; Bai et al., 2023; Kim & Kim, 2025). Because these inputs are replayed during continual training, compression artifacts can affect representation updates. We instead keep high-fidelity exemplars for backbone rehearsal and reserve compressed inputs for frozen-backbone classifier calibration.

Classifier Bias and Alignment. Limited historical supervision also causes classifier bias. BiC reserves held-out data for frozen-network bias correction, while other methods use normalization or gradient reweighting (Wu et al., 2019; Zhao et al., 2020; He, 2024). Historical representations can remain discriminative even when the classifier degrades under limited replay (Lanzillotta et al., 2026), motivating alignment through class-wise statistics, semantic-shift correction, or boundary-aware synthesis (Zhang et al., 2023; He et al., 2025; Xu & Krawczyk, 2026). We recover coverage from the input side and re-embed reconstructed samples with the *current* frozen backbone, avoiding historical statistics tied to an earlier feature space.

Task-Aware Compression and Input Sensitivity. Generic image compression may not preserve task-relevant information. DeepN-JPEG and task-aware JPEG quantization adapt coding to downstream DNN objectives (Liu et al., 2018; Choi & Han, 2020), while continual-learning methods prioritize discriminative regions, saliency, or frequency importance (Luo et al., 2023; Saha & Roy, 2023; Kim & Kim, 2025). We use prototype-margin gradients as AGOP-based input sensitivity (Radhakrishnan et al., 2024; Beaglehole et al., 2024) to guide both AGOP-PCA encoding and transform-codec rate allocation for classifier calibration.

3 Method

3.1 Role-Decoupled Calibration Memory

At stage t, let $\mathcal{D}_t$ denote the currently available training data. Ordinary continual training updates the backbone F_t and classifier h_ψ using $\mathcal{D}_t$ together with a high-fidelity replay memory $\mathcal{M}_t$. While $\mathcal{D}_t$ remains available, we randomly select a class-balanced subset of samples from $\mathcal{D}_t$, and construct a compressed calibration memory $\mathcal{B}_t$ with the selected samples. $\mathcal{B}_t$ is used exclusively after ordinary training for classifier-only calibration; at calibration time, its samples are decoded and re-embedded with the then-current frozen backbone.

After the memory update, let $\widetilde{\mathcal{B}}_t$ denote the decoded labeled inputs from $\mathcal{B}_t$. With F_t frozen, we re-embed these inputs and continue optimizing only the classifier, starting from the parameters obtained at the end of ordinary training. Let $\mathcal{L}_{\mathrm{replay}}$ and $\mathcal{L}_{\mathrm{comp}}$ denote mean cross-entropy on the replay samples $\mathcal{M}_t$ and decoded calibration samples $\widetilde{\mathcal{B}}_t$, respectively. We calibrate the classifier with a mixture of $\mathcal{M}_t$ and $\widetilde{\mathcal{B}}_t$, minimizing $\mathcal{L}_{\mathrm{cal}}(\psi) = (1-\omega_{\mathrm{comp}})\mathcal{L}_{\mathrm{replay}}(\psi) + \omega_{\mathrm{comp}}\mathcal{L}_{\mathrm{comp}}(\psi)$, where $\omega_{\mathrm{comp}} \in [0,1]$ controls the contribution of compressed calibration samples. Since compressed samples never update the backbone, their representation can trade input fidelity for historical coverage while remaining useful for classifier estimation. Figure 2 illustrates the overall framework. Below, we instantiate this framework with two compression schemes that preserve classifier-relevant information in different ways.

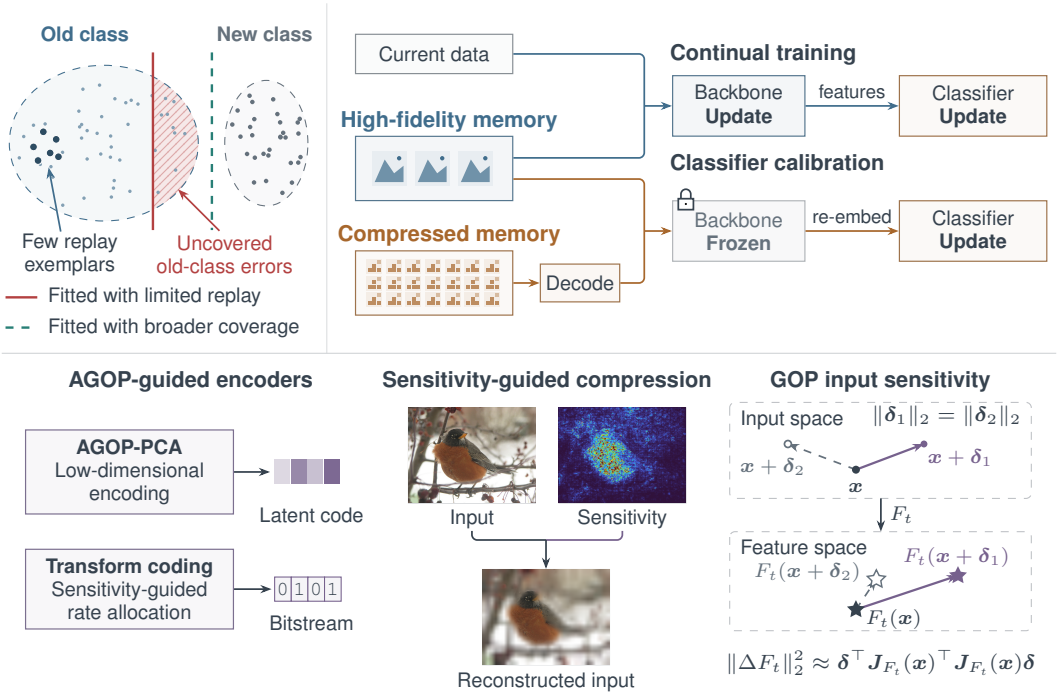

Figure 2: Role-decoupled memory. **(Top left) Historical distribution coverage gap.** Fitting on limited replay samples can leave historical data regions misclassified (hatched). **(Top right) Separate memory by its role.** High-fidelity exemplars support backbone updates, while compressed samples are used exclusively for classifier calibration. **(Bottom) Sensitivity-guided feature-preserving compression.** Prototype-margin input sensitivities guide two alternative encoders to recover classification-relevant feature distributions under compression.

3.2 Classifier-Relevant Input Sensitivity

We use Average Gradient Outer Products (AGOPs) to identify input directions that consistently influence classifier-relevant responses, providing a shared sensitivity signal for both compression schemes. The key intuition is that (i) input gradients reveal which directions most strongly affect a model response, and (ii) averaging their outer products emphasizes directions that repeatedly matter across responses and samples rather than sample-specific fluctuations.

For a vector-valued response $f(\boldsymbol{x}) \in \mathbb{R}^{d_{\text{out}}}$ with input dimension d_{in} and Jacobian $\boldsymbol{J}_f(\boldsymbol{x}) \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, its Gradient Outer Product (GOP) is $\boldsymbol{G}_f(\boldsymbol{x}) = \boldsymbol{J}_f(\boldsymbol{x})^\top \boldsymbol{J}_f(\boldsymbol{x})$, equivalently the sum of gradient outer products over response coordinates. The diagonal entries of $\boldsymbol{G}_f(\boldsymbol{x})$ measure coordinatewise squared sensitivity, while the off-diagonal entries capture co-sensitivity between input directions. Averaging GOPs over relevant inputs yields the Average Gradient Outer Product (AGOP), highlighting directions that consistently influence the responses (Radhakrishnan et al., 2024). For a small perturbation $\boldsymbol{\delta}$, $\|f(\boldsymbol{x} + \boldsymbol{\delta}) - f(\boldsymbol{x})\|_2^2 \approx \boldsymbol{\delta}^\top \boldsymbol{G}_f(\boldsymbol{x})\boldsymbol{\delta}$. Thus, with $f = F_t$, the GOP locally approximates feature-space reconstruction MSE (Appendix B).

To make the sensitivity classifier-relevant, we focus on margins along competing class-prototype directions. Let $\mathcal{C}_{\leq t}$ denote all classes observed through stage t, and let $\boldsymbol{\mu}_y$ denote the prototype of class y. We compute current-class prototypes from the available training data; historical prototypes combine the replay and decoded-calibration feature means using the same ω_{comp} as the calibration objective (Appendix B). For a competing class $c \in \mathcal{C}_{\leq t} \setminus \{y\}$, let $\boldsymbol{u}_{y,c}$ denote the unit feature-space direction from $\boldsymbol{\mu}_c$ towards $\boldsymbol{\mu}_y$. Given a sample $\boldsymbol{x}_i$ of class y_i, we define its prototype margin against class c as $m_{i,c} = F_t(\boldsymbol{x}_i)^\top \boldsymbol{u}_{y_i,c}$, i.e., the projection of its feature $F_t(\boldsymbol{x}_i) \in \mathbb{R}^{d_{\text{feat}}}$ onto the contrast direction between $\boldsymbol{\mu}_{y_i}$ and $\boldsymbol{\mu}_c$. We then compute the corresponding input gradient $\boldsymbol{g}_{i,c} = \nabla_{\boldsymbol{x}_i} m_{i,c}$, which measures the sensitivity of this margin to input perturbations.

Each margin induces a GOP $\boldsymbol{g}_{i,c}\boldsymbol{g}_{i,c}^\top$. We combine these GOPs over all other observed classes $c \in \mathcal{C}_{\le t} \setminus \{y_i\}$, assigning larger weights to classes whose prototypes are closer to that of y_i. This gives the weighted prototype-margin GOP $\sum_{c\in\mathcal{C}_{\le t}\setminus\{y_i\}} q_{y_i,c}\, \boldsymbol{g}_{i,c}\boldsymbol{g}_{i,c}^\top$, where $q_{y_i,c}$ denotes the normalized inverse-distance weight for competing class c.

This matrix admits a direct local interpretation. For a reconstruction perturbation $\boldsymbol{\delta}_i = \widetilde{\boldsymbol{x}}_i - \boldsymbol{x}_i$, first-order expansion gives $\Delta m_{i,c} \approx \boldsymbol{g}_{i,c}^\top \boldsymbol{\delta}_i$, and hence

$$\sum_{c\in\mathcal{C}_{\le t}\setminus\{y_i\}} q_{y_i,c}(\Delta m_{i,c})^2 \approx \boldsymbol{\delta}_i^\top \left(\sum_{c\in\mathcal{C}_{\le t}\setminus\{y_i\}} q_{y_i,c}\, \boldsymbol{g}_{i,c}\boldsymbol{g}_{i,c}^\top \right) \boldsymbol{\delta}_i. \tag{1}$$

Thus, the weighted prototype-margin GOP defines a local input-space metric for how reconstruction error perturbs separation from competing classes.

Since our compression schemes ultimately require coordinate-wise input importance, we retain only its diagonal, discarding cross-coordinate interactions. This gives the sensitivity vector $\mathbf{sens}_i \in \mathbb{R}^{d_{\text{in}}}$:

$$\mathbf{sens}_i = \sum_{c\in\mathcal{C}_{\le t}\setminus\{y_i\}} q_{y_i,c}\, \operatorname{diag}\big(\boldsymbol{g}_{i,c}\boldsymbol{g}_{i,c}^\top\big) = \sum_{c\in\mathcal{C}_{\le t}\setminus\{y_i\}} q_{y_i,c}\, (\boldsymbol{g}_{i,c} \odot \boldsymbol{g}_{i,c}). \tag{2}$$

We normalize and square-root the sample-wise sensitivities to obtain a sample-specific scale $\boldsymbol{s}_i$, and geometrically aggregate these scales within each class to obtain a class-shared scale $\boldsymbol{s}_y$. These two scales serve complementary roles: the sample scale $\boldsymbol{s}_i$ retains sample-specific local sensitivity variation, while the class scale $\boldsymbol{s}_y$ provides stable class-level state. Details including exact prototype-direction and weight definitions, normalization, aggregation, and the equivalent feature-Jacobian form of the metric are provided in Appendix B.

3.3 AGOP-PCA

Our first compression scheme leverages the class prototype-margin sensitivity to construct AGOP-PCA, a low-dimensional linear codec built on *Principal Component Analysis* (PCA). As one of the most representative dimensionality reduction methods, PCA operates on fixed-dimensional vectorized inputs without requiring a modality-specific encoder and provides an explicit linear projection and reconstruction. AGOP-PCA applies the class-shared scale $\boldsymbol{s}_y$ before PCA, turning ordinary input MSE into sensitivity-weighted reconstruction error that locally approximates classifier-relevant feature distortion (Section 3.2).

Given a (centered) input $\boldsymbol{x}_i$ of class y_i, a PCA basis $\boldsymbol{U}_r \in \mathbb{R}^{d_{\text{in}}\times r}$ and the class scale $\boldsymbol{s}_{y_i}$, the low-dimensional PCA code $\boldsymbol{z}_i \in \mathbb{R}^r$ can be obtained by encoding the class-scaled input $(\boldsymbol{s}_{y_i} \odot \boldsymbol{x}_i)$ as $\boldsymbol{z}_i = \boldsymbol{U}_r^\top(\boldsymbol{s}_{y_i} \odot \boldsymbol{x}_i)$. The corresponding reconstruction is obtained by decoding $\boldsymbol{z}_i$ with the same $\boldsymbol{U}_r$ and $\boldsymbol{s}_{y_i}$, yielding $\widetilde{\boldsymbol{x}}_y(\boldsymbol{z}_i) = \boldsymbol{s}_{y_i}^{-1} \odot (\boldsymbol{U}_r\boldsymbol{z}_i)$. Here $\boldsymbol{s}_{y_i}^{-1}$ denotes the element-wise reciprocal of the scale vector. We fit a shared PCA basis on the first task using the class-scaled inputs; the basis then remains fixed, allowing historical codes to remain decodable without storing task-specific bases or re-encoding unavailable inputs. New classes use their own retained $\boldsymbol{s}_y$ with the same basis. The persistent representation therefore consists of low-dimensional codes and labels together with the shared basis and one scale vector per class. Under the corresponding weighted linear reconstruction objective, $\boldsymbol{z}_i$ is the optimal code for the fixed decoder; the derivation and basis-fitting details appear in Appendix C.

The input-space weighted PCA minimizes weighted MSE, which serves only as a surrogate for preserving the feature distribution relevant to classifier calibration. We therefore refine the selected current-task codes while their original inputs remain available. Unlike sensitivity weighting, which prioritizes reconstruction fidelity using currently observed classes, latent refinement directly updates the persistent codes; we thus match in the full feature space rather than restricting refinement to the prototype-induced subspace. This allows directions that may be discriminative for future classes to remain constrained. With F_t, $\boldsymbol{U}_r$ and $\boldsymbol{s}_y$ fixed, we optimize the latent code $\boldsymbol{z}_i$ by minimizing the feature-matching loss $\mathcal{L}_{\text{match}}$. The loss is defined as a weighted combination of four terms:

$$\mathcal{L}_{\text{match}} = \lambda_{\text{feat}}\mathcal{L}_{\text{feat}} + \lambda_{\text{mean}}\mathcal{L}_{\text{mean}} + \lambda_{\text{spread}}\mathcal{L}_{\text{spread}} + \lambda_{\text{shape}}\mathcal{L}_{\text{shape}}, \tag{3}$$

where $\mathcal{L}_{\text{feat}}$ preserves sample-wise feature fidelity, $\mathcal{L}_{\text{mean}}$ matches the class feature mean, $\mathcal{L}_{\text{spread}}$ matches total within-class spread, and $\mathcal{L}_{\text{shape}}$ matches normalized covariance shape against the full

current-class training distribution. The reference statistics are computed while all current-task inputs remain available, whereas optimization changes only the selected latent codes. We store the resulting refined codes $\boldsymbol{z}_i^\star$ in the calibration memory for subsequent decoding and classifier calibration; the codes remain fixed thereafter, as the reference statistics are no longer available. Details of exact loss definitions, further weighted-PCA interpretation, latent-optimization, and memory-maintenance details are given in Appendix C.

3.4 AGOP-Guided Rate-Distortion Allocation for Transform Coding

AGOP-PCA requires persistent reconstruction-side state, including $\boldsymbol{U}_r$, the input mean, and class scales; this fixed overhead occupies a larger fraction of tighter calibration-memory budgets. Moreover, using the sample-specific scale $\boldsymbol{s}_i$ in its decoder would require storing an input-dimensional vector with every low-dimensional code (Appendix D). Transform codecs offer a complementary route: their predefined transform structure exploits image-specific spatial and frequency regularities, while sample-specific sensitivity can guide encoder-side rate allocation without learned decoder state or persistent AGOP metadata.

We consider transform codecs in which (i) each coded coefficient has a deterministic linear synthesis response and (ii) the encoder exposes coefficient- or group-level decisions with associated rate and distortion changes. Representative examples include JPEG (DCT), JPEG2000 (DWT), JPEG XL (VarDCT), and embedded wavelet coders such as EZW and SPIHT. During encoding, we form an input-coordinate importance vector $\boldsymbol{w}_i^{\mathrm{enc}} = (1-\lambda_{\mathrm{sample}})(\boldsymbol{s}_{y_i} \odot \boldsymbol{s}_{y_i}) + \lambda_{\mathrm{sample}}(\boldsymbol{s}_i \odot \boldsymbol{s}_i)$, where $\lambda_{\mathrm{sample}} \in [0,1]$ controls the contribution of the local sample-level scale relative to the stable class-level scale. Its diagonal matrix $\boldsymbol{W}_i^{\mathrm{enc}} = \mathrm{diag}(\boldsymbol{w}_i^{\mathrm{enc}})$ is the sensitivity-weighted input metric used by the encoder. Because $\boldsymbol{w}_i^{\mathrm{enc}}$ is consumed only at encode time, this route can exploit sample-specific sensitivity without increasing the stored representation.

We next map input-coordinate sensitivity to transform coefficients. For coefficient k, let $\boldsymbol{\phi}_k$ denote the reconstructed-image response to a unit coefficient error. We define ρ_k as the relative importance of errors in that coefficient under the sensitivity-weighted metric:

$$\rho_k = \frac{\boldsymbol{\phi}_k^\top \boldsymbol{W}_i^{\mathrm{enc}} \boldsymbol{\phi}_k}{\boldsymbol{\phi}_k^\top \boldsymbol{\phi}_k}. \tag{4}$$

The numerator measures how strongly the coefficient's reconstruction response overlaps sensitive input coordinates, while the denominator removes its ordinary synthesis energy. Hence $\rho_k > 1$ increases the value assigned to reducing errors in sensitive regions, and uniform input weighting gives $\rho_k = 1$.

We use ρ_k to reweight the codec's native distortion reductions. For a refinement decision, let ΔD_k denote its baseline distortion reduction on coefficient k and let $\Delta R > 0$ denote the additional coded bytes. Its sensitivity-weighted distortion benefit is $\Delta D^w = \sum_k \rho_k \Delta D_k$, and $\eta_{\mathrm{RD}}^w = \Delta D^w / \Delta R$ denotes the corresponding benefit per additional byte used for rate–distortion selection. In pixel space, this allocation preferentially reduces reconstruction error along high-sensitivity input coordinates. The codec therefore keeps its representation syntax and decoder unchanged; sensitivity only changes how the encoder values competing distortion reductions under the byte budget. The generic decision-level formulation is detailed in Appendix D.

We instantiate this formulation with JPEG2000 (Skodras et al., 2001; Taubman, 2000). Its inverse color transform and inverse DWT define the synthesis responses $\boldsymbol{\phi}_k$, while code-block coding passes provide discrete rate–distortion refinements. We weight each coefficient's PCRD distortion contribution by ρ_k and apply a shared weighted slope threshold across the encoding batch, reallocating bytes across images and code-blocks according to classifier-relevant benefit (Appendix D).

4 Experiment

4.1 Experimental setup

Datasets and protocol. We conduct CIL experiments on CIFAR-100 (Krizhevsky, 2009), TinyImageNet (Le & Yang, 2015), and ImageNet-100 (Deng et al., 2009; Tian et al., 2020). CIFAR-100 and ImageNet-100 are split into 10 tasks of 10 classes, while TinyImageNet is split into 20 tasks of

Table 1: Average incremental accuracy (A_{inc}) and final accuracy (A_{fin}) in percentage, reported as mean ± standard deviation over five independent runs. Higher is better. **Bold** and underlined values denote the best and second-best results in each column, respectively, excluding ER (Oracle CLS), which is shown separately for reference.

	CIFAR-100		TINYIMAGENET		IMAGENET-100	
METHOD	A_{inc}	A_{fin}	A_{inc}	A_{fin}	A_{inc}	A_{fin}
ER	55.54 ±0.50	34.15 ±1.40	44.33 ±0.67	27.43 ±0.32	66.68 ±0.15	49.18 ±0.76
MRDC	56.31 ±0.21	33.08 ±0.56	49.23 ±0.46	35.30 ±0.19	69.23 ±0.14	55.06 ±0.26
CIM	54.39 ±0.31	31.70 ±0.98	44.19 ±0.55	28.59 ±0.35	66.65 ±0.21	51.78 ±0.63
ER-ACE	62.99 ±0.32	49.10 ±0.61	47.94 ±0.40	34.05 ±0.11	69.27 ±0.53	55.62 ±0.40
LODE	64.05 ±0.34	49.80 ±0.44	48.73 ±0.47	34.18 ±0.50	69.79 ±0.07	55.71 ±0.28
DGR	63.83 ±0.69	47.27 ±0.73	50.16 ±0.71	33.50 ±0.65	69.70 ±0.19	56.60 ±0.17
DPCR	63.05 ±0.16	47.68 ±0.48	49.72 ±0.53	34.46 ±0.66	69.30 ±0.13	54.67 ±0.51
CEOS	64.36 ±0.38	49.63 ±1.01	50.59 ±0.21	35.79 ±0.28	70.29 ±0.08	56.71 ±0.25
AGOP-PCA (ours)	**66.16 ±0.79**	**51.36 ±0.71**	50.57 ±0.32	36.10 ±0.31	70.83 ±0.30	57.72 ±0.54
AGOP-J2K (ours)	65.33 ±0.34	50.67 ±0.53	**51.56 ±0.66**	**37.03 ±0.81**	**71.66 ±0.32**	**58.85 ±0.53**
ER (Oracle CLS)	–	59.30 ±0.15	–	42.48 ±0.07	–	63.14 ±0.09

10 classes. For each dataset, we use five shared random class orders across all methods. We use a ResNet-18 backbone trained from scratch for all experiments. All methods are implemented within a common ER-based continual-training framework.

Storage budget. We set the historical-data storage budget to the equivalent of 2,000 raw images on CIFAR-100, and 5,000 on TinyImageNet and ImageNet-100. For our role-decoupled memory, we constrain the high-fidelity rehearsal memory and compressed calibration memory to contain the same number of samples: $|\mathcal{M}_t| = |\mathcal{B}_t| = n$, where n is the largest feasible value under the common byte budget after accounting for all method-specific persistent storage overhead. Compressed calibration samples are counted by their actual byte sizes.

Baselines. We compare our methods against vanilla ER and representative CIL baselines for old–new imbalance/bias mitigation and classifier correction: ER-ACE (Caccia et al., 2022), LODE (Liang & Li, 2023), DGR (He, 2024), DPCR (He et al., 2025), and CEOS (Xu & Krawczyk, 2026). We also compare with the compressed-replay baselines MRDC (Wang et al., 2022) and CIM (Luo et al., 2023) under the same historical-data byte budget. Since some baselines inherently require retaining the model from the previous stage, we provide every method, including ours, with the same previous-model access and use a common knowledge distillation (KD) objective. The shared previous-model copy is not included in the historical-data storage budget.

Evaluation. Let A_t denote the test-set classification accuracy over all classes seen up to task t, measured at the corresponding task endpoint. We evaluate the performance of methods via two metrics: (i) *Average Incremental Accuracy* $A_{\text{inc}} = 1/T \sum_{t=1}^{T} A_t$; (ii) *Final Accuracy* $A_{\text{fin}} = A_T$.

4.2 MAIN RESULTS

We report average incremental accuracy (Avg. Acc.) and final accuracy (Final Acc.) on CIFAR-100, TinyImageNet, and ImageNet-100 in Table 1. The comparison includes vanilla ER, compressed-replay baselines, methods for old–new imbalance mitigation and classifier correction, and our two compressed calibration memory variants, AGOP-PCA and AGOP-J2K. ER (Oracle CLS) is the linear-probe result obtained by freezing the ER backbone and fitting a classifier with all training samples seen so far, and is shown separately as a reference for classifier headroom.

Table 2: Ablation of AGOP-guided sensitivity weighting and PCA latent refinement. Final accuracy (A_{fin}, %) is reported as mean ± standard deviation. For PCA, disabling both components gives ordinary PCA; for J2K, disabling AGOP uses the standard rate–distortion criterion. **Bold**/underlined: best/second-best within each encoder group.

			Final Accuracy (A_{fin})		
Encoder	AGOP	Latent FT	CIFAR-100	TinyImageNet	ImageNet-100
PCA	×	×	48.66 ±0.89	34.68 ±0.41	55.46 ±0.72
	×	✓	49.56 ±0.68	35.49 ±0.54	55.70 ±0.47
	✓	×	<u>50.91 ±0.53</u>	**36.42 ±0.37**	<u>56.68 ±0.64</u>
	✓	✓	**51.36 ±0.71**	<u>36.10 ±0.31</u>	**57.72 ±0.54**
J2K	×	–	<u>49.98 ±0.85</u>	<u>36.60 ±0.98</u>	<u>58.14 ±0.39</u>
	✓	–	**50.67 ±0.53**	**37.03 ±0.81**	**58.85 ±0.53**

Both proposed variants occupy the top two positions in final accuracy among all non-oracle methods on all three benchmarks. It is worth noting that ER exhibits a substantial gap to its oracle linear probe, indicating that its learned representation retains considerably more discriminative information than the online classifier exploits. Both variants substantially narrow this gap using compact calibration memory.

Compared with compressed-replay baselines, both AGOP-PCA and AGOP-J2K achieve higher final accuracy across all three datasets. Relative to the strongest compressed-replay baseline, the best proposed variant improves final accuracy by 18.28 pp on CIFAR-100, 1.73 pp on TinyImageNet, and 3.79 pp on ImageNet-100. Together with the controlled study in Appendix A (Figure 1), these results support using compressed historical samples as dedicated calibration memory rather than simply enlarging conventional replay.

The gains also persist over the stronger bias-mitigation and classifier-correction baselines. AGOP-PCA achieves the best final accuracy on CIFAR-100, exceeding LODE by 1.56 pp, while AGOP-J2K achieves the best final accuracy on TinyImageNet and ImageNet-100, exceeding CEOS by 1.24 and 2.14 pp, respectively. Average incremental accuracy follows the same overall trend: AGOP-PCA improves over CEOS by 1.80 pp on CIFAR-100, while AGOP-J2K improves by 0.97 and 1.37 pp on TinyImageNet and ImageNet-100, respectively. On TinyImageNet, AGOP-PCA essentially matches CEOS in average incremental accuracy (within 0.02 pp) while achieving higher final accuracy. Overall, the two distinct encoding strategies consistently complement high-fidelity replay with compact calibration memory under limited storage.

4.3 Ablation Studies

Component contributions. Table 2 isolates AGOP-guided sensitivity weighting and, for PCA, latent refinement toward the current class feature distribution. AGOP is the more consistent contributor: relative to its non-sensitivity-weighted counterpart, it improves final accuracy by 1.22–2.25 pp for PCA and 0.43–0.71 pp for J2K across all three datasets. That the gain transfers from a generic linear representation to an image-specific transform codec indicates that it is not tied to a particular encoder, supporting prototype-margin sensitivity as the shared compression criterion. Latent refinement gives smaller, dataset-dependent gains: it improves ordinary PCA on all datasets and further improves AGOP-PCA on CIFAR-100 and ImageNet-100, while slightly reducing the TinyImageNet mean. Its benefit is therefore complementary rather than uniformly additive, consistent with refinement correcting residual distribution mismatch after weighted reconstruction.

Feature-distribution coverage. We test whether compressed calibration samples recover historical support in the final feature space. Using the final frozen backbone, Table 3 compares the full CIFAR-100 training distribution with the historical support represented by memory: high-fidelity replay for ER, and replay plus decoded compressed samples for our methods. Both AGOP-PCA and AGOP-J2K substantially reduce the mean/covariance discrepancies and MMD^2 relative to ER;

Table 3: Feature-distribution discrepancy on CIFAR-100 between the full training set and the historical support represented by memory. Δ_μ and Δ_Σ denote mean and covariance discrepancies; barred quantities are scale-normalized. Lower is better.

	Raw Discrepancy		Normalized Discrepancy		
Method	Δ_μ	Δ_Σ	$\bar{\Delta}_\mu$	$\bar{\Delta}_\Sigma$	MMD^2
ER	75.257	1251.055	1.747	0.740	0.411
AGOP-PCA	8.701	347.967	0.201	0.211	0.025
AGOP-J2K	20.188	437.317	0.575	0.481	0.074

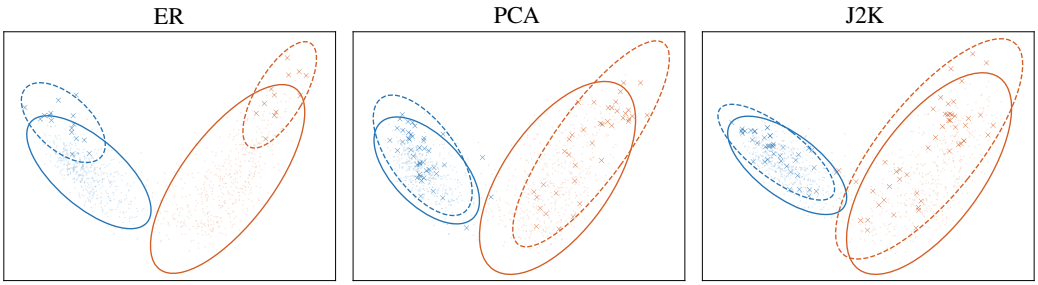

Figure 3: Feature-distribution coverage on CIFAR-100 for ER, AGOP-PCA, and AGOP-J2K. Representative classes are projected to two dimensions with PCA. Dots denote full training-set features and crosses denote historical support represented by memory; solid and dashed ellipses summarize the corresponding full-data and memory-support distributions.

AGOP-PCA gives the smallest discrepancy under every measure. Relative to ER, AGOP-PCA reduces the normalized mean and covariance discrepancies by 88.5% and 71.5%, respectively, and MMD^2 by 93.9%; the corresponding reductions for AGOP-J2K are 67.1%, 35.0%, and 81.9%. Figure 3 provides the complementary geometric view: compressed samples expand the historical support toward the location and spread of the full class distribution. Together, these results support the intended use of compressed memory as broader support for classifier calibration. Metric definitions and further analysis are provided in Appendix G.

Replay–calibration allocation. Under the same fixed byte budget, varying the calibration-to-replay sample ratio $r_{\mathrm{cal}} = |\mathcal{B}_t|/|\mathcal{M}_t|$ on CIFAR-100 shows a stable region around balanced sample counts, with the highest observed accuracy at $r_{\mathrm{cal}} = 1$ for both encoders. Performance degrades when compressed calibration memory is either severely under-provisioned or allowed to displace too much high-fidelity replay, supporting their complementary roles in historical coverage and representation rehearsal, respectively. Appendix F provides the full ablation and analysis.

5 Conclusion

We study replay memory in class-incremental learning and show that compressed samples provide broader historical coverage that is more useful for classifier estimation than representation learning, revealing a role-dependent fidelity–coverage trade-off. We therefore introduce a role-decoupled memory framework that combines high-fidelity replay for representation learning with compressed memory for classifier calibration. We instantiate it with two classifier-sensitive compression schemes, AGOP-PCA and AGOP-guided transform coding, based on prototype-margin input sensitivity. Across CIFAR-100, TinyImageNet, and ImageNet-100, both variants consistently improve final accuracy over representative replay, compressed-replay, and classifier-alignment baselines under comparable persistent-storage budgets. These results support allocating historical memory by learning role: high fidelity for representation rehearsal and compact storage for broader classifier coverage.

AI USE STATEMENT

In this work, we used generative AI tools (i) to aid or polish writing, and (ii) for research ideation or execution. We used ChatGPT to assist in paper writing and to help verify the correctness of mathematical derivations and proofs. We also used OpenAI Codex to write some of the code.

We have reviewed all AI-assisted work. We have carefully reviewed the final version of the paper to ensure that its content is accurate, faithfully reflects our work, and is free from plagiarism/hallucination. We have carefully reviewed and verified all LLM-generated code to make sure that (i) it passed all hand-crafted tests; (ii) it faithfully reflects our design and is consistent with the paper. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have submitted the relevant code in the supplementary materials, including the implementations of the role-decoupled memory framework, the prototype-margin sensitivty, the AGOP-PCA and AGOP-J2K algorithms, and other components necessary to reproduce our work.

We have described the key components of our method in the main paper. For more detailed information on the method and implementation, please refer to the relevant sections in the Appendix.

We have described the experimental setups that are crucial to reproducibility in the main paper. For more detailed information on the experiments, please refer to Appendix E.

REFERENCES

Guangji Bai, Chen Ling, Yuyang Gao, and Liang Zhao. Saliency-augmented memory completion for continual learning. In *Proceedings of the SIAM International Conference on Data Mining (SDM'23)*, pp. 244–252. SIAM, 2023.

Daniel Beaglehole, Peter Súkeník, Marco Mondelli, and Mikhail Belkin. Average gradient outer product as a mechanism for deep neural collapse. In *Advances in Neural Information Processing Systems 37 (NeurIPS'24)*, pp. 130764–130796, Red Hook, NY, USA, 2024. Curran Associates Inc.

Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New Insights on Reducing Abrupt Representation Change in Online Continual Learning. In *10th International Conference on Learning Representations (ICLR'22)*, 2022.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K. Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. On Tiny Episodic Memories in Continual Learning. *arXiv preprint*, 2019.

Jinyoung Choi and Bohyung Han. Task-Aware Quantization Network for JPEG Image Compression. In *European Conference on Computer Vision (ECCV'20)*, pp. 309–324, Berlin, Heidelberg, 2020. Springer-Verlag.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A Continual Learning Survey: Defying Forgetting in Classification Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*, pp. 248–255, 2009.

Jiangpeng He. Gradient Reweighting: Towards Imbalanced Class-Incremental Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'24)*, pp. 16668–16677, 2024.

Run He, Di Fang, Yicheng Xu, Yawen Cui, Ming Li, Cen Chen, Ziqian Zeng, and Huiping Zhuang. Semantic Shift Estimation via Dual-Projection and Classifier Reconstruction for Exemplar-Free Class-Incremental Learning. In *Proceedings of the 42nd International Conference on Machine Learning (ICML'25)*, pp. 22392–22406. PMLR, 2025.

Junsu Kim and Suhyun Kim. Salient Frequency-aware Exemplar Compression for Resource-constrained Online Continual Learning. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence (AAAI'25)*, pp. 17895–17903, 2025.

Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25 (NIPS'12)*. Curran Associates, Inc., 2012.

Giulia Lanzillotta, Damiano Meier, and Thomas Hofmann. Heads collapse, features stay: Why Replay needs big buffers. In *14th International Conference on Learning Representations (ICLR'26)*, 2026.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Yan-Shuo Liang and Wu-Jun Li. Loss Decoupling for Task-Agnostic Continual Learning. In *Advances in Neural Information Processing Systems 36 (NeurIPS'23)*, pp. 11151–11167. Curran Associates, Inc., 2023.

Zihao Liu, Tao Liu, Wujie Wen, Lei Jiang, Jie Xu, Yanzhi Wang, and Gang Quan. DeepN-JPEG: A Deep Neural Network Favorable JPEG-based Image Compression Framework. In *55th ACM/ESDA/IEEE Design Automation Conference (DAC'18)*, pp. 1–6, 2018.

Zilin Luo, Yaoyao Liu, Bernt Schiele, and Qianru Sun. Class-Incremental Exemplar Compression for Class-Incremental Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'23)*, pp. 11371–11380, 2023.

Adityanarayanan Radhakrishnan, Daniel Beaglehole, Parthe Pandit, and Mikhail Belkin. Mechanism for feature learning in neural networks and backpropagation-free machine learning models. *Science*, 383(6690):1461–1467, 2024.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental Classifier and Representation Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*, pp. 5533–5542, 2017.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience Replay for Continual Learning. In *Advances in Neural Information Processing Systems 32 (NeurIPS'19)*. Curran Associates, Inc., 2019.

Gobinda Saha and Kaushik Roy. Saliency Guided Experience Packing for Replay in Continual Learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV'23)*, pp. 5262–5272, 2023.

A. Skodras, C. Christopoulos, and T. Ebrahimi. The JPEG 2000 still image compression standard. *IEEE Signal Processing Magazine*, 18(5):36–58, 2001.

D. Taubman. High performance scalable image compression with EBCOT. *IEEE Transactions on Image Processing*, 9(7):1158–1170, 2000.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive Multiview Coding. In *European Conference on Computer Vision (ECCV'20)*, pp. 776–794, Berlin, Heidelberg, 2020. Springer-Verlag.

Liyuan Wang, Xingxing Zhang, Kuo Yang, Longhui Yu, Chongxuan Li, Lanqing Hong, Shifeng Zhang, Zhenguo Li, Yi Zhong, and Jun Zhu. Memory Replay with Data Compression for Continual Learning. In *10th International Conference on Learning Representations (ICLR'22)*, 2022.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A Comprehensive Survey of Continual Learning: Theory, Method and Application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5362–5383, 2024.

Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large Scale Incremental Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'19)*, pp. 374–382, 2019.

Hongye Xu and Bartosz Krawczyk. Revisiting Prototype Rehearsal for Exemplar-Free Continual Learning: Manifold-Aware Boundary Sampling with Adaptive Class-Balanced Loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'26) Findings*, pp. 7852–7861, 2026.

Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. SLCA: Slow Learner with Classifier Alignment for Continual Learning on a Pre-trained Model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV'23)*, pp. 19091–19101, 2023.

Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining Discrimination and Fairness in Class Incremental Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'20)*, pp. 13208–13217, 2020.

Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Class-Incremental Learning: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9851–9873, 2024.

APPENDIX

A ADDITIONAL MOTIVATION ANALYSIS

Population and replay statistics. For an old class c, let $\mathcal{D}_c$ denote its historical population and $\mathcal{M}_c \subset \mathcal{D}_c$ the replay exemplars. Under the current feature extractor F_t, define

$$\boldsymbol{\mu}_c^t = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_c}[F_t(\boldsymbol{x})], \qquad \boldsymbol{\Sigma}_c^t = \mathrm{Cov}_{\boldsymbol{x}\sim\mathcal{D}_c}[F_t(\boldsymbol{x})],$$
$$\widehat{\boldsymbol{\mu}}_c^t = \frac{1}{|\mathcal{M}_c|}\sum_{\boldsymbol{x}\in\mathcal{M}_c} F_t(\boldsymbol{x}), \quad \widehat{\boldsymbol{\Sigma}}_c^t = \mathrm{Cov}_{\boldsymbol{x}\in\mathcal{M}_c}[F_t(\boldsymbol{x})].$$

Under limited replay, the replay statistics can differ substantially from their population counterparts. In particular, for $n_c = |\mathcal{M}_c|$ samples in a d_{feat}-dimensional feature space, $\mathrm{rank}(\widehat{\boldsymbol{\Sigma}}_c^t) \le \min\{d_{\mathrm{feat}}, n_c - 1\}$, so the replay set directly represents within-class variation only within a restricted empirical subspace.

Replay-equivalent classifiers outside the replay affine span. The same limitation can be stated directly for a linear classifier. Let $\mathcal{A} = \mathrm{aff}\{\boldsymbol{f}_i\}$ be the affine span of the stored historical features $\boldsymbol{f}_i = F_t(\boldsymbol{x}_i)$. If $\mathcal{A}$ is a proper affine subspace of $\mathbb{R}^{d_{\mathrm{feat}}}$, there exist a nonzero vector $\boldsymbol{v}_\perp$ and scalar β such that $\boldsymbol{v}_\perp^\top \boldsymbol{f} = \beta$ for all $\boldsymbol{f} \in \mathcal{A}$. For any class logit $\boldsymbol{w}_k^\top \boldsymbol{f} + b_k$, the perturbation

$$\boldsymbol{w}_k' = \boldsymbol{w}_k + \gamma \boldsymbol{v}_\perp, \qquad b_k' = b_k - \gamma\beta$$

leaves that logit unchanged for every replay feature in $\mathcal{A}$, while changing it by $\gamma(\boldsymbol{v}_\perp^\top \boldsymbol{f} - \beta)$ for features outside $\mathcal{A}$. Thus, matching classifier behavior on the replay support does not uniquely determine behavior over the full historical feature distribution. New-class data and regularization can select among such replay-equivalent solutions, but do not by themselves guarantee alignment with the historical population. This argument illustrates one concrete classifier-side information bottleneck under limited replay; it is not intended as a complete explanation of recency bias or classifier degradation.

Controlled fidelity–coverage study. To isolate the effect of broader approximate coverage from our proposed compression mechanisms, we fix the ER memory and use the same extra-byte budget to store either raw inputs or off-the-shelf JPEG-compressed inputs. We then use the additional samples either during full-model training or only for frozen-backbone classifier fine-tuning. Figure 1 visualizes the resulting fidelity–coverage trade-off discussed in the Introduction.

B CLASSIFIER-RELEVANT SENSITIVITY DETAILS

Prototype estimation and geometry. Let $\mathcal{C}_t$ and $\mathcal{C}_{<t}$ denote the current and historical classes, respectively, and $\mathcal{C}_{\le t} = \mathcal{C}_t \cup \mathcal{C}_{<t}$. For a current class $y \in \mathcal{C}_t$, its prototype $\boldsymbol{\mu}_y$ is the mean feature over the available current-task samples $\mathcal{D}_{t,y}$. For a historical class $y \in \mathcal{C}_{<t}$, let $\boldsymbol{\mu}_y^{\mathrm{rep}}$ and $\boldsymbol{\mu}_y^{\mathrm{comp}}$ denote the mean features, under the current backbone, of its replay samples and decoded calibration samples, respectively. Since the original historical data are unavailable, we use the calibration-weighted estimate

$$\boldsymbol{\mu}_y = (1-\omega_{\mathrm{comp}})\boldsymbol{\mu}_y^{\mathrm{rep}} + \omega_{\mathrm{comp}}\boldsymbol{\mu}_y^{\mathrm{comp}}, \qquad y \in \mathcal{C}_{<t},$$

using the same mixture weight as the classifier-calibration objective. This makes the historical prototype reflect the relative contributions of the two memory sources used during classifier optimization.

For each class y, we treat every other observed class $c \in \mathcal{C}_{\le t} \setminus \{y\}$ as a competing prototype. We define the prototype distance $d_{y,c}$, unit contrast direction $\boldsymbol{u}_{y,c}$, normalized inverse-distance weight $q_{y,c}$, and projected margin $m_{i,c}$ of a sample $\boldsymbol{x}_i$ with label y_i as

$$d_{y,c} = \|\boldsymbol{\mu}_y - \boldsymbol{\mu}_c\|_2, \qquad \boldsymbol{u}_{y,c} = \frac{\boldsymbol{\mu}_y - \boldsymbol{\mu}_c}{d_{y,c}},$$
$$q_{y,c} = \frac{d_{y,c}^{-1}}{\sum_{c'\in\mathcal{C}_{\le t}\setminus\{y\}} d_{y,c'}^{-1}}, \qquad m_{i,c} = F_t(\boldsymbol{x}_i)^\top \boldsymbol{u}_{y_i,c}.$$

For non-zero prototype distances, $\sum_{c\in\mathcal{C}_{\le t}\setminus\{y\}} q_{y,c} = 1$, so closer prototype contrasts receive greater weights. As $\boldsymbol{g}_{i,c} = \boldsymbol{J}_{F_t}(\boldsymbol{x}_i)^\top \boldsymbol{u}_{y_i,c}$, the weighted prototype-margin GOP in Eq. (1) can equivalently be written as

$$\sum_{c\in\mathcal{C}_{\le t}\setminus\{y_i\}} q_{y_i,c}\, \boldsymbol{g}_{i,c}\boldsymbol{g}_{i,c}^\top = \boldsymbol{J}_{F_t}(\boldsymbol{x}_i)^\top \left(\sum_{c\in\mathcal{C}_{\le t}\setminus\{y_i\}} q_{y_i,c}\, \boldsymbol{u}_{y_i,c}\boldsymbol{u}_{y_i,c}^\top \right) \boldsymbol{J}_{F_t}(\boldsymbol{x}_i).$$

Hence, the metric retains local feature changes along prototype-contrast directions against all observed competing classes, with nearby prototypes emphasized through distance weights. The main text gives the corresponding Taylor interpretation and its coordinate-wise diagonal approximation.

Sample normalization and class aggregation. The diagonal sensitivity $\mathbf{sens}_i$ in Eq. (2) preserves per-coordinate importance but also carries a sample-specific overall magnitude. We remove that magnitude by normalizing its coordinate mean and taking the element-wise square root:

$$\boldsymbol{s}_i = \sqrt{\frac{\mathbf{sens}_i}{d_{\text{in}}^{-1}\mathbf{1}^\top \mathbf{sens}_i}},$$

so $(\boldsymbol{s}_i \odot \boldsymbol{s}_i)$ has unit coordinate mean. For persistent class-level scaling, we aggregate the normalized sample scales coordinatewise in log space and renormalize the result to unit root-mean-square magnitude:

$$\bar{\boldsymbol{s}}_y = \exp\left(\frac{1}{|\mathcal{D}_{t,y}|} \sum_{i:y_i=y} \log \boldsymbol{s}_i \right), \qquad \boldsymbol{s}_y = \frac{\bar{\boldsymbol{s}}_y}{\sqrt{d_{\text{in}}^{-1}\|\bar{\boldsymbol{s}}_y\|_2^2}},$$

where $\mathcal{D}_{t,y} \subseteq \mathcal{D}_t$ denotes the available current-task samples of class y. This geometric aggregation preserves the multiplicative relative structure of the normalized sample scales. The class scale is computed when the class first arrives and then retained for decoding historical codes.

C AGOP-PCA Derivation and Details

Weighted linear reconstruction. For a centered input $\boldsymbol{x}$ of class y, AGOP-PCA measures reconstruction in the class-scaled coordinates defined by $\boldsymbol{s}_y$. With decoder $\widetilde{\boldsymbol{x}}_y(\boldsymbol{z}) = \boldsymbol{s}_y^{-1} \odot (\boldsymbol{U}_r\boldsymbol{z})$, where $\boldsymbol{s}_y^{-1}$ is the element-wise reciprocal and $\boldsymbol{U}_r^\top\boldsymbol{U}_r = \boldsymbol{I}$, the weighted input distortion is

$$\|\boldsymbol{s}_y \odot (\boldsymbol{x} - \widetilde{\boldsymbol{x}}_y(\boldsymbol{z}))\|_2^2 = \|(\boldsymbol{s}_y \odot \boldsymbol{x}) - \boldsymbol{U}_r\boldsymbol{z}\|_2^2.$$

For fixed $\boldsymbol{U}_r$, the least-squares minimizer is therefore $\boldsymbol{z} = \boldsymbol{U}_r^\top(\boldsymbol{s}_y \odot \boldsymbol{x})$. On Task 1, stacking the scaled inputs $\boldsymbol{s}_{y_i} \odot \boldsymbol{x}_i$ and taking the leading r right singular vectors minimizes the summed weighted reconstruction error over an orthonormal rank-r subspace. A class-specific basis would require $\mathcal{O}(d_{\text{in}}r)$ persistent values per class. We instead keep $\boldsymbol{U}_r$ fixed afterward so stored historical codes remain decodable without retaining multiple task-specific bases or reconstructing unavailable original inputs. Later classes contribute their own $\boldsymbol{s}_y$ and therefore use class-dependent decoding directions $\mathrm{diag}(\boldsymbol{s}_y)^{-1}\boldsymbol{U}_r$ without refitting the shared basis.

Exact latent-refinement objective. We deliberately define refinement in the full feature space rather than only over the prototype-margin directions induced by classes observed so far. Directions outside the current prototype geometry can still become discriminative when future classes arrive. This concern is more acute for latent refinement than for the sensitivity weighting used by AGOP-PCA: the latter prioritizes where reconstruction fidelity is allocated, whereas refinement directly updates the persistent latent codes and could otherwise improve current-class statistics by drifting along feature directions left unconstrained by the current classifier. Full-space matching therefore provides a future-agnostic constraint on refinement.

Let $\mathcal{C}_t$ denote the classes in the current task, let $\mathcal{D}_{t,y}$ denote all currently available training inputs of class y, and let $\mathcal{I}_t$ index the n_{sel} selected current-task samples stored in calibration memory. For class y, let $(\widehat{\boldsymbol{\mu}}_y, \widehat{\boldsymbol{\Sigma}}_y)$ denote the mean and unbiased covariance of reconstructed features $F_t(\widetilde{\boldsymbol{x}}_y(\boldsymbol{z}_i))$,

and let $(\boldsymbol{\mu}_y^\star, \boldsymbol{\Sigma}_y^\star)$ be the corresponding statistics over $\mathcal{D}_{t,y}$. The components of Eq. (3) are

$$\mathcal{L}_{\text{feat}} = \frac{1}{n_{\text{sel}} d_{\text{feat}}} \sum_{i \in \mathcal{I}_t} \|F_t(\widetilde{\boldsymbol{x}}_{y_i}(\boldsymbol{z}_i)) - F_t(\boldsymbol{x}_i)\|_2^2,$$

$$\mathcal{L}_{\text{mean}} = \frac{1}{|\mathcal{C}_t|} \sum_{y \in \mathcal{C}_t} \|\widehat{\boldsymbol{\mu}}_y - \boldsymbol{\mu}_y^\star\|_2,$$

$$\mathcal{L}_{\text{spread}} = \frac{1}{|\mathcal{C}_t|} \sum_{y \in \mathcal{C}_t} \left| \frac{\operatorname{tr} \widehat{\boldsymbol{\Sigma}}_y}{\operatorname{tr} \boldsymbol{\Sigma}_y^\star + \varepsilon} - 1 \right|,$$

$$\mathcal{L}_{\text{shape}} = \frac{1}{|\mathcal{C}_t|} \sum_{y \in \mathcal{C}_t} \left\| \frac{\widehat{\boldsymbol{\Sigma}}_y}{\operatorname{tr} \widehat{\boldsymbol{\Sigma}}_y + \varepsilon} - \frac{\boldsymbol{\Sigma}_y^\star}{\operatorname{tr} \boldsymbol{\Sigma}_y^\star + \varepsilon} \right\|_F.$$

Here $\varepsilon > 0$ stabilizes covariance normalization. The first term is averaged over both selected samples and feature dimensions, while the three distribution-level terms are averaged over current classes.

Latent optimization and memory maintenance. Refinement starts from the PCA projection $\boldsymbol{z}_i$ and updates only the selected current-task latent codes; the backbone, classifier, basis, and class scales remain fixed. After optimization, $\boldsymbol{z}_i^\star$ replaces $\boldsymbol{z}_i$ in calibration memory, and the fixed decoder constrains refined reconstructions to the class-scaled subspace represented by the shared basis.

Calibration memory uses approximately balanced per-class quotas. Current-task entries are sampled uniformly without replacement within each class; when quotas shrink, historical classes are reduced only by subsampling their stored latent entries, without original-image access, re-encoding, or re-refinement. Optimizer, clipping, and serialized-byte accounting details are reported with the experimental protocol.

D AGOP-Guided Transform Coding and JPEG2000 Instantiation

Why encoder-side sensitivity. For input dimension d_{in}, retained PCA dimension r, and C classes, AGOP-PCA's shared basis, mean, and class scales contain $d_{\text{in}}(r + C + 1)$ stored values. Let b_s and b_x denote bytes per stored state value and raw input coordinate, respectively. Under a calibration-memory budget equal to a fraction α of the raw storage of n_{img} inputs, this state alone occupies the fraction

$$\frac{b_s}{b_x} \cdot \frac{r + C + 1}{\alpha n_{\text{img}}}$$

of the budget. Thus the fixed reconstruction-side overhead becomes increasingly important as α decreases; exact data types and serialized-byte accounting are reported with the experimental memory accounting. Storing a distinct input-dimensional scale for every PCA code would add another d_{in} values per sample. Transform coding instead consumes the class- and sample-level scales $\boldsymbol{s}_y$ and $\boldsymbol{s}_i$ only during encoding, allowing sample-specific sensitivity without carrying AGOP scales into the persistent representation.

Generic synthesis-response weighting. For one image, suppress the sample index and write $\boldsymbol{W}^{\text{enc}} = \boldsymbol{W}_i^{\text{enc}}$. Let the transform coefficient error vector be $\delta\boldsymbol{z}$ and let $\boldsymbol{\phi}_k$ denote the reconstructed-image response to a unit error in coefficient k. The exact image error is $\boldsymbol{e} = \sum_k \boldsymbol{\phi}_k \delta z_k$, so under the diagonal input metric $\boldsymbol{W}^{\text{enc}}$,

$$\boldsymbol{e}^\top \boldsymbol{W}^{\text{enc}} \boldsymbol{e} = \sum_{k,k'} \delta z_k \delta z_{k'} \boldsymbol{\phi}_k^\top \boldsymbol{W}^{\text{enc}} \boldsymbol{\phi}_{k'}.$$

A codec's ordinary squared-error distortion uses the analogous expression with $\boldsymbol{W}^{\text{enc}} = \boldsymbol{I}$. Our coefficient-diagonal surrogate drops cross-coefficient terms and rescales each codec-native coefficient distortion by the relative factor in Eq. (4). This keeps the codec's own quantization- and transform-dependent distortion model while changing the value assigned to errors whose synthesis responses fall on classifier-sensitive input coordinates.

More generally, let refinement decision ξ incur $\Delta R_\xi > 0$ bytes and yield baseline distortion reductions $\Delta D_{k,\xi} \geq 0$ on affected coefficients. The sensitivity-weighted benefit and slope are

$$\Delta D^w_\xi = \sum_k \rho_k \Delta D_{k,\xi}, \qquad \eta^w_{\mathrm{RD},\xi} = \frac{\Delta D^w_\xi}{\Delta R_\xi},$$

This is the generic decision-level sensitivity-weighted rate–distortion criterion used in the main text. The construction assumes that the codec exposes synthesis responses, discrete coding or refinement decisions, baseline distortion changes, and a mechanism for satisfying a rate constraint.

Cross-image byte allocation. We allocate storage jointly over each batch of calibration images encoded at a stage instead of imposing the same compression ratio on every image. Let $\boldsymbol{\xi}_i$ collect the selected codec decisions for image i, $\Delta D^w_i(\boldsymbol{\xi}_i)$ their total weighted distortion benefit, and $R_i(\boldsymbol{\xi}_i)$ the resulting encoded size. For batch $\mathcal{S}$ and target compressed-to-raw ratio α, the conceptual allocation is

$$\begin{aligned} &\max_{\{\boldsymbol{\xi}_i\}} \sum_i \Delta D^w_i(\boldsymbol{\xi}_i) \\ &\text{s.t.} \quad S_{\mathrm{shared}} + \sum_i R_i(\boldsymbol{\xi}_i) \leq \alpha S_{\mathrm{raw}}(\mathcal{S}), \end{aligned} \tag{5}$$

where S_{shared} denotes fields shared across images when present. This objective describes the intended allocation of storage and does not require an exact global solution of the codec's discrete decisions or exact budget saturation.

JPEG2000 coefficient weights. JPEG2000 optionally applies a color transform, decomposes each component by a discrete wavelet transform (DWT), groups coefficients into code-blocks, and encodes each block into a truncatable sequence of coding passes (Skodras et al., 2001; Taubman, 2000). For the irreversible configuration used here, let $\boldsymbol{T}^{-1}_{\mathrm{color}}$ be the inverse color-transform matrix and let $\psi_{c,k}(p)$ be the reconstructed component response at pixel p to a unit error in wavelet coefficient k of component c. Let $w^{\mathrm{enc}}_{p,\mathrm{ch}}$ denote the RGB-coordinate entries of the input-importance vector $\boldsymbol{w}^{\mathrm{enc}}_i$ after reshaping it by pixel p and channel ch. The RGB synthesis response is $\phi_{c,k}(p,\mathrm{ch}) = [\boldsymbol{T}^{-1}_{\mathrm{color}}]_{\mathrm{ch},c}\psi_{c,k}(p)$. Substituting this factorization into Eq. (4) gives

$$w^{\mathrm{comp}}_{p,\mathrm{c}} = \frac{\sum_{\mathrm{ch}}[\boldsymbol{T}^{-1}_{\mathrm{color}}]^2_{\mathrm{ch},c} w^{\mathrm{enc}}_{p,\mathrm{ch}}}{\sum_{\mathrm{ch}}[\boldsymbol{T}^{-1}_{\mathrm{color}}]^2_{\mathrm{ch},c}},$$
$$\rho_{c,k} = \frac{\sum_p w^{\mathrm{comp}}_{p,c}\psi_{c,k}(p)^2}{\sum_p \psi_{c,k}(p)^2}.$$

Without a color transform, $\boldsymbol{T}^{-1}_{\mathrm{color}} = \boldsymbol{I}$. The denominators remove color and spatial synthesis energy, so uniform RGB weights give $\rho_{c,k} = 1$ and leave JPEG2000's baseline distortion unchanged. The coefficient-diagonal approximation omits cross-component and cross-coefficient couplings from the exact weighted RGB error.

Weighted PCRD. JPEG2000 post-compression rate-distortion optimization (PCRD) chooses truncation points from coding-pass prefixes according to distortion reduction per additional coded byte (Taubman, 2000). Let i index images, b code-blocks, and $\Delta D_{ibk,\tau}$ the baseline distortion reduction attributable to coefficient k in pass τ. Retaining the first L passes of block b gives

$$\Delta D^w_{ib}(L) = \sum_{\tau=1}^{L} \sum_{k \in b} \rho_{ibk} \Delta D_{ibk,\tau},$$

and a candidate extension from prefix L_0 to L_1 has weighted slope

$$\eta^w_{\mathrm{RD},ib}(L_0, L_1) = \frac{\Delta D^w_{ib}(L_1) - \Delta D^w_{ib}(L_0)}{R_{ib}(L_1) - R_{ib}(L_0)},$$

for positive byte increment, where $R_{ib}(L)$ is cumulative coded length. We apply one weighted slope threshold across the encoding batch to select code-block prefixes under the global byte budget. This permits image-specific rates without assuming that Eq. (5) is solved exactly.

The stored representation consists of the shared codestream fields and image-specific coded data needed to reconstruct a standard JPEG2000 codestream. Sensitivity affects only encoder-side decisions, so the decoder requires no AGOP state or sensitivity map. Because the codestream decisions are discrete, we do not apply the gradient-based latent refinement used by AGOP-PCA.

E Additional Experimental Details

Preprocessing and architectures. For CIFAR-100 and TinyImageNet, training images are augmented with random crops and horizontal flipping. Evaluation on both datasets uses the original spatial resolution without random augmentation. Inputs are normalized using the corresponding dataset-specific channel statistics.

For ImageNet-100, we follow the 256-pixel ImageNet resizing convention of AlexNet (Krizhevsky et al., 2012) by first resizing the shorter side to 256 pixels, then adopting a random resized crop to 224×224 for training and a center crop to 224×224 for classifier calibration and evaluation. Inputs are normalized with ImageNet channel statistics. Accordingly, calibration samples are encoded and their storage is accounted for at the 224×224 spatial resolution actually presented to the model.

We use a CIFAR-style ResNet-18 stem for CIFAR-100 and TinyImageNet, and the standard ImageNet ResNet-18 stem for ImageNet-100.

Continual training and classifier calibration. We optimize the network with SGD using an initial learning rate of 0.01, momentum 0.9, weight decay 5×10^{-4}, and batch size 16. Each task is trained for 30 epochs on CIFAR-100 and TinyImageNet and for 50 epochs on ImageNet-100. The high-fidelity replay memory is maintained by reservoir sampling.

At each classifier-calibration stage of our methods, the backbone is frozen and only the classifier is optimized. We perform 1,000 SGD steps with learning rate 0.01, without momentum or weight decay, using the full calibration objective defined in Section 3.1. We set the calibration mixing weight to $\omega_{\mathrm{comp}} = 0.6$ for AGOP-PCA and 0.8 for AGOP-J2K in all experiments.

AGOP-PCA settings. We use a single global PCA basis fitted from Task 1 training samples and keep it fixed thereafter. The retained rank is $r = 128$ for CIFAR-100 and TinyImageNet and $r = 256$ for ImageNet-100. Persistent PCA quantities, including latent codes, the basis, input mean, and class scales, are stored in 8-bit floating point (FP8).

Latent refinement is initialized from the PCA projection and optimized for 1,000 Adam steps with learning rate 10^{-3}. We set $(\lambda_{\mathrm{feat}}, \lambda_{\mathrm{mean}}, \lambda_{\mathrm{spread}}, \lambda_{\mathrm{shape}}) = (1, 1, 3, 30)$ in Equation (3). Reconstructed inputs are clipped to $[0, 1]$ before being passed to the network. During latent refinement, we use a straight-through gradient through this clipping operation.

AGOP-J2K settings. We implement AGOP-J2K on top of OpenJPEG, using its default codec configuration except for our sensitivity-guided rate–distortion allocation. The sensitivity mixture coefficient is set to $\lambda_{\mathrm{sample}} = 0.25$, and the dataset-level target compression ratio is $20\times$.

Knowledge distillation. For tasks after the first, KD is applied to replayed old-class samples and restricted to logits of previously observed classes. With teacher and student distributions at temperature T_{KD}, we use

$$\mathcal{L}_{\mathrm{KD}} = T_{\mathrm{KD}}^2 \, \mathrm{KL}\Big(p_{\mathrm{teacher}}^{(T_{\mathrm{KD}})} \,\|\, p_{\mathrm{student}}^{(T_{\mathrm{KD}})}\Big),$$

with $T_{\mathrm{KD}} = 2.0$. The base KD coefficient is 1.0 and is scaled by $\sqrt{n_{\mathrm{old}}/n_{\mathrm{new}}}$, where n_{old} and n_{new} denote the numbers of training samples from previous tasks and the current task, respectively.

Storage accounting and baseline implementations. Raw images in the high-fidelity replay memory are stored as RGB uint8 data. For AGOP-PCA, latent codes and the persistent reconstruction-side quantities described above are included in the same budget. For AGOP-J2K, we count the actual serialized image-specific codestream bytes. Our serialization extracts codestream fields that are identical for all samples of a dataset and stores these shared fields once; their storage cost is included once per dataset, while image-specific coded data are counted per sample. Table 4 illustrates the actual memory composition.

Table 4: Memory composition under the common persistent-storage budget. The replay and calibration sample counts are equal by design. Storage shares report the fractions of the total budget occupied by high-fidelity replay samples, compressed calibration samples, and the remaining method-specific persistent storage, respectively.

ENCODER	QUANTITY	CIFAR-100	TINYIMAGENET	IMAGENET-100
PCA	Replay/calibration samples	1,686	4,613	4,820
	Replay storage (%)	84.5	92.3	94.4
	Calibration storage (%)	4.0	1.1	0.1
	Other storage (%)	11.5	6.6	5.5
J2K	Replay/calibration samples	1,895	4,755	4,815
	Replay storage (%)	95.0	95.2	96.3
	Calibration storage (%)	5.0	4.8	3.7
	Other storage (%)	0.0	0.0	0.0

We follow the official implementations of the compared baselines and adapt them to the common ER-based training framework and storage-budget convention used in our experiments.

F MEMORY ALLOCATION ABLATION

To further characterize the trade-off between high-fidelity replay and compressed calibration coverage, we vary the calibration-to-replay sample ratio under a fixed persistent-storage budget. We vary the calibration-to-replay sample ratio $r_{\mathrm{cal}} = |\mathcal{B}_t|/|\mathcal{M}_t|$ on CIFAR-100 while keeping the total persistent-storage budget in bytes fixed. For each ratio, we set $|\mathcal{B}_t| = r_{\mathrm{cal}}|\mathcal{M}_t|$ and choose the largest feasible replay/calibration pair under the common byte budget after accounting for method-specific persistent storage. [1] All other training, compression, and calibration settings are kept unchanged, including the calibration mixing weights ω_{comp}. Since $\mathcal{L}_{\mathrm{replay}}$ and $\mathcal{L}_{\mathrm{comp}}$ are mean cross-entropy terms, changing r_{cal} changes the historical support available under the fixed storage budget without implicitly changing their relative loss weights.

As shown in Figure 4, both encoders attain their highest observed final accuracy at $r_{\mathrm{cal}} = 1$, while performance remains relatively stable around balanced sample counts ($r_{\mathrm{cal}} \in [1/2, 2]$). The degradation at small ratios indicates that allocating too few samples to compressed memory limits the additional historical coverage available for classifier calibration. On the other hand, performance drops increasingly sharply at large ratios. This implies diminishing marginal returns in coverage from additional compressed calibration samples, while their storage increasingly displaces high-fidelity replay that is needed to maintain the representation. Because compressed samples are used only for classifier-only calibration with a frozen backbone, they cannot compensate for insufficient representation rehearsal. The consistent trend for AGOP-PCA and AGOP-J2K therefore supports the intended role separation: compressed calibration memory complements high-fidelity replay by increasing historical coverage, but does not replace the replay needed to maintain the representation.

G FEATURE-DISTRIBUTION COVERAGE ANALYSIS

We evaluate how well the finite historical support represented by memory recovers the full feature distribution after the final task on CIFAR-100. Using each method's final frozen backbone, let P_{mem} and P_{full} denote the feature distributions of its stored historical support and the full training set, respectively, with means $\boldsymbol{\mu}_{\mathrm{mem}}, \boldsymbol{\mu}_{\mathrm{full}}$ and covariances $\boldsymbol{\Sigma}_{\mathrm{mem}}, \boldsymbol{\Sigma}_{\mathrm{full}}$. For ER, this support is its high-fidelity replay memory; for AGOP-PCA and AGOP-J2K, it is the union of high-fidelity replay and decoded compressed calibration samples, which is the historical support used during classifier calibration.

[1] Thus, r_{cal} controls the allocation in sample counts rather than in bytes: because compressed calibration samples are much smaller than raw replay samples, equal sample counts do not imply equal storage shares.

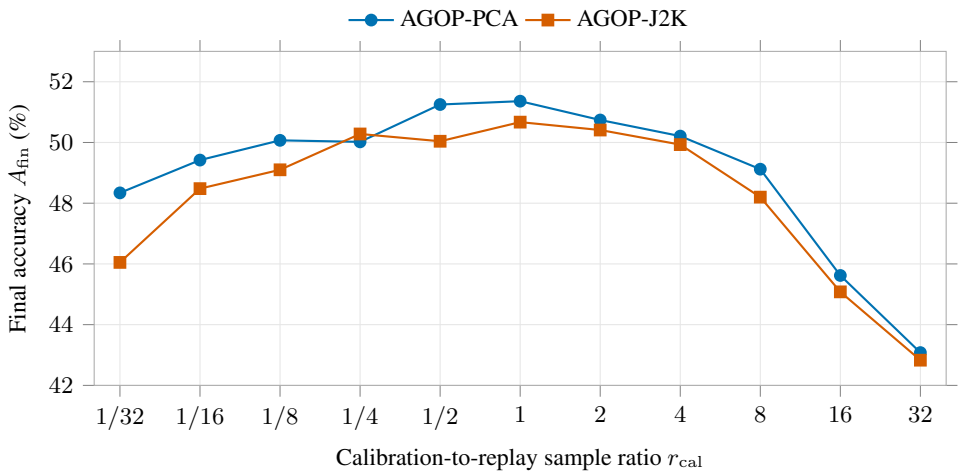

Figure 4: Memory allocation ablation on CIFAR-100 under a fixed total byte budget. The horizontal axis shows the calibration-to-replay sample count ratio r_{cal} on a base-2 logarithmic scale; the vertical axis reports final accuracy A_{fin} in percent.

The discrepancies reported in Table 3 are $\Delta_\mu = \|\boldsymbol{\mu}_{\text{mem}} - \boldsymbol{\mu}_{\text{full}}\|_2$ and $\Delta_\Sigma = \|\boldsymbol{\Sigma}_{\text{mem}} - \boldsymbol{\Sigma}_{\text{full}}\|_F$. We additionally normalize them as $\bar{\Delta}_\mu = \Delta_\mu / \sqrt{\text{tr}(\boldsymbol{\Sigma}_{\text{full}})}$ and $\bar{\Delta}_\Sigma = \Delta_\Sigma / \|\boldsymbol{\Sigma}_{\text{full}}\|_F$ to reduce dependence on feature scale, and report MMD^2 as a complementary non-parametric distribution discrepancy. All metrics are lower-is-better.

The main-text visualization in Figure 3 illustrates the same phenomenon captured quantitatively by Table 3. With ER, the sparse replay support can be displaced from the full class distribution and substantially underestimate its spread. Adding compressed calibration samples expands the occupied region and brings the memory-support ellipses closer to the full-data ellipses for both AGOP-PCA and AGOP-J2K. Because the visualization uses a two-dimensional PCA projection, it should be interpreted only as a geometric illustration rather than as the quantitative metric itself; the full-dimensional discrepancies provide the primary evidence. Their agreement indicates that the compressed memory recovers historical feature-space support beyond the sparse high-fidelity exemplars available to ER.