HISTORICAL SUPERVISION REACTIVATION WITH DYNAMIC HETEROGENEOUS MEMORY FOR LONG-HORIZON CLASS-INCREMENTAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Rehearsal-free prompt-based methods adapt pretrained models to new classes by training a small set of prompt parameters, yet can still forget old classes during subsequent learning. We investigate whether a limited history of examples can alleviate this forgetting by restoring direct supervision. We introduce Historical Supervision Reactivation (HSR), which enables losses on stored images to update trainable prompts and the classifier. To sustain this training process within limited storage, we further introduce Dynamic Heterogeneous Memory (DHM). DHM protects historical images and reserves replay quotas to retain their update paths and opportunities for use, while managing the remaining examples through a classification-stability-guided lifecycle from FP32 images to FP16 images and compact features. Our controlled study evaluates 23 configurations across three seeds on CIFAR-100 and ImageNet-R, for 69 runs with ten tasks and three training epochs per task. In supervision-path comparisons, HSR improves mean final accuracy by 1.84 and 3.15 percentage points over image replay with historical prompt gradients blocked. Under the same memory caps, DHM improves accuracy by 1.99 and 1.01 points over a fixed heterogeneous buffer. Historical image supervision mitigates forgetting in prompt-based models, and DHM further improves accuracy and forgetting under a fixed memory budget.

1 INTRODUCTION

Class-incremental learning requires a model to learn new classes sequentially and recognize all seen classes at test time. As the sequence grows, early classes must remain recognizable through more subsequent updates, making forgetting a central challenge for long-term learning. Rehearsal-free prompt-based methods adapt pretrained models by training a small number of prompt parameters while retaining old knowledge without storing historical examples (Wang et al., 2022c;b; Smith et al., 2023).

Freezing the visual backbone, however, does not keep old-class recognition unchanged: prompts and the classifier continue to evolve. Without replay, the model can no longer obtain direct supervision from its errors on old examples. Our ten-task supervision-path comparisons show that replaying historical images improves final accuracy and reduces forgetting relative to training without replay (Table 3). This motivates our question: *when a small amount of historical data may be retained, how can direct old-class supervision alleviate continued forgetting in a prompt-based model?*

We introduce *Historical Supervision Reactivation* (HSR), which returns stored images to prompt learning. Losses computed from historical images under the current model act on both trainable prompts and the classifier, restoring their direct supervision of subsequent training. We study this process through old-class retention and isolate the contribution of prompt updates by restricting which parameters historical losses can update.

Reintroducing historical examples also creates a storage problem. A fixed memory cannot continually accommodate accumulating images, while replacing images with compact features changes their training role. We therefore introduce *Dynamic Heterogeneous Memory* (DHM), which adjusts stored representations according to learning behavior while retaining historical images for prompt

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.

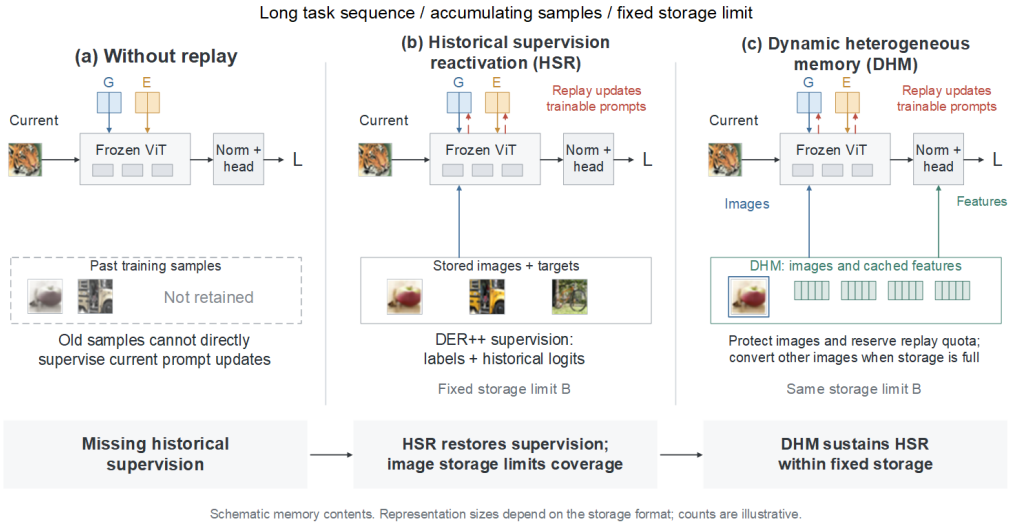


Figure 1: Motivation. (a) Without replay, old examples cannot directly supervise current prompt updates. (b) HSR restores supervision through historical images. (c) DHM protects images and reserves replay quotas while converting other records under a fixed storage cap. Memory contents and counts are schematic.

training. HSR restores historical supervision; DHM supports its continued use within finite storage (Figure 1).

Our controlled study comprises 69 runs on CIFAR-100 and ImageNet-R, with ten tasks, three epochs per task, and seeds 0, 1, and 2. Supervision-path controls examine historical prompt updates, budget-matched comparisons and ablations assess DHM, and memory diagnostics reveal storage and replay behavior. Our contributions are:

- **Restoring historical supervision for prompt learning.** HSR uses a limited set of historical images, labels, and stored predictions to supervise trainable prompts and the classifier. Controlled update paths establish the contribution of historical supervision to prompt training and old-class retention.
- **Managing memory to sustain that supervision.** DHM combines image protection and replay quotas with a stability-guided representation lifecycle. Protected images retain their update paths and replay opportunities, while other records transition from images to lower-precision images and then features to release space for subsequent examples.

2 Related Work

Rehearsal-free prompts and old-class forgetting. L2P, DualPrompt, and CODA-Prompt organize prompts to reduce interference without retaining past examples in their original settings (Wang et al., 2022c;b; Smith et al., 2023). PGP constrains prompt gradients, while MISA addresses initialization and output-layer forgetting (Qiao et al., 2024; Kang et al., 2025). These approaches primarily preserve knowledge through prompt organization or update rules. HSR instead studies direct supervision from a limited history; its access to stored data is an explicit difference. Lanzillotta et al. (2026) distinguish loss of feature separability from classifier failure. Accordingly, we use forgetting to mean declining old-class recognition, without attributing all of it to prompt parameters. The effect of historical prompt updates is tested with an explicit gradient-blocking control.

Image and feature replay. DER stores historical predictions, and DER++ adds supervision from their labels (Buzzega et al., 2020). HSR uses these two established replay losses. Latent Replay, REMIND, and ACAE-REMIND investigate replaying intermediate or compressed representations and the associated trainable network components (Pellegrini et al., 2020; Hayes et al., 2020; Wang

et al., 2021). In our model, images allow historical losses to update prompts, whereas cached features supervise the classifier. DuPt already combines replay and feature transfer with prompt learning (Jiang et al., 2025); our contribution is therefore not the first use of replay with prompts. We retain DualPrompt's architecture and focus on the historical supervision path and the memory lifecycle that sustains it. MRDC compresses replay data and selects compression quality using a determinantal point process (Wang et al., 2022a); AHR replays reconstructions from autoencoder representations (Khademi Nori et al., 2025). DHM feeds cached features directly into the classifier. Converting an image into a feature therefore changes both its storage cost and which parameters it can supervise.

Resource-constrained memory management. Equal record counts need not imply equal storage. MEMO accounts for model and exemplar costs, and aL-SAR considers storage and computation through adaptive freezing and retrieval (Zhou et al., 2023; Seo et al., 2025). DHM constrains persistent historical records in bytes and uses stability to order conversions while retaining image-based supervision. This budget excludes model parameters and temporary training tensors.

3 Problem Setup

The learner encounters T stages. Stage t provides dataset $\mathcal{D}_t$ and new classes $\mathcal{C}_t$, with disjoint class sets across stages. At test time, the model predicts among all seen classes $\mathcal{C}_{\leq t}$ without a task identifier. Later training can access historical records only through memory $\mathcal{R}_t$, rather than the complete old training sets. Its byte cap B is fixed as the stream grows. An example from stage τ experiences $t - \tau$ subsequent stages by stage t.

Let G be a frozen visual backbone, P its trainable prompts, and f_{cls} the feature-normalization and classification module. With DualPrompt as the base architecture, the logits for image x are

$$u(x) = f_{\text{cls}}(G(x; P)). \tag{1}$$

We call an example's replay entry point and the parameters reached by its loss its *training path*. Images pass through current prompts, allowing gradients to propagate through the frozen backbone to trainable prompts. Cached features enter the classifier module directly and update only that module.

A record stores an image or feature r_i, label y_i, historical logits z_i, origin stage τ_i, state $q_i^{(t)}$, and protection flag a_i. States H, M, and L denote FP32 images, FP16 images, and cached features, respectively. H and M have the same resolution. Costs include the representation, label, stored predictions, and metadata. The cap applies to persistent historical memory, not Python object overhead or training-device memory.

4 Method

4.1 Historical Supervision Reactivation

HSR reintroduces stored images into the current model so that historical losses update prompts and the classifier (Figure 2). H and M retain this path; L supplies classifier-only supervision. Image replay suffices for HSR, while features complement it within DHM.

DualPrompt uses shared G-Prompts and expert E-Prompts. Current examples select experts using the current stage during training. Historical images are grouped by stored origin stage τ_i and use the corresponding experts; stability evaluation follows the same rule. At test time, query–key matching selects prompts without ground-truth task identities. Origin stages are recorded on arrival and included in the memory accounting.

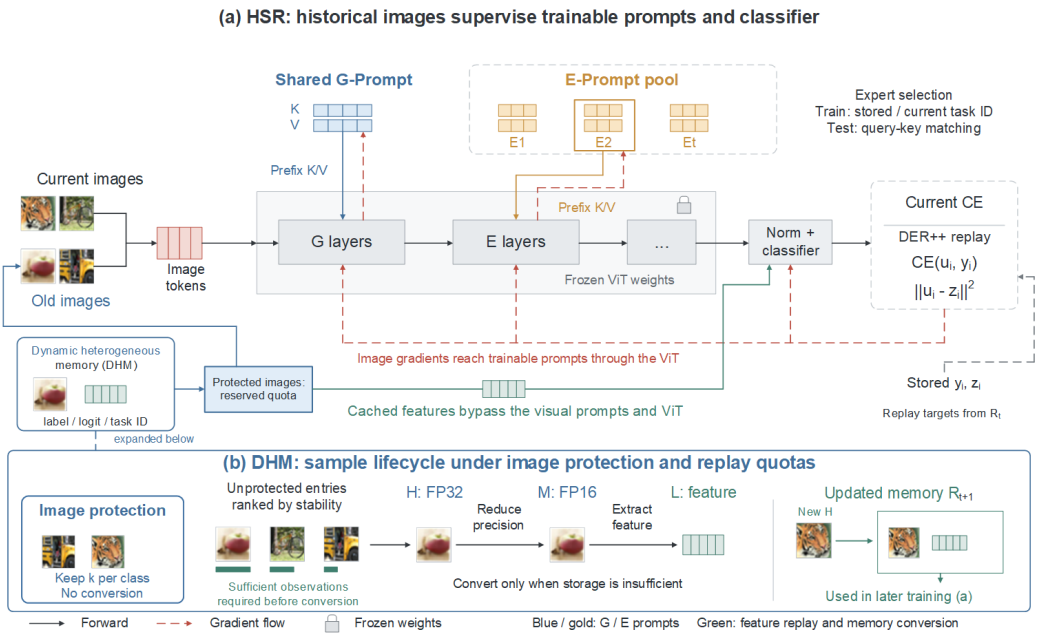


Figure 2: HSR training paths and the DHM lifecycle. Image losses reach trainable prompts through the frozen backbone; features enter the classifier module. Protection and quotas retain image replay while eligible unprotected records are converted under storage pressure. Red dashed arrows denote gradient flow. The logit discrepancy in the schematic is dimension-normalized as in Equation 5; the complete objective also includes the pull term in Equation 2.

For current batch $\mathcal{B}_c$ and historical batch $\mathcal{B}_r$, the objective is

$$\mathcal{L} = \mathcal{L}_{\text{cur}} - \lambda_p s_{\text{pull}} + \lambda_y \mathcal{L}_{\text{rep}}^{\text{cls}} + \lambda_z \mathcal{L}_{\text{rep}}^{\text{logit}}, \tag{2}$$

$$\mathcal{L}_{\text{cur}} = \frac{1}{|\mathcal{B}_c|} \sum_{j \in \mathcal{B}_c} \text{CE}(u_j | c_t, y_j), \tag{3}$$

$$\mathcal{L}_{\text{rep}}^{\text{cls}} = \frac{1}{|\mathcal{B}_r|} \sum_{i \in \mathcal{B}_r} \text{CE}(u_i | c_{\le t}, y_i), \tag{4}$$

$$\mathcal{L}_{\text{rep}}^{\text{logit}} = \frac{1}{|\mathcal{B}_r|} \sum_{i \in \mathcal{B}_r} \frac{1}{d_i} \sum_{c=1}^{d_i} (u_{i,c} - z_{i,c})^2. \tag{5}$$

Here s_{pull} is DualPrompt's query–key similarity term (`reduce_sim`), and d_i is the valid logit-prefix length when record i was stored. Prediction matching does not constrain classes unseen at that time. We use $\lambda_p = 1$, $\lambda_y = 1$, and $\lambda_z = 0.5$. Equation 5 averages over each record's valid dimensions before averaging over records; it is not an unnormalized squared Euclidean norm. The two replay losses follow DER++ (Buzzega et al., 2020). Image records permit historical gradients to reach prompts and the classifier, while feature records update only the classifier. If memory is empty, replay terms are omitted.

4.2 Dynamic Heterogeneous Memory

Newly admitted examples enter as H records. H→M reduces numerical precision while preserving image-based training; M→L releases further space and changes the record to classifier-only supervision. Labels, historical predictions, and origin stages are retained (Table 1). If c_H, c_M, c_L are complete per-record costs, then

$$\sum_{i \in \mathcal{R}_t} c_{q_i^{(t)}} \le B, \qquad c_H > c_M > c_L. \tag{6}$$

Table 1: Storage states and training roles. The classifier module includes feature normalization.

State	Representation	Replay entry	Updated parameters
H	FP32 image	Image input	Prompts and classifier
M	FP16 image	Image input	Prompts and classifier
L	Cached feature	Feature normalization	Classifier only

Eligible unprotected records either stay in their current state or advance one step, with $\mathrm{next}(H) = M$ and $\mathrm{next}(M) = L$. Not every record must eventually become a feature.

Protecting update paths and replay opportunities. DHM retains the first example of each class as a permanent H anchor, exempt from conversion and eviction ($k = 1$). Space reservation ensures room for future class anchors. Appendix A specifies protection and replay sampling.

Uniform sampling draws bA/N protected images in expectation from N records containing A protected images. Feature accumulation can therefore dilute image replay. DHM reserves $\mathrm{round}(\rho b_r)$ slots for protected images when available, filling remaining slots from other records. Appendix A specifies sampling and shortfall handling.

Capacity-triggered conversion. Conversion is triggered only when admitting a new H record would exceed B: $S_t^{\mathrm{mem}} + c_H + B_{\mathrm{reserve}}^{\mathrm{post}} > B$, where S_t^{mem} is resident occupancy and $B_{\mathrm{reserve}}^{\mathrm{post}}$ is the reservation after the proposed admission. Candidates must be unprotected, convertible, and sufficiently observed; conversion must also respect the minimum image retention per class. The protocol knows the final class count when setting k, but does not access future-class examples. Among eligible candidates, DHM converts the record with the highest classification stability. H→M and M→L release $c_H - c_M$ and $c_M - c_L$ bytes. Capacity is checked after each conversion. Conversion stops once the new record fits; if no eligible candidate remains, non-anchor eviction provides admission space while preserving anchors and reservations.

Anchor-reserved admission. DHM uses one permanent H anchor per class and $(C_{\mathrm{total}} - C_{\mathrm{seen}})c_H$ bytes reserved for unseen classes. Resident plus reserved bytes stay within B. Resident duplicate IDs are skipped; non-anchor eviction provides a fallback when conversion cannot make room. It requires the known final class count and $B \geq C_{\mathrm{total}} c_H$, without accessing future examples. The full-DHM results in Tables 4–5 use this mechanism, whose memory behavior is shown in Figure 3. Appendix A.2 gives details.

4.3 Classification Stability Across Stages

At each stage end, unprotected images are evaluated with fixed test transforms and origin-stage prompts. Let $m_{i,t}$ be the correct-class probability margin over the strongest competing seen class, and $\mu_{i,t}$ and $\sigma_{i,t}$ its mean and population standard deviation within the current storage state. We rank eligible candidates by

$$S_{i,t} = \min\{m_{i,t}, \mu_{i,t} - \lambda\sigma_{i,t}\}, \qquad \lambda \geq 0. \tag{7}$$

The highest-scoring eligible record is converted first, retaining difficult or unstable images longer. Eligibility requires $n_{\min}$ observations. H→M resets the statistics, so conversion to L requires fresh observations. Appendix A.1 defines the computations in full.

5 Experiments

5.1 Setup and Metrics

We use CIFAR-100 (Krizhevsky, 2009) and ImageNet-R (Hendrycks et al., 2021), each split into ten tasks (10 and 20 classes per task, respectively). ImageNet-R uses fixed splits of 24,000 training and 6,000 test images across 200 classes. Seeds 0, 1, and 2 vary class order; methods share the same order within each seed. The controlled study evaluates 23 configurations with three seeds each: 42

Table 2: External-method comparison under shared task schedules, data splits, and class orders. Values are three-seed means ± sample standard deviations. Architectures and replay resources differ; this is not an equal-byte comparison. Bold marks the best mean among listed methods; — indicates unavailable results.

	CIFAR-100		ImageNet-R	
Method	Acc.↑	Fgt.↓	Acc.↑	Fgt.↓
PGP	86.16 ± 0.12	15.18 ± 0.43	69.90 ± 0.19	15.82 ± 0.21
CODA-Prompt	83.06 ± 0.94	16.89 ± 0.56	67.02 ± 0.22	17.07 ± 0.50
ER	31.06 ± 5.95	73.49 ± 6.54	—	—
DER++	39.28 ± 3.88	64.53 ± 4.09	42.73 ± 5.99	52.39 ± 6.25
SigLIP So400m + LoRA-ER	78.52 ± 0.74	21.48 ± 0.88	—	—
HSR + DHM	$\mathbf{89.33} \pm 0.16$	$\mathbf{9.29} \pm 1.99$	$\mathbf{70.11} \pm 0.61$	$\mathbf{10.34} \pm 1.79$

runs on CIFAR-100 and 27 on ImageNet-R. Shared reference configurations are counted once; the external-method comparisons are additional experiments.

The controlled study uses DualPrompt with frozen ViT-B/16, 224×224 inputs, and 768-dimensional features. We train for three epochs per task with Adam at an effective learning rate of 0.00375. Current and historical batch caps are 32. DHM uses $k = 1$, $n_{\min} = 2$, $\lambda = 1$, and $\rho = 0.5$, without test-accuracy selection. Appendix B specifies optimization and preprocessing; Appendix E lists class orders.

The historical memory caps are 6,376,000 bytes for CIFAR-100 and 301,488,000 bytes for ImageNet-R. Accounting includes representations, reserved logits for all classes, and 64 bytes of metadata per record. These are logical record budgets, not measured Python or GPU memory. No replay uses zero historical memory. Other budget-matched methods share the cap, without requiring identical realized occupancy. The path-analysis FEATURE baseline stores 500 cached features; FEATURE_BYTES is the separate budget-matched feature baseline.

Let $a_{t,i}$ be Class-IL accuracy on task i after learning task t, with predictions over all seen classes. Final accuracy and forgetting are

$$A_T = \frac{1}{T}\sum_{i=1}^{T} a_{T,i}, \qquad F_T = \frac{1}{T-1}\sum_{i=1}^{T-1}\left(\max_{t=i,\dots,T} a_{t,i} - a_{T,i}\right). \tag{8}$$

The maximum includes the final stage, and forgetting excludes the last task. Accuracy is reported in percent and forgetting in percentage points. In particular, ImageNet-R accuracy weights tasks equally rather than weighting all test examples equally. Tables report three-seed means and sample standard deviations ($\mathrm{ddof} = 1$), not confidence intervals.

5.2 Comparison with External Methods

We compare with PGP, CODA-Prompt, ER, DER++, and SigLIP So400m + LoRA-ER (Table 2). All methods use ten tasks, three epochs per task, seeds 0–2, and matched splits and class orders. PGP and CODA-Prompt provide prompt-based references; ER and DER++ fully train ViT-B/16 on CIFAR-100; SigLIP So400m + LoRA-ER tests adaptation of a larger pretrained backbone. The external CIFAR-100 replay methods retain 500 records, whereas DHM uses a byte cap. Appendix C gives resource and training details.

Accuracy and old-class retention. HSR + DHM has the highest mean accuracy and lowest mean forgetting among the listed methods on both datasets. Compared with PGP, accuracy improves by 3.17 points on CIFAR-100 and 0.21 on ImageNet-R, while forgetting decreases by 5.89 and 5.48 points. Compared with CODA-Prompt, accuracy improves by 6.27 and 3.09 points, with forgetting reductions of 7.60 and 6.73 points. These are comparisons of complete systems with differing replay access and architectures; controlled analyses below isolate historical supervision and memory design.

Comparison with larger-backbone LoRA adaptation. On CIFAR-100, DHM exceeds SigLIP So400m + LoRA-ER by **10.81 accuracy points** (89.33 vs. 78.52%) and reduces forgetting by **12.19**

Table 3: Supervision-path comparison. Higher final accuracy (Acc., %) and lower forgetting (Fgt., points) are better. FEATURE stores 500 records. Values are mean ± sample standard deviation over three seeds.

	CIFAR-100		ImageNet-R	
Method	Acc.↑	Fgt.↓	Acc.↑	Fgt.↓
NO_REPLAY	85.33 ± 0.16	13.76 ± 0.47	64.35 ± 0.61	14.79 ± 1.00
FEATURE	86.51 ± 0.39	10.94 ± 0.53	67.86 ± 0.56	12.49 ± 0.92
DETACH	86.64 ± 0.20	10.60 ± 0.50	66.29 ± 0.20	13.49 ± 1.52
HSR	88.48 ± 0.84	9.88 ± 0.63	69.44 ± 1.12	11.21 ± 0.81

Table 4: Results under matched historical-memory caps. NO_REPLAY is a zero-memory reference. The NO_REPLAY and FP32/HSR rows reuse Table 3. All values are three-seed mean ± sample standard deviation.

	CIFAR-100		ImageNet-R	
Method	Acc.↑	Fgt.↓	Acc.↑	Fgt.↓
NO_REPLAY	85.33 ± 0.16	13.76 ± 0.47	64.35 ± 0.61	14.79 ± 1.00
FP32 / HSR	88.48 ± 0.84	9.88 ± 0.63	69.44 ± 1.12	11.21 ± 0.81
FP16 images	85.85 ± 0.52	12.65 ± 1.38	67.65 ± 0.97	15.35 ± 0.39
uint8 images	85.80 ± 0.25	12.99 ± 1.67	66.69 ± 0.37	15.56 ± 1.36
FEATURE_BYTES	85.64 ± 0.69	13.97 ± 0.47	66.77 ± 0.23	15.42 ± 1.18
Fixed heterogeneous	87.34 ± 1.13	10.65 ± 1.27	69.10 ± 0.71	12.88 ± 1.01
DHM	$\mathbf{89.33} \pm 0.16$	$\mathbf{9.29} \pm 1.99$	$\mathbf{70.11} \pm 0.61$	$\mathbf{10.34} \pm 1.79$

points (9.29 vs. 21.48; Figure 5). The frozen ViT-B/16 with prompt-based historical supervision thus achieves stronger retention and final recognition than this larger-backbone LoRA configuration under the shared three-epoch schedule. This comparison does not isolate model scale or adaptation type, or establish equal computational cost.

5.3 HSR and Supervision Paths

NO_REPLAY trains only on current data. HSR allows image replay losses to update prompts and the classifier. DETACH uses the same historical images and sampling but blocks their prompt gradients. FEATURE replays cached features into the classifier. HSR versus DETACH is the key update-path control; image versus feature comparisons also change the stored representation and replay entry point.

HSR achieves $88.48 \pm 0.84\%$ on CIFAR-100 and $69.44 \pm 1.12\%$ on ImageNet-R (Table 3). Relative to DETACH, accuracy improves by 1.84 and 3.15 points and forgetting decreases by 0.72 and 2.28 points. Relative to NO_REPLAY, the gains are 3.15 and 5.09 points, with forgetting reductions of 3.88 and 3.58 points. These controls support historical image supervision and its prompt-update path without attributing all forgetting to prompts.

5.4 Budget-Matched Comparisons and Ablations

We compare FP32 images, FP16 images, uint8 images, features, a fixed heterogeneous buffer, and DHM under the same storage caps (Table 4). The fixed buffer allocates half its bytes to images and half to features, with no subsequent conversion. DHM combines stability-guided H→M→L conversion with permanent class anchors, future-class reservations, and non-anchor eviction when conversion cannot make room. Their comparison measures the complete memory schemes, including their different admission rules.

DHM reaches $89.33 \pm 0.16\%$ and $70.11 \pm 0.61\%$. Compared with the fixed heterogeneous buffer, accuracy increases by 1.99 and 1.01 points and forgetting falls by 1.36 and 2.54 points. Compared with HSR, accuracy increases by 0.85 and 0.67 points and forgetting falls by 0.59 and 0.87 points. Gains over no replay are 4.00 and 5.76 points. DHM has the highest mean accuracy and lowest

Table 5: CIFAR-100 component ablations. The full DHM result is reused from Table 4.

Method	Acc. (%)↑	Fgt. (points)↓
DIRECT	86.69 ± 1.59	14.24 ± 1.04
PROTECT_ONLY	86.02 ± 1.36	14.97 ± 1.01
QUOTA_ONLY	85.12 ± 1.45	11.51 ± 1.85
RANDOM	86.37 ± 1.77	12.87 ± 1.90
CLASS_BALANCED	87.73 ± 1.06	10.35 ± 1.82
DHM	$\mathbf{89.33} \pm 0.16$	$\mathbf{9.29} \pm 1.99$

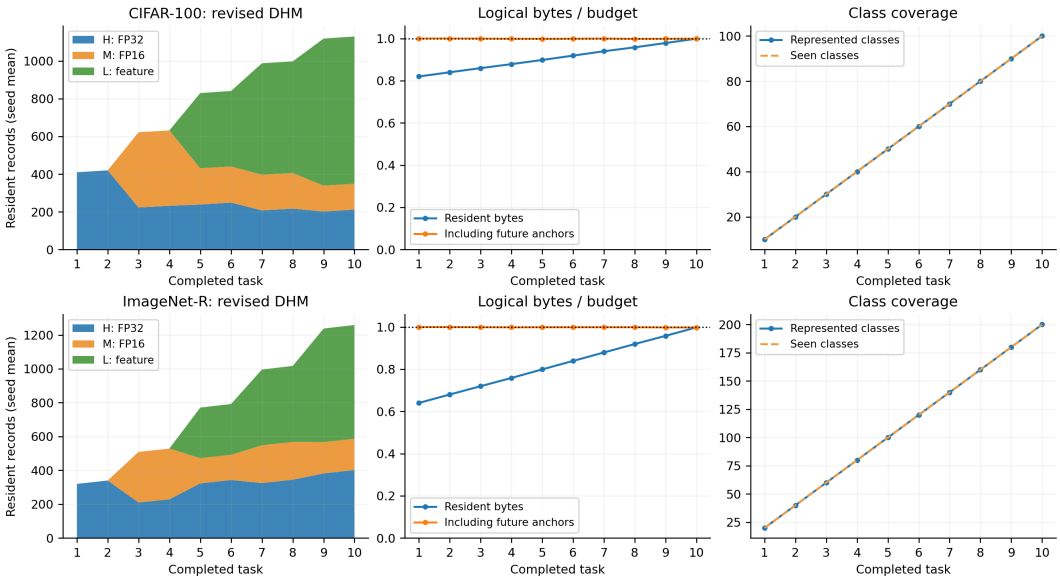


Figure 3: Anchor-reserved DHM across ten tasks (three-seed means). Left: resident FP32 image (H), FP16 image (M), and feature (L) records; stacked areas show means. Middle: logical resident bytes and resident plus reserved bytes, divided by the budget. "Including future anchors" denotes space reserved for unseen classes, not stored future samples. Right: represented and seen classes. Permanent anchors ensure coverage; no variability bands are plotted.

mean forgetting among the memory schemes in Table 4 on both datasets. These mean differences are not claims of statistical significance.

Storage pipelines also differ in preprocessing, as detailed in Appendix B; the precision and feature controls therefore compare complete pipelines.

DIRECT omits protection and quotas; PROTECT_ONLY protects images without reserving replay slots; QUOTA_ONLY reserves slots but permits conversion of those images. DHM exceeds their mean accuracies by 2.64, 3.31, and 4.21 points, respectively (Table 5). RANDOM retains eligibility rules but selects conversion candidates randomly; CLASS_BALANCED replaces the protected-image quota with class-balanced sampling. DHM exceeds them by 2.96 and 1.60 points. These results support the joint design and stability-based ordering in the tested setting; they do not establish monotonic gains for every module combination. Component ablations are performed on CIFAR-100.

5.5 Memory Lifecycle and Admission

DHM maintains H, M, and L records within the combined resident and reserved budget (Figure 3). Permanent anchors and advance reservation ensure coverage of every seen class. Growing M and L populations confirm conversion, but do not establish it as the cause of full coverage.

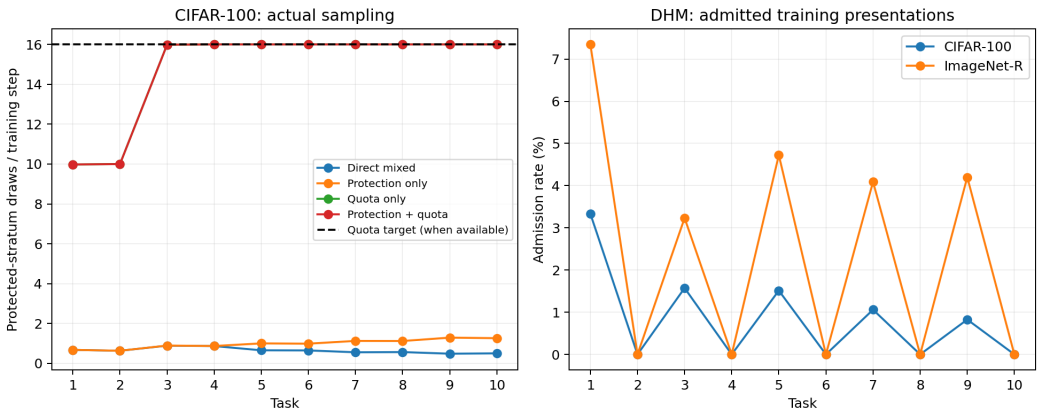


Figure 4: Diagnostics under the original admission rule: protected-stratum draws per optimization step on CIFAR-100 (left) and successful insertions divided by training presentations on both datasets (right). Draws and insertions need not represent distinct original images. The quota target applies when sufficient protected records are available.

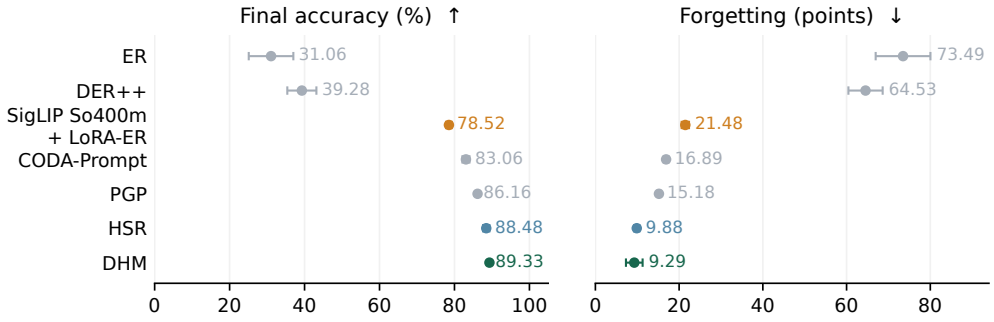


Figure 5: CIFAR-100 external comparisons. Points and bars show three-seed means and sample standard deviations. DHM improves both metrics over the tested SigLIP So400m + LoRA-ER configuration (orange). Shared data and training schedules do not imply matched backbones, trainable parameters, or byte budgets; Appendix C specifies the differences.

The quota sustains protected-image replay as feature records accumulate (Figure 4, left). Anchor reservation and non-anchor eviction complement conversion by preserving class representation and allowing admission when no conversion candidate is eligible.

6 Conclusion

HSR restores historical supervision for prompt learning, and DHM manages its representation lifecycle within a fixed memory cap. HSR improves accuracy over gradient-blocked replay by 1.84 and 3.15 points on CIFAR-100 and ImageNet-R; DHM further exceeds fixed heterogeneous memory by 1.99 and 1.01 points. External comparisons show strong accuracy and retention, including gains over the tested SigLIP So400m + LoRA-ER configuration. Future work will examine longer sequences and the interaction between admission and replay.

Reproducibility Statement

Appendices A–E specify memory operations, implementation settings, external protocols, and class orders. Equations 2–5 and 8 define losses and metrics. Experiments use three seeds with matched class orders.

AI USE STATEMENT

Generative AI tools were used for literature search, language editing and translation, checking result summaries, and manuscript formatting. Generative AI was not used to implement the experimental code. The authors take responsibility for the final text, claims, and reported results.

REFERENCES

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: A strong, simple baseline. In *Advances in Neural Information Processing Systems*, volume 33, pp. 15920–15930, 2020.

Tyler L. Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. REMIND your neural network to prevent catastrophic forgetting. In *European Conference on Computer Vision*, pp. 466–483, 2020. URL https://arxiv.org/abs/1910.02509.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8349, 2021.

Shengqin Jiang, Daolong Zhang, Fengna Cheng, Xiaobo Lu, and Qingshan Liu. DuPt: Rehearsal-based continual learning with dual prompts. *Neural Networks*, 187:107306, 2025. doi: 10.1016/j.neunet.2025.107306.

Zhiqi Kang, Liyuan Wang, Xingxing Zhang, and Karteek Alahari. Advancing prompt-based methods for replay-independent general continual learning. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2503.00677.

Milad Khademi Nori, Il Min Kim, and Guanghui Wang. Autoencoder-based hybrid replay for class-incremental learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 29753–29767, 2025. URL https://proceedings.mlr.press/v267/khademi-nori25a.html.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Giulia Lanzillotta, Damiano Meier, and Thomas Hofmann. Heads collapse, features stay: Why replay needs big buffers. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=IdW0d0mRnG.

Lorenzo Pellegrini, Gabriele Graffieti, Vincenzo Lomonaco, and Davide Maltoni. Latent replay for real-time continual learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020.

Jingyang Qiao, Zhizhong Zhang, Xin Tan, Chengwei Chen, Yanyun Qu, Yong Peng, and Yuan Xie. Prompt gradient projection for continual learning. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=EH2O3h7sBI.

Minhyuk Seo, Hyunseo Koh, and Jonghyun Choi. Budgeted online continual learning by adaptive layer freezing and frequency-based sampling. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=dOAkHmsjRX.

James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. CODA-Prompt: COntinual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Kai Wang, Joost van de Weijer, and Luis Herranz. ACAE-REMIND for online continual learning with compressed feature replay. *Pattern Recognition Letters*, 150:122–129, 2021. doi: 10.1016/j.patrec.2021.06.025.

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 *International Conference on Learning Representations*, 2022a. URL https://arxiv.org/abs/2202.06592.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. DualPrompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, 2022b.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022c.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11975–11986, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Zhai_Sigmoid_Loss_for_Language_Image_Pre-Training_ICCV_2023_paper.html.

Da-Wei Zhou, Qi-Wei Wang, Han-Jia Ye, and De-Chuan Zhan. A model or 603 exemplars: Towards memory-efficient class-incremental learning. In *International Conference on Learning Representations*, 2023.

A IMAGE PROTECTION AND REPLAY QUOTAS

Full DHM protects the first example of each class as a permanent FP32 anchor ($k = 1$). Anchor selection does not score representativeness. The original diagnostic protocol in Figure 4 protects the first k images per admitted class in buffer-index order.

Let the actual historical batch size be $b = \min(b_r, |\mathcal{R}_t|)$. The protected-image target is $\min(\mathrm{round}(\rho b_r), b)$, capped by the available protected images. Images are sampled uniformly without replacement from the protected stratum. Unprotected images and features then fill the remaining slots. If too few unprotected records exist, unused protected images fill the shortfall. The resulting batch is shuffled.

The parameter k controls per-class protected storage, while ρ specifies a reserved count relative to the historical batch cap. Sampling is over all protected images and does not ensure that each old class appears in every batch. Conversion affects only unprotected records. The protection and sampling rules jointly retain the inputs and replay opportunities required for HSR.

Full DHM retains the first example of every class permanently as an H anchor. Currently resident duplicate IDs are skipped; an evicted example may be inserted again on a later presentation. Its admission rate is defined as successful insertions divided by all training presentations, including skipped resident duplicates in the denominator. Insertions following eviction count as successful admissions. This measures entry into memory, not long-term retention of the training set.

A.1 STABILITY COMPUTATION DETAILS

Consistently easy unprotected examples are converted earlier; difficult or unstable examples retain their image path longer. At the end of each stage, candidates are evaluated with fixed test preprocessing and origin-stage prompt selection. For current logits z_{θ_t}, let

$$p_{i,t} = \mathrm{softmax}\big(z_{\theta_t}(x_i;\tau_i)|_{\mathcal{C}_{\le t}}\big), \tag{9}$$

$$m_{i,t} = p_{i,t}(y_i) - \max_{c\in\mathcal{C}_{\le t}\setminus\{y_i\}} p_{i,t}(c). \tag{10}$$

A positive margin means the correct class has the largest probability. Let $\mathcal{T}_i(t)$ contain the observation stages in the record's *current storage state*, and $n_i = |\mathcal{T}_i(t)|$. We compute

$$\mu_{i,t} = \frac{1}{n_i}\sum_{s\in\mathcal{T}_i(t)} m_{i,s}, \qquad \sigma_{i,t} = \sqrt{\frac{1}{n_i}\sum_{s\in\mathcal{T}_i(t)} (m_{i,s}-\mu_{i,t})^2}, \tag{11}$$

$$S_{i,t} = \min\{m_{i,t},\ \mu_{i,t} - \lambda\sigma_{i,t}\}, \qquad \lambda \ge 0. \tag{12}$$

The minimum prevents a high historical mean from masking current difficulty, and prevents a single large margin from masking unstable history. At least $n_{\min}$ observations are required. For eligible transition candidates $\mathcal{E}_t^{A\to B}$, select $i^\star = \arg\max_{i\in\mathcal{E}_t^{A\to B}} S_{i,t}$. After H→M, margin statistics are reset, so conversion to L requires new observations in the M state. The score orders candidates; it does not directly predict future forgetting or replace protection and quotas.

A.2 ANCHOR-RESERVED ADMISSION DETAILS

Given the known total class count C_{total}, DHM reserves one H record for each unseen class:

$$B_{\mathrm{reserve}} = (C_{\mathrm{total}} - C_{\mathrm{seen}})c_H, \qquad S_t^{\mathrm{mem}} + B_{\mathrm{reserve}} \le B. \tag{13}$$

Here c_H includes the image, stored predictions, and metadata. The first example of each class becomes a permanent H anchor, exempt from conversion and eviction. Reservation uses class counts, not future examples, and requires $B \ge C_{\mathrm{total}}c_H$. Resident duplicate IDs are skipped. When conversion cannot make room because no eligible candidate remains, non-anchor records may be evicted to admit a new record. This admission mechanism is combined with the stability score and replay quota in the full-DHM results of Tables 4–5.

B Implementation and Data Protocol

B.1 Backbone and Checkpoint

The model is ViT-B/16 with 224×224 inputs and 768-dimensional features. The implementation identifier is `vit_base_patch16_224_in21k_fn_in1k_old`, with `pretrain_type=in21k_old`. The configured checkpoint is `ViT-B_16.npz`, obtained from

https://storage.googleapis.com/vit_models/imagenet21k/ViT-B_16.npz.

Its SHA-256 is

ac7478131e1b77e5c2f0225849730bd19ee89
cfffe600c2dfd2272b34a50a43d.

The two displayed hash segments concatenate without whitespace. We specify the actual checkpoint rather than infer additional ImageNet-1k fine-tuning from the implementation identifier.

B.2 Prompt Configuration and Training Environment

Both G-Prompt and E-Prompt use a prompt length of 5 and prefix tuning. G-Prompt is inserted into Transformer blocks 0 and 1, while E-Prompt is inserted into blocks 2, 3, and 4 (zero-based indexing). The E-Prompt pool contains 10 entries, with top-k set to 1. Batchwise prompt selection is disabled. The embedding key is `cls`; prompts and prompt keys use uniform initialization. The implementation flags are `same_key_value=False` and `use_permute_fix=True`.

Experiments run on an NVIDIA GeForce RTX 4090 with a 300 W power limit. The training environment uses PyTorch 2.9.1+cu128 (CUDA build 12.8) and timm 0.9.8. Software versions were read from the training environment rather than archived separately for each run.

B.3 Optimization

All 69 controlled-study runs use Adam with an effective learning rate of 0.00375, no learning-rate schedule, zero weight decay, and gradient clipping at 1.0. The input learning-rate setting is 0.03; initialization scales it by batch size as $0.03 \times 32/256 = 0.00375$. Current batches contain 32 examples and replay batches have an upper bound of 32. Each of the ten tasks is trained for three epochs. No replay batch is used for NO_REPLAY. The pull coefficient is 1.0, replay CE coefficient is 1.0, and replay MSE coefficient is 0.5; there is no common 0.5 multiplier on the entire replay loss. Current CE masks to the current task, historical CE covers all seen classes, and MSE uses each record's valid stored prefix.

B.4 Preprocessing and Splits

CIFAR-100 uses 100 classes (Krizhevsky, 2009). Current training images undergo random resized cropping to 224×224 with bicubic interpolation, random horizontal flipping with probability 0.5, conversion to tensors, and normalization with mean $(0.485, 0.456, 0.406)$ and standard deviation $(0.229, 0.224, 0.225)$. Test preprocessing resizes to 224 with bicubic interpolation, converts to a tensor, and applies the same normalization.

ImageNet-R contains visual renditions of 200 classes (Hendrycks et al., 2021). We use the fixed `imagenet-r_train.yaml` and `imagenet-r_test.yaml` splits containing 24,000 and 6,000 images, respectively. Current training images undergo random resized cropping to 224×224 with bicubic interpolation, random horizontal flipping, and tensor conversion. Testing resizes to 256×256, center-crops to 224×224, and converts to a tensor. Normalization uses mean $(0, 0, 0)$ and standard deviation $(1, 1, 1)$, i.e., the identity transform. All seeds share the train/test split and vary class order.

Historical image replay applies `ToPILImage` followed by the corresponding training transform. Stability evaluation uses fixed test transforms. The fixed heterogeneous buffer extracts features

with deterministic preprocessing at admission; feature-only controls cache features from current augmented inputs. These are distinct protocols, not identical feature extraction pipelines.

B.5 Run Accounting and Storage

The 69 controlled-study runs comprise 24 supervision-path runs, 18 additional byte-budget baselines, six full-DHM runs, six fixed-buffer runs, and 15 CIFAR-100 ablation runs. Reused reference configurations do not count as additional runs. The controlled-study tables use three seeds and sample standard deviations. Budgets include reserved all-class logits and 64 metadata bytes per record. The known final class count is used when configuring protection; future images are unavailable during training. Byte caps constrain logical persistent records only, excluding model storage, optimizer states, temporary tensors, and language-runtime overhead.

C External Comparison Protocol

Table 2 reports additional results for PGP (Qiao et al., 2024), CODA-Prompt (Smith et al., 2023), experience replay (ER), DER++ (Buzzega et al., 2020), and a SigLIP So400m + LoRA-ER configuration using a SigLIP backbone (Zhai et al., 2023). These experiments use the same ten tasks, three epochs per task, seeds 0, 1, and 2, train/test splits, and within-seed class orders as the controlled study. Results are means and sample standard deviations over three seeds. Unavailable dataset–method results are marked with a dash.

CIFAR-100 replay configurations. ER and DER++ use an ImageNet-21k-pretrained ViT-B/16 with full-parameter training (approximately 85.88 million trainable parameters), without DualPrompt. Both retain 500 records. ER stores images and labels and draws 32 replay samples per update. DER++ also stores historical logits and draws two batches of 32: one for logit matching and one for historical-label supervision. Its 64 draws need not correspond to 64 distinct images. SigLIP So400m + LoRA-ER uses LoRA adaptation, a capacity of 500 original-image records, and 32 replay draws per update. Capacity and per-update draws are separate quantities; equal capacity does not imply equal storage bytes, especially with DER++'s additional logits. These configuration details apply to CIFAR-100; the ImageNet-R DER++ backbone and replay-resource settings are not specified here.

Relative to PGP, DHM's reported accuracy is 3.17 points higher on CIFAR-100 and 0.21 points higher on ImageNet-R. Relative to CODA-Prompt on ImageNet-R, DHM improves mean accuracy by 3.09 points and reduces forgetting by 6.73 points. These aggregate differences do not establish statistical significance or isolate architecture, pretraining, and memory effects. The external summaries are not included in the 69-run accounting of the controlled study.

D Accuracy Differences

Table 6 consolidates differences calculated from the corresponding table means. It does not provide a paired significance test or a confidence interval.

Table 6: Differences in final mean accuracy, in percentage points. Positive values favor the first method.

Comparison	Source	CIFAR-100	ImageNet-R
HSR − DETACH	Table 3	+1.84	+3.15
HSR − NO_REPLAY	Table 3	+3.15	+5.09
DHM − Fixed heterogeneous	Table 4	+1.99	+1.01
DHM − HSR	Table 4	+0.85	+0.67

E Complete Class Orders

The following tables list original dataset label IDs by seed and task. All methods use the same order within a seed. CIFAR-100 has ten classes per task and ImageNet-R has twenty.

E.1 CIFAR-100, seed 0

Task	Original class IDs
1	26, 86, 2, 55, 75, 93, 16, 73, 54, 95
2	53, 92, 78, 13, 7, 30, 22, 24, 33, 8
3	43, 62, 3, 71, 45, 48, 6, 99, 82, 76
4	60, 80, 90, 68, 51, 27, 18, 56, 63, 74
5	1, 61, 42, 41, 4, 15, 17, 40, 38, 5
6	91, 59, 0, 34, 28, 50, 11, 35, 23, 52
7	10, 31, 66, 57, 79, 85, 32, 84, 14, 89
8	19, 29, 49, 97, 98, 69, 20, 94, 72, 77
9	25, 37, 81, 46, 39, 65, 58, 12, 88, 70
10	87, 36, 21, 83, 9, 96, 67, 64, 47, 44

E.2 CIFAR-100, seed 1

Task	Original class IDs
1	80, 84, 33, 81, 93, 17, 36, 82, 69, 65
2	92, 39, 56, 52, 51, 32, 31, 44, 78, 10
3	2, 73, 97, 62, 19, 35, 94, 27, 46, 38
4	67, 99, 54, 95, 88, 40, 48, 59, 23, 34
5	86, 53, 77, 15, 83, 41, 45, 91, 26, 98
6	43, 55, 24, 4, 58, 49, 21, 87, 3, 74
7	30, 66, 70, 42, 47, 89, 8, 60, 0, 90
8	57, 22, 61, 63, 7, 96, 13, 68, 85, 14
9	29, 28, 11, 18, 20, 50, 25, 6, 71, 76
10	1, 16, 64, 79, 5, 75, 9, 72, 12, 37

E.3 CIFAR-100, seed 2

Task	Original class IDs
1	83, 30, 56, 24, 16, 23, 2, 27, 28, 13
2	99, 92, 76, 14, 0, 21, 3, 29, 61, 79
3	35, 11, 84, 44, 73, 5, 25, 77, 74, 62
4	65, 1, 18, 48, 36, 78, 6, 89, 91, 10
5	12, 53, 87, 54, 95, 32, 19, 26, 60, 55
6	9, 96, 17, 59, 57, 41, 64, 45, 97, 8
7	71, 94, 90, 98, 86, 80, 50, 52, 66, 88
8	70, 46, 68, 69, 81, 58, 33, 38, 51, 42
9	4, 67, 39, 37, 20, 31, 63, 47, 85, 93
10	49, 34, 7, 75, 82, 43, 22, 72, 15, 40

E.4 IMAGENET-R, SEED 0

Task	Original class IDs
1	18, 170, 107, 98, 177, 182, 5, 146, 12, 152, 61, 125, 180, 154, 80, 7, 33, 130, 37, 74
2	183, 145, 45, 159, 60, 123, 179, 185, 122, 44, 16, 55, 150, 111, 22, 189, 129, 4, 83, 106
3	134, 66, 26, 113, 168, 63, 8, 75, 118, 143, 71, 124, 184, 97, 149, 24, 30, 160, 40, 56
4	131, 96, 181, 19, 153, 92, 54, 163, 51, 86, 139, 90, 137, 101, 144, 89, 109, 14, 27, 141
5	187, 46, 138, 195, 108, 62, 2, 59, 136, 197, 43, 10, 194, 73, 196, 178, 175, 126, 93, 112
6	158, 191, 50, 0, 94, 110, 95, 64, 167, 41, 69, 49, 48, 85, 13, 161, 23, 186, 135, 20
7	15, 78, 104, 52, 100, 76, 3, 116, 164, 198, 6, 68, 84, 121, 155, 171, 156, 91, 199, 11
8	119, 102, 35, 57, 65, 1, 120, 162, 42, 105, 132, 173, 17, 38, 133, 53, 157, 128, 34, 28
9	114, 151, 31, 166, 127, 176, 32, 142, 169, 147, 29, 99, 82, 79, 115, 148, 193, 72, 77, 25
10	165, 81, 188, 174, 190, 39, 58, 140, 88, 70, 87, 36, 21, 9, 103, 67, 192, 117, 47, 172

E.5 IMAGENET-R, SEED 1

Task	Original class IDs
1	58, 40, 34, 102, 184, 198, 95, 4, 29, 168, 171, 18, 11, 89, 110, 118, 159, 35, 136, 59
2	51, 16, 44, 94, 31, 162, 38, 28, 193, 27, 47, 165, 194, 177, 176, 97, 174, 73, 69, 172
3	108, 107, 189, 14, 56, 19, 114, 39, 185, 124, 98, 123, 119, 53, 33, 179, 181, 106, 199, 138
4	116, 67, 78, 42, 17, 5, 127, 105, 48, 66, 54, 84, 183, 158, 166, 113, 12, 117, 93, 120
5	154, 90, 81, 122, 191, 13, 82, 132, 187, 45, 99, 36, 161, 186, 153, 103, 195, 197, 148, 173
6	75, 21, 91, 152, 2, 70, 85, 150, 6, 112, 0, 155, 77, 65, 55, 167, 88, 130, 46, 62
7	74, 92, 147, 160, 143, 87, 180, 145, 164, 10, 32, 83, 182, 100, 125, 23, 126, 9, 170, 104
8	151, 135, 111, 188, 64, 15, 41, 163, 109, 80, 52, 26, 76, 43, 24, 3, 169, 49, 149, 131
9	190, 30, 121, 115, 175, 8, 60, 128, 1, 57, 22, 61, 63, 7, 196, 141, 86, 96, 68, 50
10	142, 157, 156, 139, 146, 101, 20, 178, 25, 134, 71, 129, 144, 192, 79, 133, 137, 72, 140, 37

E.6 IMAGENET-R, SEED 2

Task	Original class IDs
1	112, 29, 182, 199, 193, 85, 10, 54, 115, 35, 12, 92, 13, 126, 174, 2, 44, 3, 113, 14
2	23, 25, 6, 134, 165, 173, 45, 65, 48, 122, 178, 64, 9, 57, 78, 71, 128, 176, 131, 53
3	137, 163, 111, 123, 109, 141, 41, 130, 140, 5, 159, 100, 11, 187, 24, 89, 66, 8, 172, 175
4	28, 133, 94, 42, 169, 82, 184, 106, 108, 143, 180, 166, 146, 79, 1, 119, 192, 149, 160, 188
5	147, 36, 171, 179, 62, 0, 27, 157, 98, 118, 20, 158, 156, 142, 77, 30, 154, 17, 59, 181
6	114, 127, 139, 191, 93, 151, 21, 55, 16, 152, 91, 99, 120, 197, 74, 190, 161, 144, 196, 87
7	90, 84, 18, 97, 101, 125, 164, 135, 61, 81, 68, 129, 56, 19, 86, 70, 60, 34, 40, 138
8	76, 153, 26, 32, 195, 96, 83, 110, 105, 73, 117, 150, 145, 155, 198, 136, 39, 49, 186, 132
9	50, 52, 80, 185, 121, 189, 46, 88, 69, 67, 183, 58, 33, 38, 103, 51, 107, 170, 4, 102
10	167, 37, 116, 124, 148, 31, 63, 47, 194, 95, 177, 162, 7, 104, 75, 43, 22, 72, 15, 168