RELIABLE REPLAY THROUGH SPATIAL COHERENCE IN ONLINE CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continually adapting models to new tasks requires retaining earlier knowledge under limited memory and computation. Experience replay addresses this challenge, but priorities based on individual loss increases overlook how related memories respond to the same update and can overemphasize isolated responses. We introduce *SPatial coHErent risk control for REplay* (SPHERE), a general replay-allocation method applicable across a broad range of learning settings. SPHERE uses a representation kernel to aggregate signed prospective loss changes, attenuating unsupported spikes while retaining coherent increases. It then formulates allocation as entropy-regularized transport, redistributing uniform source mass toward supported high-risk regions while penalizing long-distance transfers. We derive replay coefficients from the transport objective's sensitivity to the original loss changes and blend them with uniform replay to maintain baseline rehearsal. Our analysis establishes conditions under which kernel aggregation improves risk estimation and bounds transport-value inflation due to residual noise and smoothing bias. Experiments demonstrate that SPHERE improves accuracy and reduces forgetting across noisy-label vision tasks, continual language-model instruction tuning, and code-generation reinforcement learning with incomplete test rewards.

1 INTRODUCTION

Online continual learning requires models to acquire new capabilities while retaining those learned earlier. The requirement spans supervised learning and language-model post-training, including instruction tuning and reinforcement learning for generation tasks. Although their training objectives differ, sequential updates can improve performance on current tasks while degrading earlier capabilities. Experience replay (Rolnick et al., 2019) addresses this tension by revisiting stored examples or problems alongside new data (Kirkpatrick et al., 2017; Li & Hoiem, 2017; Buzzega et al., 2020). However, limited memory and computation allow only part of the history to be rehearsed at each update. Effective replay therefore depends on how rehearsal is allocated across past experience under the learning objective in use.

A common allocation signal is the loss increase that a stored example would experience after a virtual update on the current batch. The same construction can use a supervised loss or a policy-learning surrogate. However, treating each increase independently overlooks whether related memories are affected together. Interference-based retrieval methods use the loss increase to assess a memory's vulnerability and guide replay prioritization (Aljundi et al., 2019a; Li et al., 2024). Such pointwise increments quantify how strongly individual memories respond, but not whether nearby memories respond similarly. Under unreliable supervision, the increase may also reflect conflict with an incorrect target rather than a decline in actual task performance. This raises a complementary question:

When does an observed loss increase justify higher replay priority?

Our insight is to assess each loss increase using nearby memories' responses to the same update. When the underlying response is locally smooth, *spatially coherent increases* support a shared disturbance, whereas an *isolated spike* surrounded by unchanged or decreasing losses has weaker support. Moderate but mutually supported increases can therefore receive higher replay priority than a larger isolated response. Spatial information thus changes the risk criterion: an increment is assessed not only by its magnitude, but also by its agreement with neighboring responses.

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.

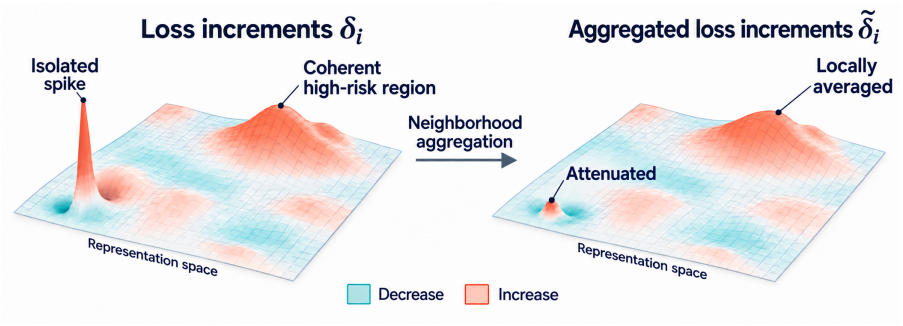

Figure 1: Isolated Spike and Coherent Forgetting.

Label noise is exactly one possible source of such isolated spike. An update may increase loss against a corrupted target while improving prediction of the true class. Prioritizing such increases may then reinforce incorrect supervision. Existing methods reduce the influence of noisy labels through purification and robust training (Kim et al., 2021; Bang et al., 2022; Karim et al., 2022; Millunzi et al., 2024). However, if such corrupted examples remain in memory, their loss increases can be mistaken for forgetting that needs to be prevented. Extra replay can then reinforce incorrect targets and consume capacity needed to preserve useful knowledge. A similar problem arises in reinforcement learning with imperfect rewards. In code generation, incomplete unit tests can reward incorrect programs (Le et al., 2022). If a later update reduces the likelihood of such programs, using replay to reverse that change could preserve faulty behavior. To address this problem, spatial aggregation can reduce the influence of isolated loss increases before replay allocation. It checks whether nearby memories show similar loss changes under the same update rather than directly identifying changes in clean-label prediction.

Building on this insight, we introduce *SPatial coHErent risk control for REplay* (SPHERE), a general replay-allocation method. SPHERE aggregates loss changes within representation neighborhoods to attenuate unsupported spikes and retain coherent increases. Entropy-regularized transport then redistributes uniform mass toward supported high-risk regions while penalizing long-distance transfers. The resulting replay coefficients are blended with uniform replay to maintain baseline rehearsal. SPHERE is broadly applicable across tasks and training pipelines, without being tied to a particular model architecture or learning objective. Experiments show gains in accuracy and retention across noisy-label vision tasks, continual LLM instruction tuning, and code-generation reinforcement learning with incomplete test rewards. Our contributions are as three-fold:

- We introduce SPHERE, a general replay-allocation method applicable across a broad range of learning settings. It combines neighborhood aggregation with entropy-regularized transport to direct replay toward supported high-risk regions while penalizing long-distance mass transfers.
- We derive replay coefficients from the transport objective's sensitivity to the original loss changes, linking risk estimation to replay allocation. We establish conditions under which neighborhood aggregation improves risk estimates and bound transport-value inflation due to residual noise and smoothing bias.
- We demonstrate gains in accuracy and retention across noisy-label vision tasks and continual LLM post-training, including code-generation reinforcement learning with incomplete test rewards. Component ablations support the roles of both neighborhood information and transport-based allocation.

2 Related Works

Online Continual Learning. Online continual learning (OCL) adapts to nonstationary streams in one pass with bounded memory, often without task boundaries. Task-free formulations and memory-based replay provide foundations (Aljundi et al., 2019b; Rolnick et al., 2019; Chrysakis & Moens, 2020; Prabhu et al., 2020). Work on memory selection and retrieval uses prospective interference, Shapley valuation, and memory editing (Aljundi et al., 2019a; Shim et al., 2021; Jin et al., 2021). Other methods change replay representations and objectives through dark knowledge, contrastive learning, asymmetric classification, and evolving prototypes (Buzzega et al., 2020; Mai et al., 2021;

Caccia et al., 2022; De Lange & Tuytelaars, 2021), as well as mutual information, complementary learning systems, and extended distillation (Guo et al., 2022; Arani et al., 2022; Boschini et al., 2022). OCL has also expanded to lifelong language learning and parameter-efficient continual instruction tuning (de Masson D'Autume et al., 2019; Wang et al., 2023), while recent studies examine realistic online evaluation and generalization challenges (Ghunaim et al., 2023; Wang et al., 2024). Building on prospective interference, SPHERE assesses whether nearby memories show similar signed loss changes under the same update before assigning extra replay priority.

Catastrophic Forgetting. Catastrophic interference is the loss of earlier competence after sequential updates (McCloskey & Cohen, 1989; Goodfellow et al., 2013). Regularization protects important parameters through Fisher or synaptic importance (Kirkpatrick et al., 2017; Zenke et al., 2017; Aljundi et al., 2018), or preserves model functions through distillation and functional constraints (Li & Hoiem, 2017; Pan et al., 2020). Gradient-based methods use episodic constraints and meta-replay (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019; Riemer et al., 2019), or restrict updates to orthogonal subspaces (Farajtabar et al., 2020; Saha et al., 2021). Architectural methods isolate or expand capacity (Rusu et al., 2016; Serra et al., 2018; Yoon et al., 2018). Diagnostic studies connect forgetting to example-level events, training regimes, task semantics, and representation change (Toneva et al., 2019; Mirzadeh et al., 2020; Ramasesh et al., 2021). Under noisy supervision, methods also adapt memory management or training to unreliable labels (Kim et al., 2021; Bang et al., 2022; Karim et al., 2022; Millunzi et al., 2024). SPHERE focuses on whether a prospective loss increase has support from nearby memories, since an isolated increase may be a poor reason to allocate more replay.

Transport-Based Allocation. Classical optimal transport (OT) couples two prescribed marginals at minimum cost, and entropic regularization makes the coupling smooth and tractable (Cuturi, 2013; Genevay et al., 2016; Sun & Liu, 2026). In continual learning, transport has been used to evolve replay memories and retain diverse stored examples (Wang et al., 2022; Ye et al., 2025), align event-class predictions with pretrained language-model information (Dao et al., 2024), and transport classifiers between old and new classes (Zhou et al., 2021). These approaches leave open how to distinguish loss increases shared by neighboring memories from isolated responses when allocating replay. SPHERE addresses this problem by aggregating neighboring loss changes before transport allocation, directing replay toward supported forgetting risks while limiting the influence of isolated spikes.

3 Replay-Priority Reliability

Replay allocates limited training capacity to previously observed examples. The key decision is which memories need additional rehearsal under the current update. At step t, the learner receives a mini-batch B_t and maintains a bounded memory $\mathcal{M}_t = \{z_i\}_{i=1}^{m}$ with $m \geq 1$, where $z_i = (x_i, \widetilde{y}_i)$ records an input and its stored supervision. Let $L_t(\theta)$ be the current-batch loss, $\ell(z_i; \theta)$ the memory loss, and $\lambda \geq 0$ the replay strength. This formulation accommodates different tasks and training pipelines, including supervised learning, contrasive loss in LLM, and reinforcement learning. Uniform experience replay (ER) minimizes

$$L_t(\theta) + \frac{\lambda}{m} \sum_{i=1}^{m} \ell(z_i; \theta). \tag{1}$$

To assess which memories are vulnerable to a candidate update direction g, consider a virtual step of size $\eta > 0$ and define

$$\delta_i = \ell(z_i; \theta_t - \eta g) - \ell(z_i; \theta_t). \tag{2}$$

A positive increment means that the update increases the loss on a stored example under its stored supervision. Ranking examples by this increment alone ignores how similar examples respond. It can therefore favor one isolated large increase over several moderate increases shared by neighboring memories. To account for this, we examine neighboring memories' loss changes under the same update. When nearby memories tend to respond similarly, their losses rising together provides stronger support for extra replay. If neighboring losses barely change or decrease, that support is weaker. We therefore assess each loss increase together with its neighbors before deciding how much replay it should receive.

Let $h_i = \phi(x_i)$ be an input-derived representation and let $d(h_i, h_j)$ measure separation in that representation space. Throughout, d is finite, nonnegative, symmetric, and satisfies $d(h, h) = 0$; the analysis does not require a triangle inequality. At a fixed scoring state and candidate direction, we model the signed increments as

$$\delta_i(g) = F_t(h_i; g) + \xi_i. \tag{3}$$

Here $F_i = F_t(h_i; g)$ is the conditional mean response at memory i, $F = (F_1, \ldots, F_m)$ collects these responses, and ξ_i is the residual variation around that mean. Write $\mathbb{E}_t$ and $\mathbb{P}_t$ for expectation and probability conditional on the fixed scoring state and geometry, denoted by $\mathcal{G}_t$. Appendix B specifies this conditioning and gives a sufficient condition for the following spatial assumption.

Assumption 1 (Spatial coherence). *For some $L_F \geq 0$, the conditional mean responses satisfy $|F_i - F_j| \leq L_F d(h_i, h_j)$ for all memories i, j.*

Assumption 2 (Idiosyncratic score noise). *Conditional on $\mathcal{G}_t$, the residuals are independent, centered, and σ-sub-Gaussian: $\mathbb{E}_t[e^{a\xi_i}] \leq e^{\sigma^2 a^2/2}$ for every $a \in \mathbb{R}$, with $\sigma > 0$.*

These assumptions express when neighboring scores provide useful corroboration: proximity limits variation in the mean response, and averaging can reduce residual variation that is not shared across memories. We then introduce the two kinds of forgetting as shown in Figure 1.

Unsupported high-score spike. Suppose $\delta_i(g)$ is large and positive, while nearby memories have small or negative increments. Assumption 1 limits how much their mean loss changes can differ. Any observed difference beyond this bound must come from residual variation. The large increment at z_i therefore has little support from its neighbors that the whole region needs additional replay.

Spatially coherent high-risk region. Suppose instead that several nearby memories have positive increments of similar size. Their agreement supports a shared increase in the mean loss response. Under Assumption 2, independent, zero-mean residuals become less likely to produce a large positive average as more memories contribute. Consistent increases across neighbors therefore provide stronger evidence for allocating additional replay to the region.

4 SPHERE: SPATIALLY COHERENT RISK CONTROL FOR REPLAY

In this section, we introduce SPHERE, a general method for replay allocation compatible with different base models and training pipelines. It combines neighborhood aggregation with transport to compute replay weights, which are blended with uniform replay. SPHERE supports various settings, such as supervised learning, LLM instruction tuning, and reinforcement learning through their corresponding training losses.

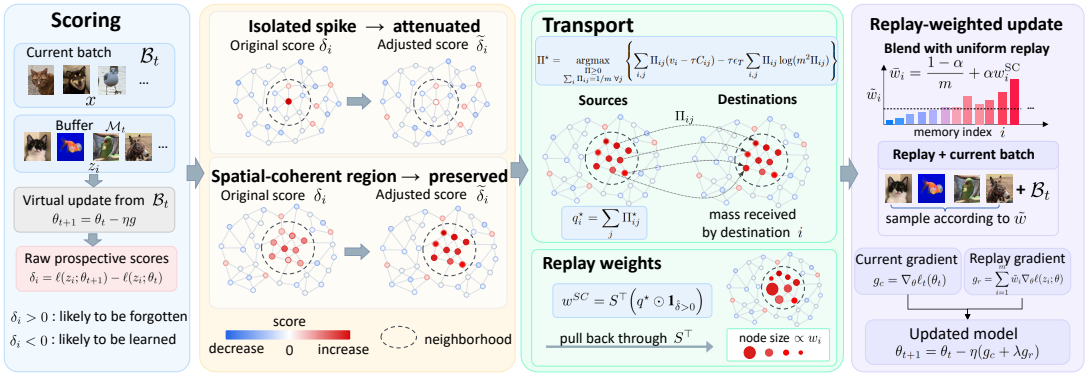

Figure 2: Overview of SPHERE.

4.1 CONSENSUS CORROBORATES PROSPECTIVE RISK

We use a representation kernel to determine how strongly neighboring memories contribute to each risk estimate. We score memories using the current-batch direction $g_c = \nabla_\theta L_t(\theta_t)$ and write $\delta_i =$

$\delta_i(g_c)$ and $\delta = (\delta_1, \ldots, \delta_m)^\top$. Holding the representation geometry fixed during scoring, we define

$$K_{ij} = \exp\left(-\frac{C_{ij}}{\epsilon_s}\right), \qquad S_{ij} = \frac{K_{ij}}{\sum_{k=1}^{m} K_{ik}}, \qquad C_{ij} = d(h_i, h_j), \tag{4}$$

where $\epsilon_s > 0$ controls the spatial scale of aggregation. Nearby memories receive larger weights, while distant responses have less influence. Kernel weights depend on representation distance, whereas the learning objective enters through the loss changes. This separation provides a common aggregation rule across learning settings.

The row-stochastic matrix S combines signed responses before retaining their positive part $\widetilde{\delta}_i = \sum_{j=1}^{m} S_{ij}\delta_j$, where $\widetilde{\delta} = S\delta$ and $\widetilde{s} = [\widetilde{\delta}]_+$. Thus, each risk estimate incorporates both the individual response and the evidence supplied by nearby memories. The kernel determines their relative influence. Let $n_i = \sum_{j \neq i} K_{ij}$ denote the neighboring kernel mass and $a_i = \sum_{j \neq i} K_{ij}\delta_j / n_i$ its weighted response when $n_i > 0$. Since $K_{ii} = 1$,

$$\widetilde{\delta}_i = \frac{\delta_i + n_i a_i}{1 + n_i} = a_i + \frac{\delta_i - a_i}{1 + n_i}. \tag{5}$$

For $n_i = 0$, set $\widetilde{\delta}_i = \delta_i$. Small neighboring mass preserves more of the individual response, whereas larger mass contracts its deviation from neighboring evidence. If $\delta_i > 0$ and $a_i \leq 0$, then $\widetilde{s}_i \leq \delta_i/(1 + n_i)$; if $a_i = \delta_i > 0$, the increase is preserved exactly.

The resulting estimator is useful when the reduction in residual variation outweighs the bias introduced by averaging responses across the neighborhood. Let $r_i = \sum_j S_{ij} d(h_i, h_j)$ be the local geometric radius. The following result makes the variance reduction and its geometric cost explicit.

Proposition 1 (Geometry-controlled risk estimation)**.** *Under Assumptions 1–2,*

$$\begin{aligned} \left|\mathbb{E}_t[\widetilde{\delta}_i] - F_i\right| &\leq L_F r_i, \\ \mathbb{E}_t[(\widetilde{\delta}_i - F_i)^2] &\leq L_F^2 r_i^2 + \sigma^2 \sum_j S_{ij}^2, \\ \mathbb{E}_t[([\widetilde{\delta}_i]_+ - [F_i]_+)^2] &\leq L_F^2 r_i^2 + \sigma^2 \sum_j S_{ij}^2. \end{aligned} \tag{6}$$

The kernel controls both the geometric radius r_i and the effective neighborhood size $N_i^{\text{eff}} = \frac{1}{\sum_j S_{ij}^2} = \frac{(\sum_j K_{ij})^2}{\sum_j K_{ij}^2}$. The estimation bound therefore takes the form $L_F^2 r_i^2 + \sigma^2 / N_i^{\text{eff}}$. Distributing kernel weight across more memories reduces the residual term, but averaging across dissimilar responses can increase the geometric bias. More neighbors help only if their latent responses remain sufficiently similar. Specifically, with $b_i = \sum_j S_{ij}(F_j - F_i)$ and $v_j = \mathbb{E}_t[\xi_j^2]$, consensus has smaller mean-squared error than the raw score exactly when $b_i^2 < v_i - \sum_j S_{ij}^2 v_j$. Thus, corroboration is useful when variance reduction exceeds smoothing bias, not merely when the buffer is large. Appendix D gives the proof and exact error decomposition. The same kernel also determines how the transport objective's sensitivity is assigned back to the memories contributing to each risk estimate, which we will discuss in next section.

4.2 Transport maps supported risk to replay coefficients

Signed consensus reduces isolated score deviations, but residual spikes can still dominate allocation based only on magnitude. We therefore redistribute uniform mass over the historical buffer with a distance penalty. A distant location receives more mass from a source than a nearby competitor only when its advantage in supported risk offsets the extra distance cost, limiting how residual outliers redirect allocation across the memory.

For input scores $v \in \mathbb{R}^m$, evaluated at $v = \widetilde{s} = [S\delta]_+$, each memory j supplies mass $1/m$ and Π_{ij} assigns mass to location i:

$$\mathcal{V}_{\text{OT}}(v) = \max_{\substack{\Pi \geq 0 \\ \sum_i \Pi_{ij} = 1/m \ \forall j}} \left\{ \sum_{i,j} \Pi_{ij}(v_i - \tau C_{ij}) - \tau\epsilon_T \sum_{i,j} \Pi_{ij} \log(m^2 \Pi_{ij}) \right\}, \tag{7}$$

where $C_{ij} = d(h_i, h_j)$, $\tau > 0$ scales the distance penalty, $\epsilon_T > 0$ sets the relative entropy scale. The term $v_i - \tau d(h_i, h_j)$ balances the neighborhood-supported risk $v_i = [\sum_k S_{ik}\delta_k]_+$ at destination i against the cost of reallocating mass from historical memory j. A distant outlier therefore receives more mass from a source than a nearby competitor only when its score advantage exceeds the additional distance cost, discouraging residual spikes from redirecting allocation across unrelated memories. Destination masses remain free to favor high-risk locations, while entropy smooths the allocation. The unique solution is

$$\Pi^*_{ij}(v) = \frac{1}{m} \frac{\exp((v_i - \tau C_{ij})/(\tau\epsilon_T))}{\sum_{k=1}^{m} \exp((v_k - \tau C_{kj})/(\tau\epsilon_T))}. \tag{8}$$

Writing $q^*_i(v) = \sum_j \Pi^*_{ij}(v)$, we have $\nabla_v \mathcal{V}_{\mathrm{OT}}(v) = q^*(v)$. Differentiating through consensus and clipping gives the sensitivity to the raw increments,

$$w^{\mathrm{SC}} = S^\top \left(q^*(\widetilde{s}) \odot \mathbf{1}_{\{\widetilde{\delta}>0\}} \right). \tag{9}$$

Here $\odot$ denotes elementwise multiplication. The $S^\top$ pullback assigns sensitivity at each positive aggregate to the memories that contributed to it. It follows from differentiating the neighborhood-based risk with respect to the original loss changes, rather than from an independently chosen weighting rule. Appendix E specifies the boundary subgradient. Each memory receives sensitivity from the locations with positive consensus it contributes to; the boundary subgradient is specified in Appendix E. Since $w^{\mathrm{SC}} \geq 0$ and $\|w^{\mathrm{SC}}\|_1 \leq 1$, we preserve baseline rehearsal by blending with uniform replay for $0 \leq \alpha < 1$:

$$\bar{w}_j = \frac{1-\alpha}{m} + \alpha w^{\mathrm{SC}}_j, \qquad s_w = \sum_j \bar{w}_j \in [1-\alpha, 1]. \tag{10}$$

With these coefficients held fixed, the uniform-ER host uses

$$\theta_{t+1} = \theta_t - \eta \left[g_c + \lambda \sum_{j=1}^{m} \bar{w}_j \nabla_\theta \ell(z_j; \theta_t) \right]. \tag{11}$$

Appendix A gives unbiased sampling and host interfaces. And the transport value inherits the estimation bound at fixed scoring geometry. For each different task, it is flexible to choose the corresponding loss function ℓ.

Theorem 1 (Noise control for the transport value)**.** *Under Assumptions 1–2,*

$$\mathbb{E}_t[\mathcal{V}_{\mathrm{OT}}(\widetilde{s})] \leq \mathcal{V}_{\mathrm{OT}}([F]_+) + L_F \max_i r_i + \sigma \sqrt{2\log(2m) \max_i \sum_j S^2_{ij}}. \tag{12}$$

The bound controls inflation of the scalar transport value through geometric bias and residual noise; systematic corruption can remain in F. Appendix F gives the proof, and Appendix G establishes local descent for virtual-parameter replay under its stated conditions.

5 Experiments

5.1 Continual language model post-training

We first evaluate whether SPHERE preserves earlier knowledge during continual language model post-training. The model sequentially learns text classification tasks covering sentiment analysis, topic classification, and natural language inference, while replaying stored examples to retain knowledge from earlier tasks. We use the 15 classification tasks of O-LoRA (Wang et al., 2023), expressed as instructions with verbalized labels and evaluated by exact match, without adding synthetic label corruption. T5-Large is adapted with LoRA of rank eight in a single pass of 860 updates with stream batches of 16.

We compare SPHERE with no replay, ER, Loss-prop, MIR-prop, and MIR-top using buffers of 300 and 3000 and replay batches of 4. Replay methods use a 512-sample candidate pool refreshed from

Table 1: Replay performance with limited memory and replay. Values are mean±SD except Min.

Method	Buffer size 300				Buffer size 3000			
	FP ↑	Worst Forget ↓	Tail Forget ↓	Min. ↑	FP ↑	Worst Forget ↓	Tail Forget ↓	Min. ↑
No replay	.415±.146	.811±.156	.693±.173	.130	.415±.146	.811±.156	.693±.173	.130
ER	.661±.075	.245±.149	.200±.137	.423	.641±.071	.329±.143	.269±.112	.479
Loss-prop	.689±.045	.188±.061	**.142±.043**	.538	.618±.050	.325±.085	.269±.076	.502
MIR-prop	.675±.045	.188±.083	.152±.072	.570	.635±.042	.296±.111	.237±.079	.561
MIR-top	.621±.023	.244±.055	.211±.060	.588	.633±.066	.375±.205	.285±.155	.500
SPHERE	**.690±.021**	**.180±.053**	.143±.035	**.661**	**.658±.028**	**.291±.093**	**.228±.067**	**.619**

the buffer every four replay events. ER samples uniformly, Loss-prop samples proportionally to loss, and MIR-prop samples proportionally to positive one-step interference $[\delta_i]_+$. MIR-top selects the top 16 interference scores, following the original MIR rule (Aljundi et al., 2019a). Across both buffer sizes, Table 1 shows that SPHERE performs higher FP, less forgetting and lower observed variation than ER and the MIR variants. The results support using neighboring loss responses to guide replay allocation when only a small portion of past data can be rehearsed.

We next evaluate the neighborhood construction as a whole. Setting $S = I$ removes signed aggregation and its corresponding $S^\top$ pullback while retaining transport with the original distance cost. In Table 2, full SPHERE improves final accuracy by 4.42 percentage points and reduces worst forgetting by 7.90 points, while also improving new-task accuracy. These gains support the joint use of neighborhood risk estimates and their sensitivity-based replay coefficients.

Table 2: Matched neighborhood-block ablation (%).

Metric	$S = I$	SPHERE	Gain
Final acc. ↑	64.90	**69.32**	+4.42
Incremental acc. ↑	66.08	**68.54**	+2.47
New task acc. ↑	70.34	**72.43**	+2.08
Mean forgetting ↓	8.88	**6.18**	+2.70
Worst forgetting ↓	24.75	**16.85**	+7.90
Tail forgetting ↓	20.59	**13.54**	+7.05

Table 3: Component controls with scarce memory and replay. Values are mean±SD except Min.

Variant	FP ↑	Worst ↓	Tail ↓	Min. ↑
Full SPHERE	**.690±.021**	**.180±.053**	**.143±.035**	**.661**
Consensus only	.680±.021	.189±.054	.154±.030	.647
No transport	.640±.063	.275±.129	.227±.110	.478
Permuted geometry	.666±.056	.213±.082	.188±.078	.497
Matched MIR	.682±.048	.192±.113	.161±.087	.517
No uniform floor	.673±.034	.230±.082	.183±.063	.608

Table 3 examines allocation after neighborhood aggregation. Consensus only replaces transport with an ESS-matched softmax of the same supported scores, retaining the positive-response mask, $S^\top$ pullback, and uniform mixture. Full SPHERE achieves 0.690 final performance versus 0.680, with lower worst and tail forgetting. The lower reported mean with permuted geometry further supports the importance of alignment between memories and their geometry. All instruction-tuning controls use unit-mass replay, so these gains do not require explicit replay-strength scaling.

5.2 Noisy-label continual learning

Corrupted labels make an individual loss increase ambiguous, since rehearsing an incorrect target can reduce accuracy on clean test data. We next test the performace of SPHERE on conventional classfication task to test whether neighborhood evidence improves performance across different training objectives. We integrate SPHERE with ER, ER-ACE (Caccia et al., 2022), and X-DER (Boschini et al., 2022). Each integration retains its host's losses and memory policy. MIR-prop (Aljundi et al., 2019a) and DER++ (Buzzega et al., 2020) provide additional replay baselines. We also evaluate SPHERE within PuriDivER's purity estimation and sample-partition rules (Bang et al., 2022). PuriDivER, GDumb (Prabhu et al., 2020), and ABS-based (Millunzi et al., 2024) use additional memory fitting and are reported separately. Split CIFAR-100 and TinyImageNet each contain ten tasks, with memory capacities of 2,000 and 4,000 examples, respectively. Online training uses a single pass with 0, 20, 40, or 60% symmetric label noise. The main evaluation measures classification accuracy on clean test data without task identity. Experiment details and additional results on MNIST and CIFAR-10 appear in Appendix H.

Table 4 shows that SPHERE improves all three replay hosts on both datasets under moderate and high label noise. The gains also extend to PuriDivER, which uses purity estimates to partition stored examples and guide training under corrupted supervision. The SPHERE integrations improve the evaluated replay hosts under moderate and high label

Table 4: Replay across hosts under symmetric label noise.

Method	CIFAR-100 ($K = 2{,}000$)				TinyImageNet ($K = 4{,}000$)			
	Clean	20%	40%	60%	Clean	20%	40%	60%
ER	19.6±3.4	14.3±1.1	10.2±1.1	5.1±0.8	16.3±1.2	12.0±0.6	7.4±0.6	3.3±0.4
ER + SPHERE	**20.9±1.8**	15.7±1.2	12.2±0.8	8.0±0.7	15.3±1.4	13.5±0.9	10.1±0.6	6.1±0.7
MIR-prop	19.8±1.4	15.0±2.3	10.2±1.6	6.2±0.8	17.1±0.5	11.6±1.1	8.0±0.8	4.7±0.2
DER++	15.3±4.1	10.9±4.1	9.0±2.0	5.6±1.0	9.8±1.0	7.4±0.9	4.8±0.7	3.4±0.3
ER-ACE	20.4±3.4	16.8±2.1	12.9±1.5	8.1±1.1	17.9±1.7	14.4±0.8	9.6±0.8	5.5±0.4
ER-ACE + SPHERE	18.7±2.9	16.6±1.6	13.5±1.6	10.4±0.6	16.6±1.4	14.2±0.6	11.0±1.1	6.8±0.6
X-DER	20.2±5.0	16.2±6.1	14.1±1.8	8.3±2.4	17.9±1.6	17.5±1.0	10.4±1.2	6.2±1.6
X-DER + SPHERE	20.7±4.6	**19.6±3.0**	**16.1±1.6**	**10.9±1.1**	**19.4±1.1**	**18.2±0.8**	**12.8±1.1**	**8.2±0.4**
Separate buffer training								
GDumb	**25.2±1.0**	17.7±0.9	12.5±0.6	7.4±0.5	**19.4±0.3**	13.0±0.5	7.7±0.7	4.3±0.2
ABS-based	–	32.9±0.7	32.6±0.8	24.9±0.9	–	26.8±0.8	23.4±0.8	16.2±0.3
PuriDivER	–	38.6±0.7	37.2±1.5	32.2±0.6	–	30.1±0.3	28.7±0.3	24.6±0.2
PuriDivER + SPHERE	–	**39.2±2.0**	**39.1±1.1**	**36.2±1.5**	–	**30.8±0.1**	**31.9±1.4**	**27.4±1.3**

noise. The improvements across hosts support its broader use in replay allocation, while gains under cleaner supervision depend on the host. We next examine whether the gains extend beyond final classification accuracy. Class-IL and Task-IL measure accuracy without and with task identity, respectively, while AAA summarizes accuracy throughout training. We also report average forgetting (avgF), worst-class accuracy, and tail forgetting ($\mathrm{CVaR}_{0.2}$) to assess retention beyond the average prediction. Table 5 shows that adding SPHERE to ER-ACE improves every reported metric. The gains in worst-class accuracy and tail forgetting support the use of neighborhood information to protect knowledge that is most affected by subsequent updates.

Table 5: CIFAR-10 at 20% noise, $K = 500$ (mean±SD).

Method	Class-IL↑	Task-IL↑	AAA↑	avgF↓	worst-class↑	$\mathrm{CVaR}_{0.2}$ ↓
ER	33.4±4.5	76.1±5.1	51.4±4.8	42.9±5.6	5.6±4.5	72.5±5.7
ER-ACE	36.1±4.5	77.9±6.5	53.2±2.1	25.3±5.5	11.1±7.2	54.8±8.9
GDumb	36.8±0.9	77.0±2.8	28.1±0.7	20.2±7.4	8.4±5.4	51.2±19.0
DER++	28.2±4.3	80.7±4.8	48.4±4.7	57.5±9.3	1.3±1.7	81.3±7.0
MIR	32.1±4.4	75.6±4.1	48.5±3.5	36.9±7.0	5.5±3.6	71.1±6.9
X-DER	28.1±8.2	75.1±8.0	44.9±5.3	25.0±9.9	0.0±0.0	60.0±16.1
ER-ACE + SPHERE	**38.8±4.2**	**81.0±4.5**	**54.6±3.1**	**15.3±6.7**	**19.2±2.4**	**46.5±10.8**

Finally, we examine whether neighborhood information remains useful when labels are clean. On clean Split CIFAR-10, we set $S = I$ and jointly remove neighborhood averaging and its associated weight pullback while retaining the other components. Table 6 shows that full SPHERE reduces mean and severe forgetting, alongside higher final accuracy and similar stage-average accuracy. The results support neighborhood information as a means of preserving earlier knowledge even without label corruption.

Table 6: Neighborhood-block ablation on clean Split CIFAR-10.

Metric	SPHERE	$S = I$
Final acc ↑	**46.61 ± 2.19**	45.42±2.99
Stage-average acc ↑	**59.01 ± 3.00**	58.93±3.30
Mean forgetting ↓	**35.10 ± 5.90**	36.55±7.71
Worst forgetting ↓	**62.66 ± 9.63**	69.85±12.42
Top-2 forgetting ↓	**59.53 ± 8.81**	61.84±10.69

5.3 Code generation with incomplete test rewards

Beyond classification, we further implement whether SPHERE can preserve earlier skills during continual reinforcement learning for code generation. For a problem x, the policy π_θ generates a program y token by token and receives the terminal reward $r^{\mathrm{vis}}(x, y) = \mathbf{1}\{y \text{ passes the visible test for } x\}$. A rewarded program can still fail other tests, so replay must allocate limited capacity using loss changes derived from incomplete feedback. The expected reward and its policy gradient are

$$J_t(\theta) = \mathbb{E}_{x\sim\mathcal{D}_t,\, y\sim\pi_\theta(\cdot|x)}[r^{\mathrm{vis}}(x, y)],$$
$$\nabla_\theta J_t(\theta) = \mathbb{E}\left[\left(r^{\mathrm{vis}}(x, y) - b_t(x)\right)\sum_{u=1}^{|y|}\nabla_\theta \log \pi_\theta(y_u \mid x, y_{<u})\right], \tag{13}$$

where $\mathcal{D}_t$ is the current problem distribution and $b_t(x)$ is a problem-dependent baseline.

We construct four successive stages from KodCode-V1 (Xu et al., 2025), covering Package, Data Structure/Algorithm, Leetcode, and Codeforces/Apps. After removing near duplicates, each stage contains 120 training problems and 100 disjoint evaluation problems. We adapt Qwen2.5-Coder-1.5B-Instruct (Hui et al., 2024) with rank-16 LoRA. Two passes over each stage give 30 updates per stage and 120 in total, with eight current problems per update. Evaluation uses greedy decoding and requires passing the full test suite, which is unavailable to training and replay selection.

We optimize the policy with a GRPO-style clipped loss (Shao et al., 2024; Liu et al., 2025). For each problem x_i, we sample a group $\mathcal{G}_i$ of eight programs and compute $A_{ik} = r_{ik} - \frac{1}{8}\sum_{\ell=1}^{8} r_{i\ell}$, where $r_{ik} = r^{\text{vis}}(x_i, y_{ik})$. The token-level surrogate weights policy probability ratios by these centered advantages, with clipping to limit the incentive for large probability changes. Let $\mathcal{L}_i^{\text{GRPO}}(\theta; \mathcal{G}_i)$ denote the resulting group loss. Only groups containing both reward outcomes are scored, since all advantages vanish otherwise. SPHERE assesses whether an update on current problems would increase the policy loss on an earlier problem. We take a AdamW step to obtain θ_t^{v} and compute $\delta_i = \mathcal{L}_i^{\text{GRPO}}(\theta_t^{\text{v}}; \mathcal{G}_i) - \mathcal{L}_i^{\text{GRPO}}(\theta_t; \mathcal{G}_i)$, holding the sampled programs and advantages fixed across both evaluations. SPHERE combines these changes across nearby prompt embeddings before allocating replay weights. All replay methods store up to 480 problems, assess 48 candidates per update, and replay eight groups with unit total weight. Hyperparameters are selected on disjoint development data and seeds.

Table 7 shows SPHERE performs stronger final accuracy and retention than for uniform replay and interference-based retrieval, alongside positive acquisition gains. The results suggest that the replay controller can help balance retention and adaptation when policy updates rely on incomplete rewards. Figure 3 complements the final comparison by tracking visible-test reward during training and full-suite accuracy after each stage. Together, the panels relate progress under the training reward to performance under broader correctness checks as the task distribution changes.

Table 7: Code-generation results.

Method	Final ↑	Old ↑	Worst ↑	F ↓	BWT ↑
Uniform replay	42.75±2.92	43.56	40.00	4.00	-2.78
MIR	44.50±1.63	45.78	38.67	4.56	-3.89
No Transport	42.67±1.70	41.78	39.67	6.56	-5.56
Permuted geometry	48.58±1.88	45.78	41.67	3.56	-2.89
SPHERE	**49.08±0.80**	**46.89**	**43.67**	**3.33**	**-1.56**

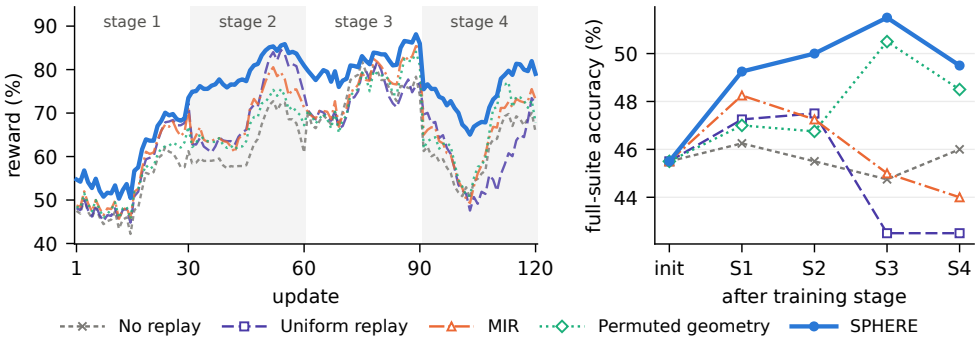

Figure 3: Reward during training (left) and evaluation accuracy after each stage (right).

6 Conclusion

We introduced SPHERE, a general replay-allocation method applicable across a broad range of learning settings. SPHERE combines kernel-based aggregation of signed loss changes with entropy-regularized transport, directing replay toward supported high-risk regions while penalizing long-distance mass transfers. Replay coefficients are derived from the transport objective's sensitivity to the original loss changes and blended with uniform replay to retain baseline rehearsal. Our analysis establishes when neighborhood aggregation improves risk estimation and bounds transport-value inflation due to residual noise and smoothing bias. Experiments demonstrate improved accuracy and retention across noisy-label vision tasks, continual language-model instruction tuning, and code-generation reinforcement learning with incomplete test rewards. However, computing pairwise distances and transport allocations adds overhead as the scored memory set grows, and our future work will explore sparse approximations to reduce this cost in large-scale continual training.

Reproducibility statement

We provide the code in the supplementary material, together with a README file that allows experimental results to be reproduced.

The use of LLM

We used LLM to polish the writing and correct grammar in some paragraphs, but it did not contribute to ideas or conceptual content. Separately, we fine-tuned and evaluated language models in our continual instruction-tuning and code-generation reinforcement-learning experiments, as described in Sections 5.1 and 5.3.

References

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *European conference on computer vision*, pp. 144–161. Springer, 2018.

Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. *Advances in neural information processing systems*, 32, 2019a.

Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11246–11255. IEEE, 2019b.

Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Learning fast, learning slow: A general continual learning method based on complementary learning system. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=uxxFrDwrE7Y.

Jihwan Bang, Hyunseo Koh, Seulki Park, Hwanjun Song, Jung-Woo Ha, and Jonghyun Choi. Online continual learning on a contaminated data stream with blurry task boundaries. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9265–9274. IEEE, 2022.

Matteo Boschini, Lorenzo Bonicelli, Pietro Buzzega, Angelo Porrello, and Simone Calderara. Class-incremental continual learning into the extended der-verse. *IEEE transactions on pattern analysis and machine intelligence*, 45(5):5497–5512, 2022.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

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 *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=N8MaByOzUfb.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-GEM. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Hkf2_sC5FX.

Aristotelis Chrysakis and Marie-Francine Moens. Online continual learning from imbalanced data. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1952–1961. PMLR, 13–18 Jul 2020.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Viet Dao, Van-Cuong Pham, Quyen Tran, Thanh-Thien Le, Linh Van Ngo, and Thien Huu Nguyen. Lifelong event detection via optimal transport. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 12610–12621, 2024.

Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8230–8239. IEEE, 2021.

Cyprien de Masson D'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. Episodic memory in lifelong language learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International conference on artificial intelligence and statistics*, pp. 3762–3773. PMLR, 2020.

Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. Stochastic optimization for large-scale optimal transport. *Advances in neural information processing systems*, 29, 2016.

Yasir Ghunaim, Adel Bibi, Kumail Alhamoud, Motasem Alfarra, Hasan Abed Al Kader Hammoud, Ameya Prabhu, Philip HS Torr, and Bernard Ghanem. Real-time evaluation in online continual learning: A new hope. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11888–11897, 2023.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

Yiduo Guo, Bing Liu, and Dongyan Zhao. Online continual learning through mutual information maximization. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8109–8126. PMLR, 17–23 Jul 2022.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

Xisen Jin, Arka Sadhu, Junyi Du, and Xiang Ren. Gradient-based editing of memory examples for online task-free continual learning. *Advances in Neural Information Processing Systems*, 34: 29193–29205, 2021.

Nazmul Karim, Umar Khalid, Ashkan Esmaeili, and Nazanin Rahnavard. Cnll: A semi-supervised approach for continual noisy label learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3877–3887. IEEE, 2022.

Chris Dongjoo Kim, Jinseo Jeong, Sangwoo Moon, and Gunhee Kim. Continual learning on noisy data streams via self-purified replay. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 517–527. IEEE, 2021.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven Chu Hong Hoi. Coderl: Mastering code generation through pretrained models and deep reinforcement learning. *Advances in Neural Information Processing Systems*, 35:21314–21328, 2022.

Xingyu Li, Bo Tang, and Haifeng Li. Adaer: An adaptive experience replay approach for continual lifelong learning. *Neurocomputing*, 572:127204, 2024.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=5PAF7PAY2Y.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Zheda Mai, Ruiwen Li, Hyunwoo Kim, and Scott Sanner. Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3584–3594. IEEE, 2021.

Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pp. 109–165. Academic Press, 1989. doi: https://doi.org/10.1016/S0079-7421(08)60536-8. URL https://www.sciencedirect.com/science/article/pii/S0079742108605368.

Monica Millunzi, Lorenzo Bonicelli, Angelo Porrello, Jacopo Credi, Petter N Kolm, and Simone Calderara. May the forgetting be with you: Alternate replay for learning with noisy labels. *arXiv preprint arXiv:2408.14284*, 2024.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of training regimes in continual learning. *Advances in neural information processing systems*, 33:7308–7320, 2020.

Pingbo Pan, Siddharth Swaroop, Alexander Immer, Runa Eschenhagen, Richard Turner, and Mohammad Emtiyaz Khan. Continual deep learning by functional regularisation of memorable past. *Advances in neural information processing systems*, 33:4453–4464, 2020.

Ameya Prabhu, Philip H. S. Torr, and Puneet K. Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II*, pp. 524–540, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58535-8. URL https://doi.org/10.1007/978-3-030-58536-5_31.

Vinay Venkatesh Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. In *International Conference on Learning Representations*, 2021.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, , and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1gTShAct7.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=3AOj0RCNC2.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International conference on machine learning*, pp. 4548–4557. PMLR, 2018.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Dongsub Shim, Zheda Mai, Jihwan Jeong, Scott Sanner, Hyunwoo Kim, and Jongseong Jang. Online class-incremental continual learning with adversarial shapley value. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 9630–9638, 2021.

Haixiang Sun and Andrew Liu. Robust generalization with adaptive optimal transport priors for decision-focused learning. In *The 29th International Conference on Artificial Intelligence and Statistics*, 2026. URL https://openreview.net/forum?id=EXv1wBpzaY.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuan-Jing Huang. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10658–10671, 2023.

Xinrui Wang, Chuanxing Geng, Wenhai Wan, Shao-Yuan Li, and Songcan Chen. Forgetting, ignorance or myopia: Revisiting key challenges in online continual learning. *Advances in Neural Information Processing Systems*, 37:58341–58375, 2024.

Zhenyi Wang, Li Shen, Le Fang, Qiuling Suo, Tiehang Duan, and Mingchen Gao. Improving task-free continual learning by distributionally robust memory evolution. In *International conference on machine learning*, pp. 22985–22998. PMLR, 2022.

Zhangchen Xu, Yang Liu, Yueqin Yin, Mingyuan Zhou, and Radha Poovendran. Kodcode: A diverse, challenging, and verifiable synthetic dataset for coding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 6980–7008, 2025.

Fei Ye, Adrian G. Bors, and Kun Zhang. Continual unsupervised generative modelling via online optimal transport. AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025. ISBN 978-1-57735-897-8. doi: 10.1609/aaai.v39i21.34362. URL https://doi.org/10.1609/aaai.v39i21.34362.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Sk7KsfW0-.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pp. 3987–3995. Pmlr, 2017.

Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Co-transport for class-incremental learning. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 1645–1654, 2021.

A CORE IMPLEMENTATION INTERFACE AND LIMITS OF THE GUARANTEES

A fully specified ideal update. Algorithm 1 implements the mathematical controller on a fixed scored set.

Algorithm 1 Core SPHERE on a fixed scored set

Require: Model θ_t, current batch B_t, scored memories $\{z_i\}_{i=1}^m$, input features $\{h_i\}$, parameters $\eta, \epsilon_s, \tau, \epsilon_T > 0$, $\lambda \geq 0$, $\alpha \in [0, 1)$, replay batch size $b \geq 1$.

1: Fix the scored set, input features, stored targets, and scoring randomness.
2: Compute $g_c = \nabla_\theta L_t(\theta_t)$ and virtual parameter $\theta^v = \theta_t - \eta g_c$.
3: Compute $\delta_i = \ell(z_i; \theta^v) - \ell(z_i; \theta_t)$ for each memory.
4: Form $C_{ij} = d(h_i, h_j)$, $K_{ij} = \exp(-C_{ij}/\epsilon_s)$, and row-normalized S.
5: Compute $\widetilde{\delta} = S\delta$ and $\widetilde{s} = [\widetilde{\delta}]_+$.
6: Compute $\Pi^*(\widetilde{s})$ by Eq. (8) and $q_i^* = \sum_j \Pi_{ij}^*$.
7: Set $w^{\mathrm{SC}} = S^\top(q^* \odot \mathbf{1}_{\{\widetilde{\delta}>0\}})$.
8: Set $\bar{w}_i = (1-\alpha)/m + \alpha w_i^{\mathrm{SC}}$, $s_w = \sum_i \bar{w}_i$, and $p_i = \bar{w}_i / s_w$.
9: Draw $J_1, \ldots, J_b$ independently with replacement from p.
10: Form $\widehat{g}_r = (s_w/b) \sum_{r=1}^b \nabla_\theta \ell(z_{J_r}; \theta_t)$, treating all weights as constants.
11: Update $\theta_{t+1} = \theta_t - \eta(g_c + \lambda \widehat{g}_r)$.

Let $\mathcal{H}_t$ contain the current model, scored memories, scoring outcomes, weights, and per-example replay gradients, all fixed before drawing the replay indices. Conditional on $\mathcal{H}_t$, independence and $\mathbb{P}(J_r = i \mid \mathcal{H}_t) = p_i$ give

$$\mathbb{E}[\widehat{g}_r \mid \mathcal{H}_t] = s_w \sum_i p_i \nabla_\theta \ell(z_i; \theta_t) = \sum_i \bar{w}_i \nabla_\theta \ell(z_i; \theta_t). \tag{14}$$

The same identity applies to conditional mean gradients if fresh per-example randomness is used independently and preserves those means.

Normalizing $\bar{w}$ and averaging sampled losses without the factor s_w instead realizes unit-mass replay. If unequal-probability sampling is performed without replacement, its inclusion probabilities need not equal bp_i; an uncorrected minibatch mean is not generally unbiased for the displayed weighted sum. With positive inclusion probabilities π_i^{inc}, the inclusion-probability-corrected form $\sum_{i \in \mathcal{B}} (\bar{w}_i / \pi_i^{\mathrm{inc}}) \nabla \ell_i$ is unbiased, but computing or using those inclusion probabilities is a separate implementation choice. This correction is distinct from the uncorrected without-replacement retrieval used in the reported configuration descriptions.

B CONDITIONAL FIELD MODEL AND SUFFICIENT GRADIENT REGULARITY

This appendix makes explicit what the conditional field represents and when nearby representations can have similar expected responses. Fix a scoring state $\mathcal{G}_t$ containing (θ_t, g), the scored candidate set, the input representations, and any randomness used to construct the geometry. The field values F_i and kernel S are $\mathcal{G}_t$-measurable, and Assumption 2 is imposed conditional on this state. Consequently $F_i = \mathbb{E}_t[\delta_i(g)]$. The assumption concerns residual variation remaining under the conditional model; simply freezing a trained model does not establish it. In particular, historical label noise may become dependent after conditioning on a model trained with those labels.

For a sufficient construction, take a generic example $Z = (X, \widetilde{Y})$ under a fixed family of conditional laws given $\phi(X) = h$ and $\mathcal{G}_t$. Choose versions of these laws and their conditional expectations at the representations under consideration, and suppose the following increments are integrable:

$$F_t(h; g) = \mathbb{E}[\ell(Z; \theta_t - \eta g) - \ell(Z; \theta_t) \mid \phi(X) = h, \mathcal{G}_t]. \tag{15}$$

To use this construction for the buffer, assume that each memory's conditional mean increment equals $F_t(h_i; g)$. This identification is part of the working model. The next proposition supplies a sufficient condition for spatial coherence within that model: if nearby representations have similar expected gradients along the virtual-update path, they also have similar expected loss increments. It does not derive either condition from adaptive buffer construction.

Proposition 2 (Gradient regularity implies spatial coherence)**.** *Similar conditional gradients at nearby representations imply similar expected virtual-update increments. For $s \in [0,1]$, let $\theta_{t,s} = \theta_t - s\eta g$ and*

$$G_t(h;\theta) = \mathbb{E}[\nabla_\theta \ell(Z;\theta) \mid \phi(X) = h, \mathcal{G}_t]\,. \tag{16}$$

Assume that $s \mapsto \ell(Z;\theta_t - s\eta g)$ is almost surely absolutely continuous and has derivative $-\eta\langle \nabla_\theta \ell(Z;\theta_{t,s}), g\rangle$ for almost every $s \in [0,1]$. Assume the gradient integrand is jointly measurable and that

$$\mathbb{E}\left[\int_0^1 \|\nabla_\theta \ell(Z;\theta_{t,s})\|_2 \, ds \mid \phi(X) = h, \mathcal{G}_t\right] < \infty \tag{17}$$

for every representation under consideration. Suppose also that, for every $s \in [0,1]$,

$$\|G_t(h;\theta_{t,s}) - G_t(h';\theta_{t,s})\|_2 \le L_G d(h,h'). \tag{18}$$

Then

$$|F_t(h;g) - F_t(h';g)| \le \eta L_G \|g\|_2 d(h,h'), \tag{19}$$

so Assumption 1 holds with $L_F = \eta L_G \|g\|_2$.

Proof. For a generic example Z, absolute continuity and the assumed path derivative give

$$\ell(Z;\theta_t - \eta g) - \ell(Z;\theta_t) = -\eta \int_0^1 \langle \nabla_\theta \ell(Z;\theta_{t,s}), g\rangle ds. \tag{20}$$

The integrability assumption permits conditional expectation and integration to be interchanged, yielding

$$F_t(h;g) = -\eta \int_0^1 \langle G_t(h;\theta_{t,s}), g\rangle ds. \tag{21}$$

Therefore, by Cauchy–Schwarz and Eq. (18),

$$\begin{aligned} |F_t(h;g) - F_t(h';g)| &\le \eta \int_0^1 \|G_t(h;\theta_{t,s}) - G_t(h';\theta_{t,s})\|_2 \|g\|_2 ds \\ &\le \eta L_G \|g\|_2 d(h,h'), \end{aligned} \tag{22}$$

which proves Eq. (19). □

B.1 Fixed-state symmetric corruption as an illustration

Let the label space contain $C \ge 2$ classes and let $\rho \in [0,1]$. With fixed inputs and clean labels y_i, symmetric corruption independently draws each observed label according to

$$\mathbb{P}(\widetilde{Y}_i = c \mid Y_i = y_i) = \begin{cases} 1-\rho, & c = y_i, \\ \rho/(C-1), & c \ne y_i. \end{cases} \tag{23}$$

The clean label is introduced only to describe the data-generating process and is never available to SPHERE. For a fixed input, model state, and candidate direction, let

$$\delta_i(c;g) = \ell(x_i, c; \theta_t - \eta g) - \ell(x_i, c; \theta_t) \tag{24}$$

denote the increment that would be observed under label c.

The corruption-averaged signed response and its centered residual are

$$\begin{aligned} F_i^{(\rho)}(g) &= (1-\rho)\delta_i(y_i;g) + \frac{\rho}{C-1}\sum_{c \ne y_i} \delta_i(c;g), \\ \xi_i^{(\rho)}(g) &= \delta_i(\widetilde{y}_i;g) - F_i^{(\rho)}(g). \end{aligned} \tag{25}$$

For this illustrative calculation, the inputs, clean labels, scoring model, candidate direction, and geometry are fixed independently of the hypothetical corrupted-label draws. Then $\xi_i^{(\rho)}$ are centered

and independent. If all class-conditioned increments are finite, their finite ranges also imply sub-Gaussian residuals. This gives one possible residual model, but does not establish spatial coherence of $F^{(\rho)}$; that remains a separate requirement. In actual continual training, historical corruption can affect the model and its geometry, so neither conditional independence nor meaningful neighborhoods follows automatically from independent stream-label corruption.

For cross-entropy with positive class probabilities at both scoring states,

$$\delta_i(c; g) = \log \frac{p_{\theta_t}(c \mid x_i)}{p_{\theta_t - \eta g}(c \mid x_i)}. \tag{26}$$

An update that increases the true-class probability and decreases a corrupted-class probability therefore gives a negative clean-target increment and a positive corrupted-target increment. This is a concrete way that the same prediction change can improve clean prediction while appearing harmful under stored supervision. It establishes a possibility, not a rule that every corrupted label creates an isolated spike.

C Concentration of fixed-region and pairwise responses

Let A be a nonempty $\mathcal{G}_t$-measurable region with k examples, chosen without inspecting the residual realizations. Let $\bar{\delta}_A$ and $\bar{F}_A$ denote its average observed and latent increments. The same fixed-choice requirement applies to the pair i, j.

Proposition 3 (Statistical spike–region separation)**.** *Suppose Assumptions 1 and 2 hold. For distinct i, j and every $x > 0$,*

$$\begin{aligned} \mathbb{P}_t(|\delta_i(g) - \delta_j(g)| \geq L_F d(h_i, h_j) + x) &\leq 2e^{-x^2/(4\sigma^2)}, \\ \mathbb{P}_t\big(|\bar{\delta}_A - \bar{F}_A| \geq x\big) &\leq 2e^{-kx^2/(2\sigma^2)}. \end{aligned} \tag{27}$$

These are concentration bounds, not a calibrated classifier of clean examples or an adaptive region-selection guarantee.

Proof. All probabilities and expectations below are conditional on $\mathcal{G}_t$. From Eq. (3),

$$\delta_i(g) - \delta_j(g) = F_i - F_j + \xi_i - \xi_j. \tag{28}$$

By Assumption 1, the event in the first line of Eq. (27) is contained in $\{|\xi_i - \xi_j| \geq x\}$. Conditional independence and Assumption 2 give

$$\begin{aligned} \mathbb{E}_t[e^{u(\xi_i - \xi_j)}] &= \mathbb{E}_t[e^{u\xi_i}]\mathbb{E}_t[e^{-u\xi_j}] \\ &\leq e^{\sigma^2 u^2}. \end{aligned} \tag{29}$$

Thus, $\xi_i - \xi_j$ is $\sqrt{2}\sigma$-sub-Gaussian, and its two-sided tail satisfies

$$\mathbb{P}_t(|\xi_i - \xi_j| \geq x) \leq 2e^{-x^2/(4\sigma^2)}, \tag{30}$$

which proves the first line of Eq. (27).

For a nonempty region A of size k,

$$\bar{\delta}_A - \bar{F}_A = \frac{1}{k} \sum_{i \in A} \xi_i. \tag{31}$$

For every $u \in \mathbb{R}$,

$$\begin{aligned} \mathbb{E}_t\left[\exp\left(\frac{u}{k} \sum_{i \in A} \xi_i\right)\right] &= \prod_{i \in A} \mathbb{E}_t[e^{u\xi_i/k}] \\ &\leq \exp\left(\frac{\sigma^2 u^2}{2k}\right). \end{aligned} \tag{32}$$

Hence, the regional average is $\sigma/\sqrt{k}$-sub-Gaussian, and the standard two-sided tail bound proves the second line of Eq. (27). □

D PROOFS FOR GEOMETRY-SUPPORTED ESTIMATION

D.1 EXACT SPIKE–REGION CONTRAST AND SMOOTHING ORDER

The ranking contrast can occur even for the dense positive kernel in the main text. Take two groups B, A of $k \geq 2$ memories each. All features in B equal h_B, all features in A equal h_A, and $d(h_A, h_B) = D > 0$. Within-group kernel weights are one; cross-group weights are $\varepsilon = \exp(-D/\epsilon_s) \in (0, 1)$. Let one memory $i \in B$ have increment b, let the remaining B increments be zero, and let every A increment be a, where $0 < a < b < ka$.

All rows have denominator $k(1 + \varepsilon)$. For a memory $c \in A$,

$$\begin{aligned} \widetilde{\delta}_i &= \frac{b + \varepsilon ka}{k(1+\varepsilon)}, \\ \widetilde{\delta}_c &= \frac{ka + \varepsilon b}{k(1+\varepsilon)}, \\ \widetilde{\delta}_c - \widetilde{\delta}_i &= \frac{(1-\varepsilon)(ka - b)}{k(1+\varepsilon)} > 0. \end{aligned} \tag{33}$$

The raw spike has $b > a$, but the coherent group has larger smoothed scores. Both scores are positive, so clipping preserves the reversal. Coincident within-group representations simplify the construction. If the distance is continuous in the representations, sufficiently small feature perturbations preserve the strict gap. This example establishes the possibility of a ranking reversal, not an accuracy improvement or a guarantee about individual final replay coefficients.

Lemma 1 (Smoothing before clipping)**.** *Averaging signed increments before clipping cannot exceed clipping first and then averaging. For every row-stochastic nonnegative matrix S and every $\delta \in \mathbb{R}^m$,*

$$[S\delta]_+ \leq S[\delta]_+. \tag{34}$$

Proof. The map $x \mapsto [x]_+$ is convex. For every row i, Jensen's inequality gives

$$\left[\sum_j S_{ij}\delta_j\right]_+ \leq \sum_j S_{ij}[\delta_j]_+. \tag{35}$$

Applying this coordinatewise proves the claim. □

D.2 PROOF OF PROPOSITION 1

Proof. All expectations are conditional on $\mathcal{G}_t$. Using $\delta_j = F_j + \xi_j$ and $\sum_j S_{ij} = 1$,

$$\tilde{\delta}_i - F_i = \sum_j S_{ij}(F_j - F_i) + \sum_j S_{ij}\xi_j. \tag{36}$$

The second term has conditional mean zero. Assumption 1 therefore gives

$$\begin{aligned} |\mathbb{E}_t[\tilde{\delta}_i] - F_i| &\leq \sum_j S_{ij}|F_j - F_i| \\ &\leq L_F \sum_j S_{ij} d(h_i, h_j) = L_F r_i, \end{aligned} \tag{37}$$

proving the first line of Eq. (6).

For every $u \in \mathbb{R}$, conditional independence gives

$$\begin{aligned} \mathbb{E}_t\left[\exp\left(u\sum_j S_{ij}\xi_j\right)\right] &= \prod_j \mathbb{E}_t[e^{uS_{ij}\xi_j}] \\ &\leq \exp\left(\frac{\sigma^2 u^2}{2}\sum_j S_{ij}^2\right). \end{aligned} \tag{38}$$

Thus, the smoothed residual has variance at most $\sigma^2 \sum_j S_{ij}^2$. The bias and centered noise terms in Eq. (36) have zero cross term, so

$$\mathbb{E}_t[(\tilde{\delta}_i - F_i)^2] \le L_F^2 r_i^2 + \sigma^2 \sum_j S_{ij}^2. \tag{39}$$

Because $0 < K_{ij} \le 1$, $K_{ii} = 1$, and $\sum_j K_{ij} = 1 + n_i$,

$$\sum_j S_{ij}^2 = \frac{\sum_j K_{ij}^2}{(1+n_i)^2} \le \frac{1}{1+n_i}. \tag{40}$$

Equation (39) proves the second line of Eq. (6); Eq. (40) gives the additional $1/(1+n_i)$ bound. Finally, the positive-part map is nonexpansive,

$$|[x]_+ - [y]_+| \le |x - y|, \tag{41}$$

so the third line of Eq. (6) follows by squaring and taking conditional expectation. □

Exact comparison with the raw signed estimator. Let $v_j = \mathbb{E}_t[\xi_j^2]$ and $b_i = \sum_j S_{ij}(F_j - F_i)$. Conditional independence and centering imply

$$\begin{aligned} \mathbb{E}_t[(\widetilde{\delta}_i - F_i)^2] &= b_i^2 + \sum_j S_{ij}^2 v_j, \\ \mathbb{E}_t[(\delta_i - F_i)^2] &= v_i. \end{aligned} \tag{42}$$

With common variance v, subtracting gives the condition $b_i^2 < v(1 - \|S_{i\cdot}\|_2^2)$. The nonexpansiveness of clipping bounds its estimation error but does not imply that this condition is necessary and sufficient for comparing two clipped estimators.

E Proofs for transport allocation

Equation (7) is a one-sided transport problem because it constrains every source-column mass but leaves the target marginal free. For finite costs, $\tau > 0$, and $\epsilon_T > 0$, its feasible set is compact and its entropy-regularized objective is continuous under $0 \log 0 = 0$. Strict concavity makes the maximizer unique.

E.1 Derivation of the transport gradient and pullback

Proof. Set $p_{i|j} = m\Pi_{ij}$ and let $\Delta_m = \{p \in \mathbb{R}_+^m : \sum_i p_i = 1\}$. For each source column j, Eq. (7) reduces, up to the factor $1/m$, to

$$\max_{p_{\cdot|j} \in \Delta_m} \left\{ \sum_i p_{i|j}(v_i - \tau C_{ij}) - \tau\epsilon_T \sum_i p_{i|j} \log(m p_{i|j}) \right\}. \tag{43}$$

The entropy-regularized objective is strictly concave on the simplex. With finite scores and positive entropy coefficient, its maximizing probabilities are positive: moving mass into a zero coordinate has a positive, unbounded entropy derivative. The interior Lagrangian first-order condition therefore yields

$$p_{i|j}^*(v) = \frac{\exp((v_i - \tau C_{ij})/(\tau\epsilon_T))}{\sum_k \exp((v_k - \tau C_{kj})/(\tau\epsilon_T))}. \tag{44}$$

Using $\Pi_{ij}^*(v) = p_{i|j}^*(v)/m$ gives Eq. (8). Substitution gives the transport value in log-sum-exp form

$$\mathcal{V}_{\mathrm{OT}}(v) = \frac{\tau\epsilon_T}{m} \sum_{j=1}^m \log \left[\frac{1}{m} \sum_i \exp((v_i - \tau C_{ij})/(\tau\epsilon_T)) \right]. \tag{45}$$

The transport value is smooth, convex, coordinatewise nondecreasing, and translation equivariant:

$$\mathcal{V}_{\mathrm{OT}}(v + c\mathbf{1}) = \mathcal{V}_{\mathrm{OT}}(v) + c. \tag{46}$$

Differentiating the log-sum-exp expression gives

$$\begin{aligned}\frac{\partial \mathcal{V}_{\mathrm{OT}}(v)}{\partial v_i} &= \frac{1}{m}\sum_j \frac{\exp((v_i - \tau C_{ij})/(\tau\epsilon_T))}{\sum_k \exp((v_k - \tau C_{kj})/(\tau\epsilon_T))} \\ &= \sum_j \Pi^*_{ij}(v) = q^*_i(v),\end{aligned} \tag{47}$$

which establishes the target-marginal sensitivity used in Eq. (9).

To differentiate through consensus and clipping, define locally $\chi_i = \mathbf{1}\{\tilde{\delta}_i > 0\}$, taking the zero subgradient when $\tilde{\delta}_i = 0$. For any δ', convexity of the positive-part map gives the following coordinatewise inequality:

$$[S\delta']_+ - [S\delta]_+ \geq \chi \odot S(\delta' - \delta). \tag{48}$$

Since $q^*(\tilde{s}) = \nabla \mathcal{V}_{\mathrm{OT}}(\tilde{s}) \geq 0$, convexity of $\mathcal{V}_{\mathrm{OT}}$ implies

$$\begin{aligned}\mathcal{V}_{\mathrm{OT}}([S\delta']_+) - \mathcal{V}_{\mathrm{OT}}([S\delta]_+) &\geq \langle q^*(\tilde{s}), [S\delta']_+ - [S\delta]_+ \rangle \\ &\geq \left\langle S^\top (q^*(\tilde{s}) \odot \chi), \delta' - \delta \right\rangle.\end{aligned} \tag{49}$$

This proves that Eq. (9) is a valid subgradient (and the gradient away from clipping boundaries). Nonnegativity follows from $S, q^*, \chi \geq 0$, and row-stochasticity gives

$$\begin{aligned}\|w^{\mathrm{SC}}\|_1 &= \mathbf{1}^\top S^\top (q^*(\tilde{s}) \odot \chi) \\ &= \sum_i q^*_i(\tilde{s})\chi_i \leq \sum_i q^*_i(\tilde{s}) = 1,\end{aligned} \tag{50}$$

This proves the nonnegativity and mass bound used in Eq. (10). □

E.2 A CONDITIONAL REGIONAL-MASS BOUND

Proposition 4 (Source-local regional mass bound)**.** *Let A contain k samples, with $1 \leq k < m$, and write $q^*(A) = \sum_{i\in A} q^*_i(\widetilde{s})$. Suppose there is $\Delta_A > 0$ such that*

$$\min_{i\in A}(\widetilde{s}_i - \tau C_{ij}) \geq \max_{\ell \notin A}(\widetilde{s}_\ell - \tau C_{\ell j}) + \Delta_A \tag{51}$$

for every source $j \in A$. Then

$$q^*(A) \geq \frac{k}{m}\left[1 + \frac{m-k}{k}\exp\left(-\frac{\Delta_A}{\tau\epsilon_T}\right)\right]^{-1}. \tag{52}$$

The premise concerns only sources inside A. For finite Δ_A, the displayed lower bound is smaller than k/m, so it does not establish that the total target mass exceeds uniform replay. Nor does it establish a lower bound on the final coefficients after the signed mask and $S^\top$ pullback. It only quantifies mass supplied by the protected source subset under the stated adjusted-score gap.

Proof. Fix $j \in A$ and write

$$M_j = \max_{\ell \notin A}(\tilde{s}_\ell - \tau C_{\ell j}). \tag{53}$$

By the adjusted-score gap assumed in Proposition 4, every $i \in A$ satisfies

$$\tilde{s}_i - \tau C_{ij} \geq M_j + \Delta_A. \tag{54}$$

Therefore,

$$\begin{aligned}\sum_{i\in A}\exp\left(\frac{\tilde{s}_i - \tau C_{ij}}{\tau\epsilon_T}\right) &\geq k\exp\left(\frac{M_j + \Delta_A}{\tau\epsilon_T}\right), \\ \sum_{\ell\notin A}\exp\left(\frac{\tilde{s}_\ell - \tau C_{\ell j}}{\tau\epsilon_T}\right) &\leq (m-k)\exp\left(\frac{M_j}{\tau\epsilon_T}\right).\end{aligned} \tag{55}$$

Hence the conditional mass sent from source j into A satisfies

$$\sum_{i\in A} m\Pi^*_{ij}(\tilde{s}) \geq \left[1 + \frac{m-k}{k}\exp\left(-\frac{\Delta_A}{\tau\epsilon_T}\right)\right]^{-1}. \tag{56}$$

Finally,

$$q^*(A) = \frac{1}{m}\sum_{j=1}^{m}\sum_{i\in A} m\Pi^*_{ij}(\tilde{s}). \tag{57}$$

Retaining only the k nonnegative source-column terms with $j \in A$ proves Eq. (52). □

F Proof and refinements of Theorem 1

Proof. All expectations are conditional on $\mathcal{G}_t$. Let $\zeta = S\xi$ and define, locally in this proof,

$$\begin{aligned} v_* &= \sigma^2 \max_i \sum_j S_{ij}^2, \\ r_* &= \max_i r_i. \end{aligned} \tag{58}$$

Equation (38) implies that every ζ_i is $\sqrt{v_*}$-sub-Gaussian. Moreover, Assumption 1 gives

$$\|SF - F\|_\infty = \max_i \left|\sum_j S_{ij}(F_j - F_i)\right| \le L_F r_*. \tag{59}$$

By monotonicity and translation equivariance of $\mathcal{V}_{\mathrm{OT}}$,

$$|\mathcal{V}_{\mathrm{OT}}(v) - \mathcal{V}_{\mathrm{OT}}(v')| \le \|v - v'\|_\infty. \tag{60}$$

The positive-part map is coordinatewise one-Lipschitz, so

$$\begin{aligned} \mathcal{V}_{\mathrm{OT}}([S\delta]_+) &= \mathcal{V}_{\mathrm{OT}}([SF + \zeta]_+) \\ &\le \mathcal{V}_{\mathrm{OT}}([F]_+) + L_F r_* + \|\zeta\|_\infty. \end{aligned} \tag{61}$$

For every $u > 0$,

$$\begin{aligned} \mathbb{E}_t[e^{u\|\zeta\|_\infty}] &\le \sum_i \mathbb{E}_t[e^{u|\zeta_i|}] \\ &\le 2m\exp(v_* u^2/2). \end{aligned} \tag{62}$$

Jensen's inequality therefore gives

$$\mathbb{E}_t[\|\zeta\|_\infty] \le \frac{\log(2m)}{u} + \frac{v_* u}{2}. \tag{63}$$

Optimizing at $u = \sqrt{2\log(2m)/v_*}$ when $v_* > 0$ yields

$$\mathbb{E}_t[\|\zeta\|_\infty] \le \sqrt{2v_*\log(2m)}. \tag{64}$$

The case $v_* = 0$ is immediate. Substituting Eqs. (58) and (64) into Eq. (61) proves Eq. (12). □

Corollary 1 (High-probability and temperature refinements)**.** *Under the assumptions of Theorem 1, the transport-value guarantee also admits a deviation form and an expectation bound depending explicitly on the entropy scale. Let $r_* = \max_i r_i$. For every $\zeta_0 \in (0,1)$, with conditional probability at least $1 - \zeta_0$,*

$$\begin{aligned} \mathcal{V}_{\mathrm{OT}}([S\delta]_+) \le \mathcal{V}_{\mathrm{OT}}([F]_+) + L_F r_* \\ + \sigma\sqrt{2\log\left(\frac{2m}{\zeta_0}\right)\max_i \sum_j S_{ij}^2}. \end{aligned} \tag{65}$$

Furthermore,

$$\begin{aligned} \mathbb{E}_t[\mathcal{V}_{\mathrm{OT}}([S\delta]_+)] \le \mathcal{V}_{\mathrm{OT}}([F]_+) + L_F r_* \\ + \min\left\{\sigma\sqrt{2\log(2m)\max_i \sum_j S_{ij}^2}, \right. \\ \left. \frac{\sigma^2 \max_i \sum_j S_{ij}^2}{2\tau\epsilon_T} + \tau\epsilon_T \log 2\right\}. \end{aligned} \tag{66}$$

Proof. Set $\zeta = S\xi$ and $v_* = \sigma^2 \max_i \sum_j S_{ij}^2$. The sub-Gaussian union bound gives

$$\mathbb{P}_t(\|\zeta\|_\infty \geq x) \leq 2m \exp\left(-\frac{x^2}{2v_*}\right). \tag{67}$$

When $v_* = 0$, the residuals vanish almost surely and the deviation statement is immediate. For $v_* > 0$, taking $x = \sqrt{2v_* \log(2m/\zeta_0)}$ in Eq. (61) proves Eq. (65).

For the temperature-dependent bound, coordinatewise,

$$[(SF)_i + \zeta_i]_+ \leq [F_i]_+ + L_F r_* + |\zeta_i|. \tag{68}$$

Monotonicity and translation equivariance yield

$$\mathcal{V}_{\mathrm{OT}}([S\delta]_+) \leq L_F r_* + \mathcal{V}_{\mathrm{OT}}([F]_+ + |\zeta|). \tag{69}$$

For a fixed source column j, let

$$a_{ij} = \exp\left(\frac{[F_i]_+ - \tau C_{ij}}{\tau \epsilon_T}\right). \tag{70}$$

Conditional Jensen's inequality and sub-Gaussianity give

$$\begin{aligned}\mathbb{E}_t\left[\tau\epsilon_T \log\left(\frac{1}{m}\sum_i a_{ij} e^{|\zeta_i|/(\tau\epsilon_T)}\right)\right] &\leq \tau\epsilon_T \log\left(\frac{1}{m}\sum_i a_{ij}\mathbb{E}_t[e^{|\zeta_i|/(\tau\epsilon_T)}]\right) \\ &\leq \tau\epsilon_T \log\left(\frac{1}{m}\sum_i a_{ij}\right) + \frac{v_*}{2\tau\epsilon_T} + \tau\epsilon_T \log 2.\end{aligned} \tag{71}$$

Averaging over source columns proves the second branch in Eq. (66). The first branch is Theorem 1; taking the smaller proves the corollary. □

G Local risk descent and its implementation limits

This appendix varies the candidate update direction while keeping the scoring geometry fixed; it does not optimize the model parameters directly. Evaluate w^{SC} at $\delta(g_c)$ and define

$$g_{\mathrm{SC}}^v = \sum_i w_i^{\mathrm{SC}} \nabla_\theta \ell(z_i; \theta_t - \eta g_c). \tag{72}$$

Theorem 2 (Prospective spatial-risk descent)**.** *Fix S, C, the stored supervision, and all scoring hyperparameters as g varies, and write $\Psi(g) = \mathcal{V}_{\mathrm{OT}}([S\delta(g)]_+)$. If the memory losses are differentiable and no coordinate of $S\delta(g_c)$ is zero, then*

$$\begin{aligned}\nabla_g \Psi(g_c) &= -\eta g_{\mathrm{SC}}^v, \\ D\Psi(g_c)[g_{\mathrm{SC}}^v] &= -\eta \|g_{\mathrm{SC}}^v\|_2^2.\end{aligned} \tag{73}$$

Proof. Because no coordinate of $S\delta(g_c)$ is zero, Ψ is differentiable at g_c. For every i,

$$\nabla_g \delta_i(g_c) = -\eta \nabla_\theta \ell(z_i; \theta_t - \eta g_c). \tag{74}$$

The chain rule and Eq. (9) give

$$\begin{aligned}\nabla_g \Psi(g_c) &= \sum_i w_i^{\mathrm{SC}} \nabla_g \delta_i(g_c) \\ &= -\eta \sum_i w_i^{\mathrm{SC}} \nabla_\theta \ell(z_i; \theta_t - \eta g_c) \\ &= -\eta g_{\mathrm{SC}}^v.\end{aligned} \tag{75}$$

Taking the inner product with g_{SC}^v proves Eq. (73). □

Corollary 2 (Finite-step and implemented-gradient extensions)**.** *Assume throughout the fixed-state, differentiability, and nonzero-consensus conditions of Theorem 2. The following bounds distinguish a correction in the virtual replay direction from the current-parameter replay direction used in the ideal update.*

Let $\Psi(g) = \mathcal{V}_{\mathrm{OT}}([S\delta(g)]_+)$. If Ψ has an L_Ψ-Lipschitz gradient on the segment from g_c to $g_c+\beta g^v_{\mathrm{SC}}$, with $L_\Psi > 0$ and $\beta > 0$, then

$$\Psi(g_c + \beta g^v_{\mathrm{SC}}) \le \Psi(g_c) - \left(\beta\eta - \frac{L_\Psi \beta^2}{2}\right) \|g^v_{\mathrm{SC}}\|_2^2. \tag{76}$$

Hence every $0 < \beta < 2\eta/L_\Psi$ gives strict descent when $g^v_{\mathrm{SC}} \neq 0$.

If each memory gradient is L_ℓ-Lipschitz in θ, define

$$\begin{aligned} g^0_{\mathrm{SC}} &= \sum_i w_i^{\mathrm{SC}} \nabla_\theta \ell(z_i; \theta_t), \\ g_u &= \frac{1}{m} \sum_i \nabla_\theta \ell(z_i; \theta_t), \\ \bar{g} &= (1-\alpha) g_u + \alpha g^0_{\mathrm{SC}}. \end{aligned} \tag{77}$$

Then

$$D\Psi(g_c)[\bar{g}] \le -\eta \|g^v_{\mathrm{SC}}\|_2 \left[\alpha\big(\|g^v_{\mathrm{SC}}\|_2 - \eta L_\ell \|g_c\|_2\big) - (1-\alpha)\|g_u\|_2\right]. \tag{78}$$

Thus, the current-parameter blended replay direction is locally descending for Ψ whenever $g^v_{\mathrm{SC}} \neq 0$ and the bracket is positive. This conclusion concerns an infinitesimal correction $g_c + \beta\bar{g}$. The actual replay strength defines a finite correction and requires its own step-size and smoothness conditions; it is not covered by the finite-step bound for g^v_{SC} without that additional argument.

Proof. The descent lemma and Theorem 2 give

$$\begin{aligned} \Psi(g_c + \beta g^v_{\mathrm{SC}}) &\le \Psi(g_c) + \beta \langle \nabla_g \Psi(g_c), g^v_{\mathrm{SC}} \rangle + \frac{L_\Psi \beta^2}{2} \|g^v_{\mathrm{SC}}\|_2^2 \\ &= \Psi(g_c) - \left(\beta\eta - \frac{L_\Psi \beta^2}{2}\right) \|g^v_{\mathrm{SC}}\|_2^2, \end{aligned} \tag{79}$$

proving Eq. (76).

For the current-point gradient, Lipschitz continuity and the mass bound $\|w^{\mathrm{SC}}\|_1 \le 1$ imply

$$\begin{aligned} \|g^0_{\mathrm{SC}} - g^v_{\mathrm{SC}}\|_2 &\le \sum_i w_i^{\mathrm{SC}} \|\nabla_\theta \ell(z_i; \theta_t) - \nabla_\theta \ell(z_i; \theta_t - \eta g_c)\|_2 \\ &\le \eta L_\ell \|g_c\|_2 \sum_i w_i^{\mathrm{SC}} \\ &\le \eta L_\ell \|g_c\|_2. \end{aligned} \tag{80}$$

Therefore,

$$\langle g^v_{\mathrm{SC}}, g^0_{\mathrm{SC}} \rangle \ge \|g^v_{\mathrm{SC}}\|_2^2 - \eta L_\ell \|g_c\|_2 \|g^v_{\mathrm{SC}}\|_2, \tag{81}$$

and Cauchy–Schwarz gives

$$\langle g^v_{\mathrm{SC}}, g_u \rangle \ge -\|g^v_{\mathrm{SC}}\|_2 \|g_u\|_2. \tag{82}$$

Since $\nabla_g \Psi(g_c) = -\eta g^v_{\mathrm{SC}}$ and $\bar{g} = (1-\alpha)g_u + \alpha g^0_{\mathrm{SC}}$, combining the last two inequalities proves Eq. (78). □

H Experimental details and additional results

This appendix collects the implementation and run details for the three experimental domains, the additional vision results, and the definitions of the component controls.

H.1 Replay implementation

Let m denote the number of scored memories, which can be smaller than memory capacity K. Scoring and aggregation follow Sections 4.1 and 4.2. Writing $u_i = 1/m$, the normalized sampling probabilities are

$$p_i = \frac{\bar{w}_i}{s_w}, \qquad \bar{w}_i = (1-\alpha)u_i + \alpha w_i^{\mathrm{SC}}, \qquad s_w = \sum_i \bar{w}_i. \tag{83}$$

The full configurations use $\alpha = 0.5$. Their replay estimators differ from Algorithm 1 as specified below.

Sampling and replay strength. Vision ER and ER-ACE draw 32 memories without replacement according to p and multiply their mean replay loss by s_w, with replay strength $\lambda = 1$. X-DER changes its label-replay and logit-replay draws but retains its own loss coefficients. Instruction tuning normalizes the weights to unit mass and averages selected losses without replacement. Code RL draws eight groups with replacement and also uses a unit-mass mean. Both use $\lambda = 1$, without the s_w multiplier. The without-replacement estimators do not use inclusion-probability corrections and are not generally unbiased for Eq. (11), even with the s_w multiplier (Appendix A).

H.2 Continual language-model post-training

Model, stream, and memory. We use the 15 instruction-formatted classification tasks in Section 5.1, with verbalized labels and exact-match evaluation, without synthetic label corruption. T5-Large uses rank-8 LoRA and a single pass of 860 updates with 16 current examples per update. Table 8 specifies the scoring and replay configuration. At each pool refresh, sampling without replacement selects $m = \min(512, |\mathcal{M}|)$ memories. Hence the candidate pool contains the entire available memory when $K = 300$; 512 is an upper bound, not a requirement to duplicate examples.

Table 8: Instruction-tuning configuration. Distances and allocation scales are computed on the scored candidate pool.

Quantity	Setting
Representation	Current T5-Large with its LoRA adapters; last decoder layer, mean-pooled over target-label tokens (1024 dimensions), then ℓ_2-normalized. Refreshed with the candidate pool.
Distance	$C_{ij} = \|h_i - h_j\|_2/\nu$, where $\nu = \mathrm{median}_{k<l}\|h_k - h_l\|_2$ within the pool.
Allocation	$\epsilon_s = 0.2$, $\epsilon_T = 0.5$; $\tau = c_\beta \,\mathrm{std}([S\delta]_+)/\epsilon_T$ with $c_\beta = 1$; $\alpha = 0.5$.
Optimization	AdamW learning rate 10^{-3} for training; one virtual SGD step on LoRA parameters with $\eta_v = 10^{-2}$.
Memory and scoring	$K \in \{300, 3000\}$; up to 512 distinct candidates, refreshed every four replay events.
Replay	Four examples per update, without replacement; unit-mass mean replay loss, $\lambda = 1$.

Parameter selection. The development search used $\alpha \in \{0.5, 0.7, 0.85\}$, $\epsilon_T \in \{0.2, 0.5\}$, and $c_\beta \in \{0.5, 1.0\}$ on held-out task order 7 with seeds 0 and 1. That order is excluded from the reported evaluations. The aggregation scale $\epsilon_s = 0.2$ and virtual step $\eta_v = 10^{-2}$ were fixed.

Task orders and paired runs. Table 9 gives the three evaluated orders. A run is an order–seed pair. For $K = 300$, the replay methods use 15 pairs: orders 4 and 5 with seeds 0–4, and order 6 with seeds 1–5. For $K = 3000$, they use nine pairs: orders 4 and 5 with seeds 0–2, and order 6 with seeds 1–3. Within a pair, the stream shuffle, reservoir decisions, and candidate-pool random draws are matched. The no-replay reference is summarized over 16 runs, separately from these paired replay comparisons.

Tables 1 and 3 report means and standard deviations across order–seed pairs; Min. is the minimum single-run final performance. The matched neighborhood comparison in Table 2 uses 11 pairs: order 4 with seeds 0–4, order 5 with seeds 0–2, and order 6 with seeds 1–3. That table reports means over these 11 pairs, rather than over all 15 scarce-memory runs.

Table 9: Task orders for instruction-tuning evaluation.

Order	Task sequence
4	mnli, cb, wic, copa, qqp, boolq, rte, imdb, yelp, amazon, sst2, dbpedia, agnews, multirc, yahoo
5	multirc, boolq, wic, mnli, cb, copa, qqp, rte, imdb, sst2, dbpedia, agnews, yelp, amazon, yahoo
6	yelp, amazon, mnli, cb, copa, qqp, rte, imdb, sst2, dbpedia, agnews, yahoo, multirc, boolq, wic

H.3 Noisy-label vision

H.3.1 Training protocol and host integrations

Split MNIST and Split CIFAR-10 each contain five two-class tasks with memory capacity 500. Split CIFAR-100 contains ten ten-class tasks with capacity 2,000; Split TinyImageNet contains ten twenty-class tasks with capacity 4,000. These are total capacities, not per-class allocations. Symmetric corruption replaces a selected training label uniformly with a different class label. Evaluation uses clean labels and, for Class-IL, no task identifier. Noise rates are 0, 20, 40, and 60%; MNIST also includes 80%.

The CIFAR experiments use one stream pass, ResNet-18 trained from scratch with SGD, current and replay batches of 32, random crops, and horizontal flips. The vision runs use an Intel Core i9-13900HX CPU (24 cores) and an NVIDIA A100 GPU (40 GB).

Host policies and optimization budgets. ER and ER-ACE use reservoir memories. ER-ACE retains its asymmetric stream cross-entropy and cosine classifier. MIR-prop samples in proportion to positive interference, using candidate pools of 96 on CIFAR-10 and 160 on CIFAR-100. X-DER retains task-end memory insertion, logit updates and transplanting, and its auxiliary contrastive draw; SPHERE replaces its label-replay and logit-distillation draws.

ER-type CIFAR methods use learning rate 0.1; DER++ and X-DER use 0.03. Their respective distillation/label-loss coefficient pairs are $(0.5, 0.5)$ and $(0.1, 0.5)$ on CIFAR-10, and $(0.3, 1.0)$ and $(0.1, 0.8)$ on CIFAR-100. GDumb fits its memory for 256 epochs. The CIFAR-10 PuriDivER configuration uses 255 epochs of MixMatch memory fitting with CutMix and learning rate 0.03. The evaluated ABS-based configuration combines ER-ACE, ABS sample selection, and the same fitting stage. These methods are grouped separately because additional fitting changes the optimization budget.

The PuriDivER integration applies SPHERE within purity estimation and sample partitioning. It is distinct from the ER-style replay-weight interface below and does not use the parameters in Table 10.

H.3.2 Representation geometry and parameter selection

The image representation is the current ResNet-18's globally averaged last residual-stage output (512 dimensions). MNIST instead uses the current MLP's last hidden layer (100 dimensions). At each rescoring, features are recomputed in evaluation mode on stored, unaugmented inputs and ℓ_2-normalized. Distances are $C_{ij} = \|h_i - h_j\|_2^2$. The entire available memory is scored at the stated interval and at the first step of each task.

Let $r_i(k)$ denote the distance from memory i to its k-th nearest neighbor. The bandwidths and transport temperature are

$$\begin{aligned} \epsilon_s &= \mathrm{median}_i\, r_i(k_s), \\ \epsilon_T &= \max\{\mathrm{median}_i\, r_i(k_T), \epsilon_s\}, \\ T &= T_{\mathrm{frac}}\, \mathrm{EMA}_{0.9}(P_{95}([\delta]_+)), \qquad \tau = T/\epsilon_T. \end{aligned} \tag{84}$$

The exponential moving average is updated at rescorings, and P_{95} is the 95th percentile. Thus $T = \tau\epsilon_T$ is the entropy scale in Eq. 7. Table 10 gives the host-specific settings. Each virtual update is one SGD step on all parameters at the host learning rate; parameters and batch-normalization statistics are restored afterward.

Table 10: Vision allocation settings. The last two columns give the rescoring interval in stream steps and the virtual-step size. $\alpha = 0.5$; ER and ER-ACE use $\lambda = 1$.

Dataset	Host	k_s	k_T	T_{frac}	Interval	η
CIFAR-10, MNIST	ER	5	25	0.5	4	0.1
CIFAR-10	ER-ACE	10	50	1.0	4	0.1
CIFAR-10	X-DER	5	50	0.5	8	0.03
CIFAR-100	ER, ER-ACE	5	25	0.5	4	0.1
CIFAR-100	X-DER	5	25	0.5	4	0.03
TinyImageNet	ER	5	25	0.5	4	0.1
TinyImageNet	ER-ACE	5	50	0.25	8	0.1
TinyImageNet	X-DER	5	25	0.5	4	0.03
MNIST	ER-ACE	5	25	0.5	4	0.1

Development search. The search used seeds 100–103 at 20, 40, and 60% noise, with $k_s \in \{5, 10\}$, $k_T \in \{25, 50\}$, $T_{\text{frac}} \in \{0.25, 0.5, 1.0\}$, rescoring interval in $\{2, 4, 8\}$, and $\alpha \in \{0, 0.5, 0.7, 0.85\}$. Selection maximized the mean improvement over the corresponding $\alpha = 0$ control on development runs. One configuration per dataset and host was fixed before evaluation. CIFAR-100 ER transfers the CIFAR-10 configuration; CIFAR-100 ER-ACE selects among twelve transferred configurations. Baseline searches used the same development seeds: learning rates $\{0.03, 0.1\}$ for ER, ER-ACE, DER++, and X-DER; MIR-prop candidate sizes $\{96, 160, 320\}$; and distillation/label-loss coefficients $\{0.1, 0.2, 0.3, 0.5\} \times \{0.5, 0.8, 1.0\}$ for DER++ and X-DER. These host loss coefficients are distinct from SPHERE's mixing coefficient α.

Repeated runs and pairing. CIFAR-10 and CIFAR-100 use seeds 0–9. TinyImageNet and MNIST use seeds 0–5, as do the PuriDivER and ABS-based evaluations. Host/+SPHERE comparisons match class orders, corruption realizations, stream orders, and seeds; reservoir-based hosts also match reservoir decisions. Matching seeds does not require identical model-dependent memory contents or stored logits for every host. The vision comparison tables use population standard deviations across evaluation seeds. The clean neighborhood-block ablation in Table 6 instead uses ten paired runs with seeds 10–19 and sample standard deviations. These evaluation seeds are separate from the development seeds.

H.3.3 Additional accuracy and retention results

Tables 11 and 12 supplement Table 4 with final Class-IL accuracy on MNIST and CIFAR-10. Improvements depend on the host and noise level: the CIFAR-10 integrations improve ER, ER-ACE, and X-DER at 40% and 60% noise, whereas X-DER's mean decreases at 20%.

Table 11: Split MNIST Class-IL accuracy (%, $K = 500$; mean±SD).

Method	Clean	20%	40%	60%	80%
Stream-time replay					
ER	87.3±0.8	65.3±4.0	46.9±5.4	31.3±5.8	19.2±2.9
ER + SPHERE	86.3±1.1	70.8±6.7	54.1±4.9	31.5±5.5	19.7±3.1
MIR	88.2±0.6	61.6±6.6	45.7±5.0	31.5±5.5	20.6±2.5
DER++	**90.7±0.6**	77.5±6.0	62.1±4.6	38.0±3.5	20.7±3.0
ER-ACE	88.8±1.1	75.7±9.4	66.2±2.8	43.7±2.9	**21.3±2.8**
ER-ACE + SPHERE	89.2±0.4	**81.6±3.8**	**69.8±3.2**	**45.8±4.5**	**21.3±2.1**

Table 13 combines the 40% and 60% retention results; Table 5 gives 20%. Class-IL and Task-IL use final accuracy without and with task identity. AAA averages incremental Class-IL accuracy. Average forgetting is computed per class; worst-class accuracy is the minimum final class accuracy, and $\text{CVaR}_{0.2}$ averages forgetting over the worst 20% of classes. ER-ACE + SPHERE improves all six reported means over ER-ACE at both additional noise levels, including the worst-class and tail metrics.

Table 12: Final Class-IL accuracy on Split CIFAR-10 (%, memory 500). Values are mean±SD. Best means are bold within each protocol.

Method	Clean	20%	40%	60%
Stream-time replay				
ER	43.4±5.7	33.4±4.5	25.1±2.9	17.4±2.7
ER + SPHERE	47.6±3.8	35.8±2.7	27.6±1.6	19.9±1.8
MIR-prop	44.6±5.2	32.1±4.4	24.7±2.1	18.1±1.2
DER++	43.7±5.1	28.2±4.3	23.3±4.8	18.2±3.3
ER-ACE	39.8±4.5	36.1±4.5	27.9±3.4	20.3±2.3
ER-ACE + SPHERE	41.7±5.8	**38.8±4.2**	**31.3±2.4**	**22.6±1.8**
X-DER	45.4±2.9	28.1±8.2	25.1±6.4	17.2±3.9
X-DER + SPHERE	**48.6**±4.4	27.5±8.9	26.6±6.9	18.4±3.7
Separate buffer training				
GDumb	46.6±1.3	36.8±0.9	27.2±0.9	19.8±1.1
PuriDivER	62.2±0.9	60.6±1.6	57.4±1.3	45.1±1.8
PuriDivER + SPHERE	**65.5±1.7**	**62.0±1.5**	**57.9±2.4**	**51.3±4.3**
ABS-based	48.5±2.9	47.6±2.5	42.4±2.3	29.1±2.1

Table 13: CIFAR-10 retention (%, $K = 500$; mean±SD). Bold compares ER-ACE with its SPHERE integration.

Method	Class-IL↑	Task-IL↑	AAA↑	avgF↓	Worst-class↑	$\text{CVaR}_{0.2}$ ↓
40% symmetric noise						
ER	25.1±2.8	71.6±3.4	45.3±3.6	49.7±5.1	2.6±1.5	81.1±3.5
ER-ACE	27.9±3.4	72.7±4.1	47.0±3.3	28.4±4.5	8.3±6.0	59.4±8.6
ER-ACE + SPHERE	**31.3±2.4**	**74.3±5.4**	**50.2±3.0**	**18.4±6.4**	**16.4±3.6**	**52.2±8.0**
DER++	23.3±4.8	76.8±4.7	39.7±4.9	60.6±9.5	2.0±4.6	86.9±7.5
MIR-prop	24.7±2.1	69.3±3.8	38.7±2.4	37.9±5.2	2.5±2.5	72.8±5.3
X-DER	25.1±6.4	71.7±8.8	38.4±7.8	24.3±8.0	0.0±0.0	59.1±14.8
GDumb (buffer fitting)	27.2±0.8	69.0±2.7	26.2±0.7	25.0±2.8	7.3±3.8	59.6±16.3
60% symmetric noise						
ER	17.4±2.7	62.4±3.0	36.4±3.6	54.8±4.5	0.8±1.2	84.5±6.9
ER-ACE	20.3±2.2	64.8±4.1	39.3±2.8	27.0±5.3	5.1±3.2	58.2±7.6
ER-ACE + SPHERE	**22.6±1.8**	**67.4±2.7**	**41.9±2.9**	**21.0±6.5**	**11.2±1.6**	**52.0±9.9**
DER++	18.2±3.3	68.8±3.4	34.2±3.3	62.2±7.3	0.3±0.4	90.1±4.9
MIR-prop	18.1±1.2	63.8±2.1	32.2±2.5	33.0±4.1	2.0±1.1	73.9±4.6
X-DER	17.2±3.9	63.8±5.8	33.7±5.0	25.5±4.9	0.0±0.0	64.8±8.7
GDumb (buffer fitting)	19.8±1.1	62.9±2.0	24.7±0.8	22.1±2.8	3.8±2.8	67.0±18.3

H.4 Code-generation reinforcement learning

H.4.1 Data and optimization

The four KodCode-V1 stages are Package, Data Structure/Algorithm, Leetcode, and Codeforces/Apps. Problems use only the standard library and have 4–12 collectible pytest nodes. Reference solutions must pass all nodes in two executions; unstable nodes are excluded. Near duplicates are removed using centered prompt embeddings with cosine similarity above 0.90. Each stage has 120 training, 100 evaluation, 40 development-fitting, and 40 development-probing problems, with evaluation disjoint from both training and development. The visible test is the first node after sorting by the SHA256 hash of problem and node identifiers. Full test suites are unavailable to training and replay selection and are used only for offline audits and evaluation.

Qwen2.5-Coder-1.5B-Instruct uses rank-16 LoRA on all projection layers, with LoRA scaling parameter 32 (not SPHERE's α). AdamW uses learning rate 10^{-4}. Eight current problems and two passes over each stage give 30 updates per stage and 120 overall. The protocol permits repeated access within a stage. Eight programs are generated per problem by vLLM using the current LoRA

version. Prompts and responses are capped at 1,536 and 512 tokens. Programs run in an isolated environment with CPU, memory, and wall-clock limits and pytest plugins disabled.

The group advantages are centered without division by reward standard deviation. The clipped token-level surrogate uses a fixed denominator of 512 and sampler correction $\min\{2, \exp(\log p_0 - \log q)\}$, where p_0 and q are the trainer and sampler probabilities. A virtual AdamW step on current problems, at learning rate 10^{-4}, scores candidate groups with the same sampled programs and advantages before and after the step. Parameters and optimizer state are then restored.

H.4.2 Geometry, candidates, and development selection

Prompt embeddings come from the frozen base model's last hidden layer, mean-pooled over prompt tokens (1,536 dimensions). They are ℓ_2-normalized, centered by subtracting the mean over all training problems, and normalized again. Let $d_0 = 0.524$ be the median cosine distance to the eighth nearest neighbor over the training pool. The geometry and allocation settings are $C_{ij} = \frac{1-h_i^\top h_j}{d_0}$, $\epsilon_s = \frac{1}{\ln 10}$, $\epsilon_T = 1$, $\tau = 0.25\, P_{95}([S\delta]_+)$ and $\alpha = 0.5$.

Embeddings and their normalization are computed before training and held fixed. This preprocessing uses training prompts from all four stages, but no evaluation problems or full-suite outcomes.

Reservoir admission follows a pregenerated schedule shared within each seed. Capacity is 480, equal to the number of distinct training problems; scoring and replay budgets, rather than this capacity, restrict access. Candidate selection favors problems whose latest replay group had both passing and failing programs. Each update generates eight programs for each of 48 candidates, and only mixed-outcome groups enter the scored set. SPHERE replays eight groups with replacement and unit total weight. All replay methods share this candidate-generation budget. Common random seeds do not force identical candidates or programs after model-dependent histories diverge. No-replay training generates no candidates.

Development selection compared learning rates $\{5 \times 10^{-5}, 10^{-4}\}$ and $\epsilon_s \in \{1/\ln 10, 1/\ln 4, 1\}$ on the fitting/probing splits with seeds 101, 202, and 303. The entropy parameter, blend coefficient, and τ rule were fixed before evaluation. Evaluation uses seeds 11, 33, and 55 in the fixed stage order, matching stream order, reservoir schedules, and random-seed schedules across methods. Table 7 reports the mean and sample standard deviation of final accuracy over these three runs; its other columns report means only.

H.4.3 Evaluation metrics

Greedy evaluation requires passing the full test suite. Let $A_{t,j}$ be full-suite accuracy in percent on stage j after training stage t, and $A_{0,j}$ its initial accuracy. For four stages,

$$\text{Final} = \frac{1}{4}\sum_{j=1}^{4} A_{4,j}, \qquad \text{Old} = \frac{1}{3}\sum_{j=1}^{3} A_{4,j}, \qquad \text{Worst} = \min_{1\le j\le 4} A_{4,j}. \tag{85}$$

$$\begin{aligned} F &= \frac{1}{3}\sum_{j=1}^{3}\left(\max_{j\le t\le 4} A_{t,j} - A_{4,j}\right), \\ \text{BWT} &= \frac{1}{3}\sum_{j=1}^{3}(A_{4,j} - A_{j,j}). \end{aligned} \tag{86}$$

Forgetting and BWT are measured in percentage points; lower forgetting and higher BWT are better. Checkpoint-average accuracy, when used, is $(1/16)\sum_{t=1}^{4}\sum_{j=1}^{4} A_{t,j}$. It includes future stages at earlier checkpoints and is not a seen-task-only incremental average. Initial macro-average accuracy is 45.5%.

H.5 Component ablations and interpretation

The controls test different alternatives to the full allocation rule. Names such as "No transport" do not denote the same operation across domains. Below, $v = [S\delta]_+$ is the supported field, $\chi =$

$\mathbf{1}_{\{S\delta>0\}}$, and $u_i = 1/m$. For a nonnegative vector of positive mass, write $\mathcal{N}(a) = a/\sum_i a_i$, and define $\mathrm{softmax}_T(a)_i = \exp(a_i/T)/\sum_j \exp(a_j/T)$ for $T > 0$. The full method uses $w^{\mathrm{SC}} = S^\top(q^*(v) \odot \chi)$ and Eq. (83).

Table 14 gives the specified replacement in each control. All retain the full method's memory capacity, candidate-pool budget, replay-batch budget, and virtual step. The two controls called "Consensus only" and "No transport" differ in their allocation rule: the former uses an ESS-matched softmax, whereas the latter directly normalizes the supported scores and omits the pullback. Their results are therefore not two evaluations of an identical transport-free method.

Table 14: Instruction-tuning controls in Tables 2 and 3. The two transport-free controls use different allocation rules.

Control	Operation
$S = I$	Use the identity for aggregation and omit its pullback; retain clipping, transport with C, and the uniform floor.
Consensus only	Replace $q^*(v)$ by $\mathrm{softmax}_T(v)$, with T chosen to match the ESS of the full method's weights. Set$w^{\mathrm{SC}} = S^\top(\mathrm{softmax}_T(v) \odot \chi)$, and retain the uniform mixture and normalization in Eq. (83).
No transport	Normalize the supported field directly, $\mathcal{N}(v)$, without transport or the $S^\top$ pullback.
Permuted geometry	At every pool refresh, randomly permute feature rows before constructing C and S, breaking their alignment with the memories while retaining the full allocation procedure.
Matched MIR	Use $\mathrm{softmax}_T([\delta]_+)$ with T chosen to match the ESS of the full method's weights.
No uniform floor	Set $\alpha = 1$; remove only the uniform mixture from the full allocation rule.

Scope of component attribution. The $S = I$ control evaluates signed aggregation and its chain-rule pullback jointly, while retaining the transport distance cost. It measures the contribution of the neighborhood construction as a whole, rather than assigning separate gains to aggregation and $S^\top$. Consensus only instead compares geometric transport with an ESS-matched softmax of the same supported scores, retaining the downstream weighting operations. These comparisons evaluate specific alternatives to the full method; their differences are not additive shares of the overall improvement. Instruction tuning and code-generation RL both use unit-mass replay without the s_w multiplier, so their gains do not depend on that explicit scaling factor. The no-uniform-floor control tests baseline uniform rehearsal, not the isolated effect of s_w.