HIERARCHICAL RESPONSE PRESERVATION FOR CONTINUAL ADAPTATION OF ZERO-SHOT GRAPH–TEXT MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Pretrained graph–text models align graph representations with textual semantics, enabling recognition of unseen classes and transfer across graph domains. However, as graph data and classes continually arrive, models should learn from new supervision while retaining their zero-shot transfer capabilities and historical task knowledge. Two challenges arise: (i) new classes can overturn historical predictions despite preserved distinctions among historical classes, and (ii) overly strict response preservation can stall learning of new classes. To address these challenges, we propose **Hi**erarchical **R**esponse **P**reservation (HiRP). HiRP represents this competition through a hierarchical response that keeps each historical-class probability and sums new-class probabilities, preserving historical distinctions and aggregate competition while allowing distinctions within the new class group to adapt. It further uses the geometry induced by this response to guide constrained updates, retaining useful adaptation directions while controlling response drift. Across three class-incremental settings, HiRP achieves absolute gains of 1.84–7.95 percentage points in average accuracy over the strongest compared baseline in each setting, while mitigating zero-shot transfer degradation. **Code** is available at `https://anonymous.4open.science/r/HiRP-807C/`.

1 INTRODUCTION

Pretrained graph–text models enable zero-shot node classification by matching node representations to textual class descriptions in a shared semantic space (Zhu et al., 2025; Liu et al., 2026). In evolving environments, however, labeled classes may arrive sequentially from different graph domains, requiring these models to acquire new knowledge beyond their pretrained capabilities. We study *cross-graph class-incremental adaptation*: each task introduces a disjoint set of classes, and inference must distinguish all classes encountered so far without access to task identities. The goal is to learn from incoming supervision while retaining both historical-class performance and zero-shot transfer to graph domains excluded from adaptation.

The central challenge is to reconcile new knowledge acquisition with knowledge preservation in a semantic prediction space that continually expands. Supervised updates can disrupt both the graph–text alignment established during pretraining and the distinctions learned from earlier tasks. The vulnerability of pretrained transfer is substantial: in our ten-task setting, joint fine-tuning reduces GraphCLIP's zero-shot accuracy on the held-out History domain from 49.69% to 1.46% (Figure 1). Meanwhile, newly introduced class descriptions change the competition faced by historical classes. Preserving previously learned distinctions is therefore not enough: those distinctions must continue to support correct predictions against an expanding set of competitors.

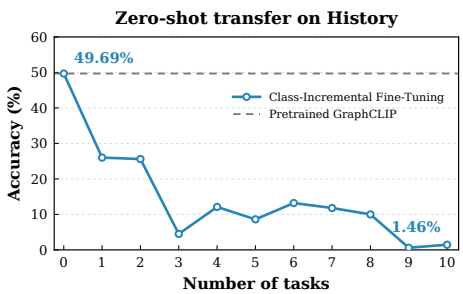


Figure 1: Zero-shot accuracy of GraphCLIP (Zhu et al., 2025) on History.

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.

Existing continual learning methods mitigate forgetting through parameter regularization (Kirkpatrick et al., 2017), functional preservation (Li & Hoiem, 2016; Titsias et al., 2020), and gradient projection (Farajtabar et al., 2020; Wang et al., 2026a). Related work addresses historical–new class separation (Hou et al., 2019), probability aggregation under changing label semantics (Cermelli et al., 2020), and functional trust regions (Schulman et al., 2015). However, simply restricting model changes or preserving historical outputs does not ensure prediction stability when new competitors enter the candidate space. At the other extreme, preserving the complete expanded response can restrict distinctions that should be learned among new classes. Moreover, a bound on functional drift does not by itself determine how to continue learning near its boundary. These limitations call for a selective preservation target for expanding class competition and an update rule that exploits the adaptation freedom left by that target, giving rise to two interconnected challenges.

Challenge 1: Preserving historical predictions under expanding class competition. A historical node's correct class may still outrank every other historical class yet be overtaken by a newly introduced class (Figure 2). Thus, stable relative preferences among historical classes need not yield stable predictions in the expanded space. The challenge is to identify a response that protects both historical distinctions and their competition with new classes, without unnecessarily constraining how the model distinguishes classes within the new group.

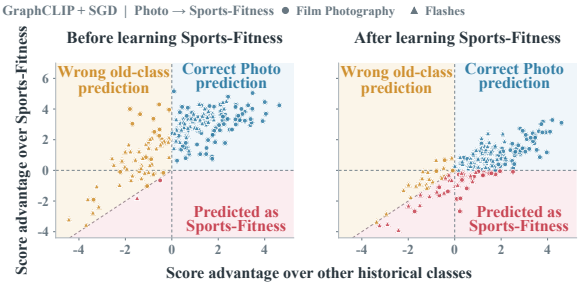


Figure 2: Evolving class competition: new classes can overturn historical predictions despite preserved discrimination among historical classes.

Challenge 2: Learning new classes without stalling model updates. Once a response is protected, adaptation must operate within its allowed drift. This restriction is inherently directional: some update components strongly alter the protected probabilities, whereas others can improve new-task performance with less disruption. Near the drift limit, uniformly shrinking an update suppresses both kinds of components and can stall learning. The challenge is to retain useful adaptation directions while keeping changes to the protected response within the prescribed limits.

To address these challenges, we propose *Hierarchical Response Preservation* (HiRP), a continual adaptation framework for pretrained graph–text models. The key idea is to use *the same hierarchical response to define both what is preserved and how adaptation proceeds*. HiRP constructs this response on unlabeled support features and derives its preservation constraints and local update geometry from that response, connecting the relationships to be protected with the directions available for learning.

To address Challenge 1, HiRP preserves historical class-level responses and aggregate new-class competition. For each support node, it retains an individual probability for every historical class and aggregates the probabilities of all new classes into a single component. Constraining this hierarchical response protects historical distinctions and limits changes in the total probability assigned to new competitors. Crucially, it leaves probability redistribution within the new-class group unconstrained by the preservation target, allowing new supervision to refine those distinctions. We further establish a sufficient condition connecting response preservation to prediction stability.

To address Challenge 2, HiRP guides adaptation using the geometry of the protected response. The same response Kullback–Leibler (KL) divergence defines per-step and task-level drift constraints, as well as a local update metric that favors directions with less impact on protected relationships. Near the task-level drift limit, tangent correction suppresses the divergence-increasing component of an update while retaining motion along the constraint boundary. Nonlinear KL feasibility checks verify both drift limits before an update is accepted. This design seeks useful new-task progress within the preservation constraints rather than relying solely on uniform step shrinkage.

Our key contributions are summarized as follows. ❶ ***Framework.*** We propose HiRP, a continual learning framework for pretrained graph–text models that jointly addresses new knowledge acquisition, zero-shot transfer retention, and historical knowledge preservation as classes arrive across graph domains. ❷ ***Mechanism.*** We design hierarchical response preservation and response-

geometry-guided adaptation, representing protected competitive relationships through historical class-level and new group-level responses. The same response KL defines drift constraints and an update metric, while tangent correction and nonlinear feasibility checks enable constrained updates. We further establish a sufficient condition for prediction stability. ❸ ***Experiments.*** We evaluate HiRP across three class-incremental settings against 20 baselines. HiRP improves average accuracy by **1.84–7.95 percentage points** over the strongest compared baseline in each setting, while mitigating zero-shot transfer degradation during continual adaptation.

2 Related Work

Continual learning and graph adaptation. Replay methods retain exemplars or historical inputs and logits (Rebuffi et al., 2017; Buzzega et al., 2020); EWC and MAS penalize changes to important parameters (Kirkpatrick et al., 2017; Aljundi et al., 2018), while OGD locally protects outputs through gradient projection (Farajtabar et al., 2020). L2P learns task-agnostic prompts (Wang et al., 2022); graph TPP predicts tasks through profiling and prompting (Niu et al., 2024). LUCIR addresses historical/current-class imbalance and separation (Hou et al., 2019), and G2LoRA protects category-level subspaces while coordinating graph/text adapter updates (Wang et al., 2026a). HiRP constrains a shared readout's response on unlabeled features; inference directly scores all arrived classes.

Distillation and functional memory. LwF distills historical responses on current inputs (Li & Hoiem, 2016); FRCL and FROMP preserve functions at inducing inputs or memorable examples (Titsias et al., 2020; Pan et al., 2020). MiB aggregates current-class and background probabilities to match a previous segmentation model's background label (Cermelli et al., 2020). DKD separates target-versus-rest mass and non-target conditional probabilities (Zhao et al., 2022). HiRP uses pretrained class semantics to evaluate both references on the same expanded candidates: its aggregate selects current-class distinctions to leave unconstrained, rather than reconciling changing background labels as in MiB. The contribution couples this response choice with local and task-level drift control and feasible adaptation on unlabeled features. Aggregation and the KL chain rule are established tools; Appendix A.1 compares protected quantities and support.

Trust-region updates. TRPO limits successive policy KL through a local quadratic approximation (Schulman et al., 2015); Natural Continual Learning shapes updates by prior precision (Kao et al., 2021). TRGP selects related historical tasks for gradient projection and scaled weight reuse (Lin et al., 2022), while a recent preprint couples generative replay with a Fisher trust region (Wang et al., 2026b). HiRP uses the selected response for its metric and nonlinear KL checks against the current iterate and task-start reference, redirecting outward proposals near the task budget.

3 Problem Setup

Learning protocol. We consider a sequence of T class-incremental tasks. Task t introduces new classes $\mathcal{C}_t$, disjoint from earlier class sets, with labeled training data $\mathcal{D}_t$. Each input x denotes a node together with its graph context. A task may introduce classes from a new graph domain or a new class subset of an existing domain. The graph instances are fixed and transductive: graph structure and unlabeled attributes, including neighbors from other tasks or splits, may be used to compute features. Only current-task training labels enter the supervised loss; validation/test nodes never enter the feature support. We call $\mathcal{O}_t = \bigcup_{i<t} \mathcal{C}_i$ the *historical classes*, $\mathcal{N}_t = \mathcal{C}_t$ the *current classes*, and $\mathcal{C}_{\le t} = \mathcal{O}_t \cup \mathcal{N}_t$ the cumulative candidates. Task boundaries are known, and each task uses its complete prescribed training split. Inference considers all classes in $\mathcal{C}_{\le t}$ without a ground-truth task identity.

Competition in a shared semantic space. HiRP freezes the pretrained GraphCLIP graph and text encoders. Let $\mathbf{h}(x) \in \mathbb{R}^d$ be the frozen node feature, $\widetilde{\mathbf{h}}(x) = [\mathbf{h}(x); 1]$ its augmentation, and $\mathbf{u}_c \in \mathbb{R}^m$ the unit-normalized text embedding of class c. Only the shared readout $\mathbf{W} \in \mathbb{R}^{m\times(d+1)}$

is updated. With a fixed scale $\gamma > 0$, the class score is

$$s_c(x; \mathbf{W}) = \gamma \mathbf{u}_c^\top \frac{\mathbf{W}\widetilde{\mathbf{h}}(x)}{\|\mathbf{W}\widetilde{\mathbf{h}}(x)\|_2}. \tag{1}$$

The predictive distribution and decision are

$$p_t(c \mid x; \mathbf{W}) = \frac{\exp s_c(x; \mathbf{W})}{\sum_{j \in \mathcal{C}_{\le t}} \exp s_j(x; \mathbf{W})}, \qquad \hat{y}_t(x) = \arg\max_{c \in \mathcal{C}_{\le t}} s_c(x; \mathbf{W}). \tag{2}$$

Candidate arrival expands the label set, while the embedding dimension stays fixed. The frozen text encoder supplies $\mathbf{u}_c$ as soon as a class is introduced, so its competition on historical features is observable before adaptation. Training uses cross-entropy on $\mathcal{D}_t$ over these same candidates.

Bounded feature support. During task t, a support $\mathcal{M}_t$ contains at most M frozen feature vectors drawn from an external unlabeled reference pool and previous training tasks. Source and task metadata are retained, but historical node labels are not used in training losses. Support features serve only to evaluate functional constraints and their local metric. After each task, features from $\mathcal{D}_t$ are incorporated within the fixed support budget. The support construction and update constraints are specified in the method section.

4 Motivating Observations

Historical discrimination and cross-group competition. For each task transition, we evaluate the same historical test samples before and after training. Write $A_{-,O}$ and $A_{+,O}$ for their accuracies among historical candidates $\mathcal{O}_t$, and $A_{-,S}$ and $A_{+,S}$ for accuracies among cumulative candidates $\mathcal{C}_{\le t}$. The accuracy change admits the identity

$$A_{-,O} - A_{+,S} = \underbrace{A_{-,O} - A_{+,O}}_{\text{historical discrimination change}} + \underbrace{A_{+,O} - A_{+,S}}_{\text{post-update competition gap}}. \tag{3}$$

We average over task transitions within each run before aggregating runs. For SGD, the competition gap ranges from 17.58 to 22.45 percentage points across the three streams; historical discrimination changes by -1.81 to 1.66 points. Stable accuracy among historical candidates can therefore coexist with substantial errors under unified prediction.

Candidate expansion also causes errors before training. In the ten-task setting, adding current candidates at fixed pre-update parameters reduces accuracy from $A_{-,O} = 63.84\%$ to $A_{-,S} = 60.82\%$ (Figure 3). This 3.02-point loss is distinct from the 20.61-point post-update competition gap. The pre-update loss measures candidate insertion; the post-update gap includes both insertion and subsequent adaptation.

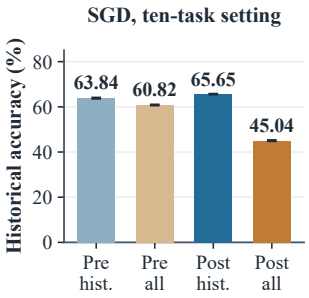


Figure 3: Historical accuracy with expanded candidates.

Uniform step shrinking can stall new-task learning. We uniformly scale down each proposed update without changing its direction (radial shrinking) on a 16-node training minibatch from CiteSeer, Task 2 of the four-task setting. Figure 4 tracks the accepted update norm and training loss as the trajectory approaches the task-level KL boundary and the local step-radius budget decreases. The update norm collapses toward zero, while the loss plateaus at 1.267. Satisfying the protection budget can therefore suppress adaptation even when the current gradient is nonzero. Appendix A.2 gives the diagnostic setup; Section 6.6 examines how tangent correction affects this behavior.

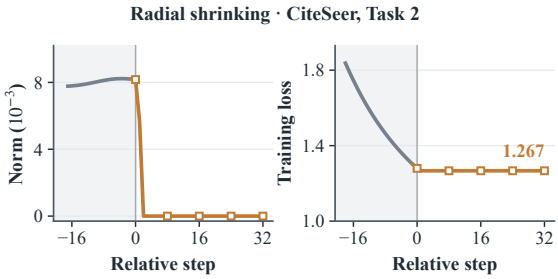


Figure 4: Uniform step shrinking near the KL limit reduces update norms while loss stalls.

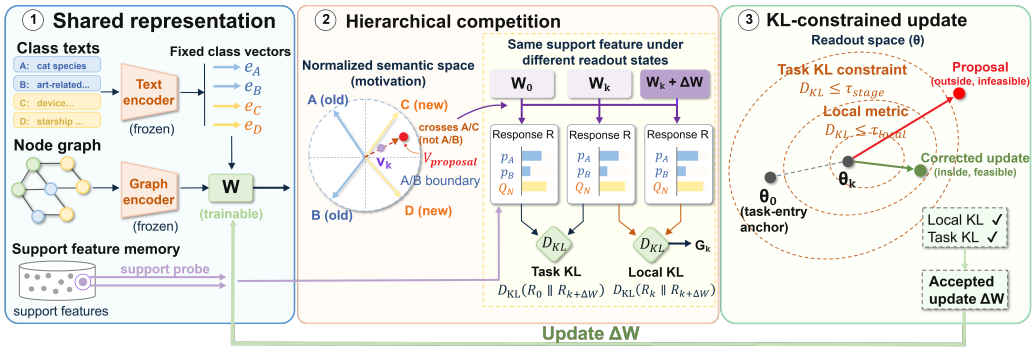


Figure 5: **Overview of HiRP. (1) Shared representation:** Frozen encoders provide node and class features; only the readout is updated. **(2) Hierarchical competition:** Historical classes remain separate, while new classes share an aggregate response. **(3) Constrained adaptation:** The response defines drift budgets and guides feasible readout updates.

These observations motivate a response that protects historical discrimination and aggregate cross-group competition while retaining useful adaptation directions. Protection acts on drift from the expanded-candidate reference, which can already contain candidate-insertion errors.

5 HiRP: Hierarchical Response Preservation

In this section, we introduce HiRP, a continual adaptation method for pretrained graph–text models that learns new classes while preserving historical performance and zero-shot transfer. As illustrated in Figure 5, HiRP comprises three connected steps: (1) a shared representation that freezes the graph and text encoders and adapts only the readout $\mathbf{W}$; (2) a hierarchical response that keeps each historical-class probability and sums current-class probabilities to capture competition between historical and new classes; and (3) a KL-constrained update that limits changes in this response and guides the readout toward useful learning directions. We write $\boldsymbol{\theta} = \mathrm{vec}(\mathbf{W})$, with $\boldsymbol{\theta}_0$ denoting the task-entry parameters and $\boldsymbol{\theta}_k$ the current iterate. The full procedure is given in Algorithm 1.

5.1 Selecting the Response to Preserve

Encoding competitive relationships. Predictive probabilities encode class competition: for $\boldsymbol{\nu} = \mathbf{W}\tilde{\mathbf{h}}/\|\mathbf{W}\tilde{\mathbf{h}}\|$, scores satisfy

$$\log \frac{p_i}{p_j} = s_i - s_j = \gamma(\mathbf{u}_i - \mathbf{u}_j)^\top \boldsymbol{\nu}. \tag{4}$$

Probability ratios therefore encode pairwise score margins exactly, without requiring probability calibration (Appendix A.2).

Suppressing the task index, let $p_c = p_t(c \mid x; \boldsymbol{\theta})$, $q_O = \sum_{o \in \mathcal{O}_t} p_o$, and $q_N = \sum_{n \in \mathcal{N}_t} p_n$. HiRP keeps each historical-class probability and represents current arrivals by their total probability:

$$\mathbf{r}_t(x; \boldsymbol{\theta}) = \big((p_o)_{o \in \mathcal{O}_t}, q_N\big), \qquad D_H(x; \boldsymbol{\theta}^a, \boldsymbol{\theta}) = D_{\mathrm{KL}}(\mathbf{r}_t(x; \boldsymbol{\theta}^a) \,\|\, \mathbf{r}_t(x; \boldsymbol{\theta}))\,. \tag{5}$$

For $t > 1$, define the historical conditional distribution $r_O(o) = p_o/q_O$. A superscript a denotes quantities evaluated at reference parameters $\boldsymbol{\theta}^a$. The KL chain rule makes the protected relationships explicit:

$$D_H = \underbrace{D_{\mathrm{KL}}((q_O^a, q_N^a) \,\|\, (q_O, q_N))}_{\text{aggregate group competition}} + \underbrace{q_O^a D_{\mathrm{KL}}(\mathbf{r}_O^a \| \mathbf{r}_O)}_{\text{historical conditional structure}}\,. \tag{6}$$

Which probabilities are constrained, and which can change. For historical A/B and current C/D, $\mathbf{r} = (p_A, p_B, p_C + p_D)$. Constraining this response limits changes in historical probabilities and the total probability assigned to current classes, while C/D can redistribute that total. This is the

coarse competition structure HiRP retains from Figure 2: current classes compete as a group, without fixing each new-class probability. For example, $(0.40, 0.10, 0.25, 0.25) \to (0.40, 0.10, 0.49, 0.01)$ keeps $\mathbf{r}$ unchanged but switches the winner from A to C. Prediction stability follows when the historical winner has sufficient margin over this aggregate.

Theorem 1 (Historical prediction stability) *For $t > 1$ and a reference historical winner c, let*

$$\delta_R = p_c^a - \max\left\{ \max_{o \in \mathcal{O}_t \setminus \{c\}} p_o^a, q_N^a \right\} > 0, \qquad D_H < \delta_R^2/2 \quad \Longrightarrow \quad \hat{y}_t(x; \boldsymbol{\theta}) = c. \tag{7}$$

The empty inner maximum is zero.

Proof. See Appendix A.3. □

This pointwise condition is not implied for every input by an average support budget. Appendix B.5 measures single-graph coverage.

5.2 Bounding response drift at two scales

We measure the chosen response's change across K nonempty support sources $\mathcal{S}_1, \ldots, \mathcal{S}_K$:

$$\widehat{D}_t(\boldsymbol{\theta}^a, \boldsymbol{\theta}) = \frac{1}{K} \sum_{r=1}^{K} \frac{1}{|\mathcal{S}_r|} \sum_{x \in \mathcal{S}_r} D_H(x; \boldsymbol{\theta}^a, \boldsymbol{\theta}). \tag{8}$$

Sources receive equal weight. Small updates can accumulate drift, so we constrain change from both the current iterate and fixed task entry:

$$\widehat{D}_t(\boldsymbol{\theta}_k, \boldsymbol{\theta}_{k+1}) \le \frac{\delta^2}{2}, \qquad \widehat{D}_t(\boldsymbol{\theta}_0, \boldsymbol{\theta}_{k+1}) \le \frac{\rho^2}{2}. \tag{9}$$

The first budget limits one step; the second limits drift from $\boldsymbol{\theta}_0$, which resets at the next task. Both references use the expanded candidates, including new-class competition before adaptation. These are support-average constraints; conditional protection is weaker where q_O^a is small.

After each task, sampled training features enter the $M = 1024$-row support (Appendix A.5).

5.3 Adapting within the response budgets

Equal-length parameter steps can change the protected response by different amounts. We therefore use the same response KL to define the local Fisher metric, with damping $\lambda > 0$:

$$\mathbf{F}_k = \nabla_{\boldsymbol{\theta}}^2 \widehat{D}_t(\boldsymbol{\theta}_k, \boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_k}, \qquad \mathbf{G}_k = \mathbf{F}_k + \lambda \mathbf{I}, \qquad \|\mathbf{v}\|_{\mathbf{G}_k}^2 = \mathbf{v}^\top \mathbf{G}_k \mathbf{v}. \tag{10}$$

This metric penalizes directions that rapidly change the response. For the current cross-entropy gradient $\mathbf{g}_k$, implicit Fisher products and conjugate gradients approximate $\mathbf{G}_k \mathbf{v}_k = \mathbf{g}_k$ (Appendix A.4), yielding

$$\mathbf{d}_k = -\delta \frac{\mathbf{v}_k}{\sqrt{\mathbf{g}_k^\top \mathbf{v}_k}}. \tag{11}$$

With an exact solve, this minimizes the linearized loss within the local metric ball.

Near the task-level KL boundary, an outward proposal may be almost eliminated by scalar shrinking even when tangent motion remains possible. Let $\mathbf{a}_k = \nabla_{\boldsymbol{\theta}} \widehat{D}_t(\boldsymbol{\theta}_0, \boldsymbol{\theta}_k)$ and compute $\mathbf{u}_k \approx \mathbf{G}_k^{-1} \mathbf{a}_k$. When the task divergence reaches 95% of its budget and $\mathbf{a}_k^\top \mathbf{d}_k > 0$, we form

$$\mathbf{d}_k^{\tan} = \mathbf{d}_k - \frac{\mathbf{a}_k^\top \mathbf{d}_k}{\mathbf{a}_k^\top \mathbf{u}_k} \mathbf{u}_k. \tag{12}$$

With an exact inverse, this $\mathbf{G}_k$-metric projection removes the first-order outward component and retains tangent motion. A heuristic inward adjustment accommodates curvature; the correction is rescaled and kept only if nondegenerate and descending (Appendix A.6). Finite CG supplies an approximate proposal, while the original nonlinear KL determines feasibility: scalar searches shrink it as needed, and both budgets are checked before acceptance.

We use $\delta = 0.002$, $\rho = 0.1$, and $\lambda = 0.01$. At task one, $\mathbf{r}_1 = (1)$ and $\mathbf{G}_k = \lambda \mathbf{I}$; the update is a normalized gradient step, so complete-system gains also include first-task optimization (Appendix A.7). Inference uses Equation (2) without the support or solver.

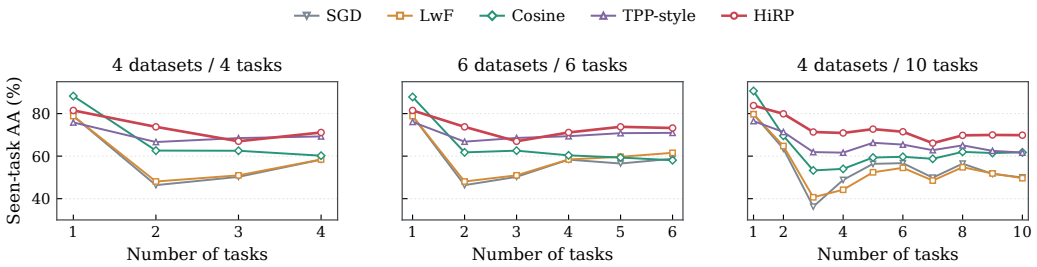


Figure 6: Accuracy throughout learning, averaged over arrived tasks. Each point uses all classes introduced up to that task.

6 Experiments

6.1 Experimental Setup

Datasets and baselines. We evaluate three class-incremental settings (Table 1). The first uses four datasets—Cora, CiteSeer, WikiCS, and Photo—in four tasks; the second adds Sports-Fitness and Actor for six datasets and six tasks. In both, each dataset forms one task. The third uses the same four datasets as the first, split into two, two, three, and three tasks, respectively, for ten tasks. Fixed graphs provide transductive context; training labels arrive task by task. We compare readout regularizers, prototype methods, LoRA-based adapters (Hu et al., 2022), and graph–text systems. SimpleCIL uses BERT as its text backbone. HiRP, SGD, and LwF share frozen GraphCLIP features and the pretrained readout; all use three epochs per task, batch size 16, and label smoothing 0.2. HiRP uses at most 1024 support features and the default budgets in Section 5.3. Datasets and baseline settings are detailed in Appendices B–B.3.

Evaluation protocol. Training uses the complete prescribed split; prediction considers all arrived classes without task identities. Let $A_{t,i}$ be accuracy on the test nodes of task i after learning task t, and n_i their number. We report final accuracy (FA), final task-averaged accuracy (AA), and signed average forgetting (AF):

$$\mathrm{FA} = \frac{\sum_{i=1}^{T} n_i A_{T,i}}{\sum_{i=1}^{T} n_i}, \qquad \mathrm{AA} = \frac{1}{T}\sum_{i=1}^{T} A_{T,i}, \qquad \mathrm{AF} = \frac{1}{T-1}\sum_{i=1}^{T-1}(A_{T,i} - A_{i,i}). \tag{13}$$

FA weights test nodes equally; AA weights tasks equally. Signed AF is the change since acquisition in percentage points; higher is better. We report means and sample standard deviations from five runs for the main table and three for aggregate auxiliary experiments. Individual-node and boundary diagnostics, common-start full-sequence controls, History trajectories, and parameter sweeps use one run, with fixed splits and class order. Bold marks the best mean; underlining marks the second best in Table 1.

6.2 Main Results on Class-Incremental Learning

Table 1 shows that HiRP leads in FA and AA across all settings. Its gains over the strongest baseline are 3.45, 3.04, and 13.43 points in FA, and 1.84, 2.51, and 7.95 points in AA, respectively; the ten-task setting shows the largest gains after repeated class expansion.

HiRP also has better AF than SGD and LwF. SimpleCIL forgets less but has substantially lower FA and AA, so lower forgetting alone does not imply better continual performance. Appendix B.6 separates acquisition and retention; Section 6.6 compares common-start updates, and Appendix B.8 compares distillation under the same memory budget.

6.3 Classification Throughout the Stream

HiRP stays above SGD and LwF throughout all three streams (Figure 6). At task 2 of the four- and six-task settings, AA is 73.77% versus LwF's 48.07%; the ten-task advantage persists through

Table 1: Class-incremental classification over three dataset/task schedules. Means are shown with sample standard deviations in subscript. FA and AA are accuracy (%); signed AF is forgetting (percentage points), with higher values better. Bold and underline mark the best and second-best means, respectively. GraphCLIP denotes sequential joint fine-tuning; backbones and † adaptations are specified in Appendix B.2.

Method	4 datasets / 4 tasks			6 datasets / 6 tasks			4 datasets / 10 tasks		
	FA ↑	AA ↑	AF ↑	FA ↑	AA ↑	AF ↑	FA ↑	AA ↑	AF ↑
GCN	$50.30_{\pm 0.06}$	$21.71_{\pm 0.03}$	$-81.26_{\pm 0.45}$	$1.00_{\pm 0.82}$	$11.99_{\pm 0.24}$	$-84.37_{\pm 0.30}$	$14.08_{\pm 0.02}$	$9.69_{\pm 0.01}$	$-90.58_{\pm 0.32}$
SGD	$75.79_{\pm 0.12}$	$58.36_{\pm 0.20}$	$-25.73_{\pm 0.17}$	$83.13_{\pm 0.19}$	$58.89_{\pm 0.27}$	$-22.64_{\pm 0.20}$	$50.24_{\pm 0.16}$	$50.05_{\pm 0.17}$	$-41.57_{\pm 0.20}$
LwF	$\underline{76.60}_{\pm 0.14}$	$58.46_{\pm 0.56}$	$-25.21_{\pm 0.73}$	$\underline{84.23}_{\pm 0.14}$	$61.63_{\pm 0.22}$	$-19.03_{\pm 0.22}$	$56.70_{\pm 0.63}$	$49.87_{\pm 0.39}$	$-40.99_{\pm 0.43}$
EWC	$75.84_{\pm 0.00}$	$58.13_{\pm 0.00}$	$-25.95_{\pm 0.00}$	$83.29_{\pm 0.00}$	$59.08_{\pm 0.00}$	$-22.44_{\pm 0.00}$	$50.30_{\pm 0.00}$	$50.12_{\pm 0.00}$	$-41.36_{\pm 0.00}$
MAS	$66.09_{\pm 0.84}$	$61.11_{\pm 0.56}$	$-12.26_{\pm 1.77}$	$66.85_{\pm 0.00}$	$62.79_{\pm 0.00}$	$-7.79_{\pm 0.00}$	$57.74_{\pm 1.10}$	$54.27_{\pm 1.12}$	$-31.22_{\pm 2.40}$
OGD	$76.44_{\pm 0.00}$	$58.38_{\pm 0.00}$	$-25.64_{\pm 0.00}$	$83.48_{\pm 0.00}$	$59.33_{\pm 0.00}$	$-22.18_{\pm 0.00}$	$47.50_{\pm 0.00}$	$48.14_{\pm 0.00}$	$-43.31_{\pm 0.00}$
Cosine	$62.95_{\pm 0.83}$	$60.02_{\pm 0.70}$	$-12.20_{\pm 1.16}$	$58.21_{\pm 3.35}$	$58.09_{\pm 0.38}$	$-7.84_{\pm 1.12}$	$\underline{61.87}_{\pm 1.49}$	$\underline{61.65}_{\pm 1.08}$	$-12.48_{\pm 0.78}$
TEEN	$44.30_{\pm 2.94}$	$40.53_{\pm 2.01}$	$-21.40_{\pm 0.66}$	$45.51_{\pm 6.53}$	$41.54_{\pm 2.88}$	$-13.25_{\pm 0.11}$	$50.29_{\pm 4.42}$	$47.45_{\pm 2.20}$	$-18.46_{\pm 1.66}$
TPP-style	$72.18_{\pm 0.11}$	$\underline{69.35}_{\pm 0.23}$	$\underline{-2.29}_{\pm 0.03}$	$76.04_{\pm 0.04}$	$\underline{70.94}_{\pm 0.06}$	$\underline{-1.40}_{\pm 0.02}$	$54.86_{\pm 0.50}$	$61.47_{\pm 0.72}$	$\underline{-9.83}_{\pm 0.38}$
BERT	$44.27_{\pm 0.12}$	$19.11_{\pm 0.05}$	$-75.19_{\pm 0.51}$	$34.02_{\pm 8.56}$	$19.03_{\pm 1.52}$	$-70.49_{\pm 2.15}$	$12.97_{\pm 0.08}$	$8.93_{\pm 0.05}$	$-83.86_{\pm 0.20}$
RoBERTa	$46.14_{\pm 0.34}$	$19.96_{\pm 0.23}$	$-77.38_{\pm 0.52}$	$75.12_{\pm 1.17}$	$27.64_{\pm 0.51}$	$-62.55_{\pm 0.62}$	$13.21_{\pm 0.04}$	$9.09_{\pm 0.03}$	$-83.57_{\pm 1.40}$
SimpleCIL	$42.47_{\pm 0.93}$	$52.15_{\pm 0.84}$	$\mathbf{-1.79}_{\pm 0.12}$	$37.66_{\pm 1.51}$	$47.21_{\pm 1.60}$	$\mathbf{-1.20}_{\pm 0.18}$	$42.09_{\pm 1.48}$	$49.61_{\pm 1.36}$	$\mathbf{-8.19}_{\pm 0.58}$
$\text{GCN}_{\text{LLMEmb}}$	$50.39_{\pm 0.40}$	$21.75_{\pm 0.17}$	$-81.82_{\pm 0.24}$	$0.66_{\pm 0.21}$	$12.21_{\pm 0.20}$	$-84.62_{\pm 0.36}$	$14.08_{\pm 0.02}$	$9.69_{\pm 0.01}$	$-91.17_{\pm 0.17}$
GraphCLIP	$44.27_{\pm 1.77}$	$19.12_{\pm 0.78}$	$-74.64_{\pm 0.95}$	$19.95_{\pm 9.81}$	$4.68_{\pm 2.15}$	$-70.40_{\pm 3.20}$	$15.37_{\pm 1.99}$	$9.88_{\pm 1.06}$	$-78.03_{\pm 0.70}$
GraphGPT†	$72.87_{\pm 0.94}$	$52.05_{\pm 0.86}$	$-37.19_{\pm 1.24}$	$83.51_{\pm 0.83}$	$47.86_{\pm 3.59}$	$-40.25_{\pm 4.42}$	$26.72_{\pm 2.19}$	$24.84_{\pm 1.74}$	$-71.19_{\pm 1.69}$
LLaGA†	$49.42_{\pm 0.14}$	$21.34_{\pm 0.08}$	$-77.58_{\pm 1.02}$	$21.35_{\pm 13.00}$	$15.34_{\pm 2.62}$	$-76.88_{\pm 3.19}$	$13.99_{\pm 0.05}$	$9.63_{\pm 0.04}$	$-87.49_{\pm 0.87}$
LoRA	$76.02_{\pm 0.15}$	$60.87_{\pm 0.13}$	$-20.26_{\pm 0.22}$	$83.63_{\pm 0.07}$	$64.07_{\pm 0.16}$	$-14.61_{\pm 0.14}$	$51.18_{\pm 0.12}$	$51.11_{\pm 0.15}$	$-38.60_{\pm 0.19}$
InfLoRA†	$54.54_{\pm 0.43}$	$30.35_{\pm 0.94}$	$-41.32_{\pm 2.40}$	$44.40_{\pm 3.99}$	$26.12_{\pm 1.07}$	$-27.31_{\pm 1.04}$	$47.48_{\pm 1.41}$	$37.35_{\pm 1.95}$	$-26.86_{\pm 6.04}$
SD-LoRA	$62.15_{\pm 1.37}$	$41.41_{\pm 1.60}$	$-27.13_{\pm 1.67}$	$65.82_{\pm 2.04}$	$38.03_{\pm 1.10}$	$-18.12_{\pm 1.00}$	$47.41_{\pm 0.53}$	$37.51_{\pm 1.24}$	$-30.09_{\pm 5.52}$
G2LoRA†	$72.70_{\pm 0.47}$	$57.90_{\pm 0.43}$	$-20.53_{\pm 0.43}$	$81.07_{\pm 0.05}$	$59.60_{\pm 0.13}$	$-16.07_{\pm 0.19}$	$49.05_{\pm 0.07}$	$51.46_{\pm 0.13}$	$-32.43_{\pm 0.16}$
HiRP (ours)	$\mathbf{80.05}_{\pm 0.15}$	$\mathbf{71.19}_{\pm 0.22}$	$-8.51_{\pm 0.36}$	$\mathbf{87.27}_{\pm 0.44}$	$\mathbf{73.45}_{\pm 0.40}$	$-6.11_{\pm 0.44}$	$\mathbf{75.30}_{\pm 0.31}$	$\mathbf{69.60}_{\pm 0.49}$	$-14.88_{\pm 0.30}$

repeated arrivals. These curves track acquisition and retention jointly. Appendices B.4, B.3, and B.9 report single-graph results, training costs, and parameter sensitivity.

6.4 Zero-Shot Transfer Retention

We revisit the History transfer degradation in Figure 1, comparing HiRP and GraphCLIP (Zhu et al., 2025) fine-tuning over ten tasks from the same pretrained model. History is excluded from training and feature support; evaluation uses its complete class set without target-domain adaptation (Appendix B.2). HiRP finishes at 52.64%, above the initial 49.69%, while GraphCLIP falls to 1.46% (Figure 7). HiRP thus retains zero-shot transfer while learning new classes; in a common-start readout comparison, it also exceeds SGD's mean across three held-out domains (Table 4).

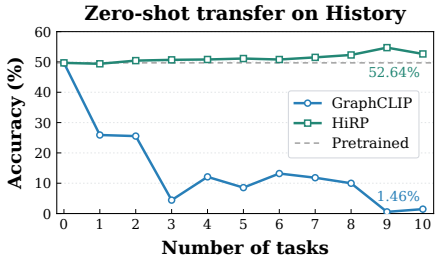


Figure 7: Zero-shot accuracy of GraphCLIP and HiRP on History during class-incremental fine-tuning.

6.5 Ablation Studies

We ablate the response and update direction. **Mass** preserves aggregate historical/current probabilities, while **Hier.** also preserves within-history distinctions. **Radial** shrinks the update; **tangent** corrects its direction. Table 2 compares Mass + radial, Mass + tangent, and Hier. + radial with full HiRP (Hier. + tangent).

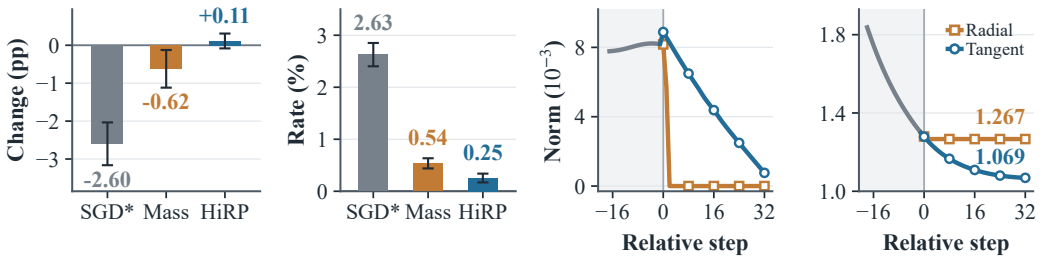
Replacing the hierarchical response with Mass while retaining tangent correction reduces FA, AA, and AF across all three settings (Table 2). Removing tangent correction lowers FA and AA but improves AF, trading adaptation for retention. Tangent correction alone does not consistently improve Mass, so response structure and update direction must be designed jointly.

Table 2: Response and update-direction ablation under the same data, support rules, and numerical KL budgets. The mass response keeps only historical/current group probabilities. HiRP combines the hierarchical response with tangent correction.

Method	4 datasets / 4 tasks			6 datasets / 6 tasks			4 datasets / 10 tasks		
	FA ↑	AA ↑	AF ↑	FA ↑	AA ↑	AF ↑	FA ↑	AA ↑	AF ↑
Mass + radial	$77.10_{\pm 0.40}$	$67.23_{\pm 0.45}$	$-13.91_{\pm 0.83}$	$84.12_{\pm 1.09}$	$69.60_{\pm 0.41}$	$-10.38_{\pm 0.26}$	$64.49_{\pm 1.37}$	$63.07_{\pm 1.20}$	$-25.38_{\pm 1.13}$
Mass + tangent	$78.50_{\pm 0.11}$	$68.07_{\pm 1.20}$	$-14.07_{\pm 2.12}$	$85.82_{\pm 0.54}$	$68.18_{\pm 0.63}$	$-13.24_{\pm 0.63}$	$62.31_{\pm 1.58}$	$58.06_{\pm 1.84}$	$-31.27_{\pm 2.04}$
Hier. + radial	$75.52_{\pm 0.88}$	$69.03_{\pm 0.14}$	$\mathbf{-7.79}_{\pm 0.41}$	$82.83_{\pm 0.47}$	$71.59_{\pm 0.14}$	$\mathbf{-5.17}_{\pm 0.17}$	$74.36_{\pm 0.11}$	$69.71_{\pm 0.15}$	$\mathbf{-12.76}_{\pm 0.27}$
HiRP (ours)	$\mathbf{80.09}_{\pm 0.19}$	$\mathbf{71.12}_{\pm 0.26}$	$-8.43_{\pm 0.32}$	$\mathbf{87.04}_{\pm 0.42}$	$\mathbf{73.22}_{\pm 0.23}$	$-6.21_{\pm 0.32}$	$\mathbf{75.43}_{\pm 0.05}$	$\mathbf{69.86}_{\pm 0.13}$	$-14.74_{\pm 0.15}$

6.6 Case Study

Preserving historical decisions. From a shared task-5 checkpoint in the ten-task setting, HiRP, Mass, and SGD take 32 identical batches; SGD* matches HiRP's step norms. For SGD*, Mass, and HiRP, old-only accuracy changes by -2.60, -0.62, and $+0.11$ points, while old-to-new crossing rates are 2.63%, 0.54%, and 0.25% (Figure 8a,b). New-class accuracy differs by less than 0.16 points: HiRP reduces crossings while preserving historical distinctions.



(a) Historical accuracy **(b) Old-to-new crossings** **(c) Update norm** **(d) Training loss**

Figure 8: Preserving historical decisions and maintaining adaptation. (a,b) Ten-task setting, Task 5: changes after 32 common-start updates; SGD* matches HiRP's step norms. Error bars show sample standard deviations. (c,d) CiteSeer, Task 2: accepted update norms and training loss under the same decreasing step-radius budget. Gray denotes the shared initial trajectory.

Learning near the protection boundary. We revisit the fixed-minibatch trajectory in Figure 4 from the same state and budget schedule. As the step-radius budget decreases, tangent correction lowers training loss to 1.069 at relative step 32, while radial shrinking stalls at 1.267 (Figure 8c,d). Tangent updates become smaller yet continue learning near the protection boundary. In separate complete task-2 runs, new-class accuracy rises from 64.47% to 66.67%, with a 0.31-point old-class decrease. Appendices A.2 and A.7 give diagnostic settings and single-step results.

7 Conclusion

We studied how pretrained graph–text models can learn classes arriving across graph domains while retaining historical performance and zero-shot transfer. Our findings show that preserving distinctions among historical classes alone is insufficient as new candidates enter the prediction space: competition between historical and new classes also matters. HiRP captures these relationships in a hierarchical response and uses it to constrain and guide readout updates. Across three class-incremental streams, HiRP improves unified classification and mitigates zero-shot degradation on held-out domains. These results suggest that continual adaptation benefits from identifying which predictive relationships to preserve and updating the model within the resulting constraints.

AI USE STATEMENT

In this work, we used generative AI tools for writing assistance and research execution. For writing assistance, these tools supported editing and polishing manuscript text and LaTeX files, as well as translating text. For research execution, they assisted with interpreting experimental results, revising scientific figures and plotting code, and debugging experimental code. The authors manually reviewed all AI-assisted content for correctness and consistency with the research and take full responsibility for the final content of this paper.

REPRODUCIBILITY STATEMENT

The learning and inference protocol is defined in Section 3, and the experimental setup is given in Section 6. Appendix A provides derivations and implementation details for HiRP. Appendix B documents the data and task protocols, baseline implementations, comparison conditions, and additional results, including hyperparameter sensitivity. An anonymized code link is provided in the abstract.

REFERENCES

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Computer Vision – ECCV 2018*, volume 11207 of *Lecture Notes in Computer Science*, pp. 144–161. Springer International Publishing, 2018. doi: 10.1007/978-3-030-01219-9_9. URL https://doi.org/10.1007/978-3-030-01219-9_9.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: A strong, simple baseline. In *Advances in Neural Information Processing Systems*, volume 33, pp. 15920–15930, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/b704ea2c39778f07c617f6b7ce480e9e-Abstract.html.

Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulò, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9230–9239, 2020. doi: 10.1109/CVPR42600.2020.00925. URL https://doi.org/10.1109/CVPR42600.2020.00925.

Runjin Chen, Tong Zhao, Ajay Kumar Jaiswal, Neil Shah, and Zhangyang Wang. LLaGA: Large language and graph assistant. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 7809–7823. PMLR, 2024. URL https://proceedings.mlr.press/v235/chen24bh.html.

Ziyang Cheng, Zhixun Li, Yuhan Li, Yixin Song, Kangyi Zhao, Dawei Cheng, Jia Li, Hong Cheng, and Jeffrey Xu Yu. Can LLMs alleviate catastrophic forgetting in graph continual learning? A systematic study. *arXiv preprint arXiv:2505.18697v2*, 2025. URL https://arxiv.org/abs/2505.18697v2.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 3762–3773. PMLR, 2020. URL https://proceedings.mlr.press/v108/farajtabar20a.html.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 831–839, 2019. doi: 10.1109/CVPR.2019.00092. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Hou_Learning_a_Unified_Classifier_Incrementally_via_Rebalancing_CVPR_2019_paper.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://iclr.cc/virtual/2022/poster/6319.

Ta-Chu Kao, Kristopher T. Jensen, Gido M. van de Ven, Alberto Bernacchia, and Guillaume Hennequin. Natural continual learning: Success is a journey, not (just) a destination. In *Advances in Neural Information Processing Systems*, volume 34, pp. 28067–28079, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/ec5aa0b7846082a2415f0902f0da88f2-Abstract.html.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/pdf?id=SJU4ayYgl.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114. URL https://doi.org/10.1073/pnas.1611835114.

Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision*, volume 9908 of *Lecture Notes in Computer Science*, pp. 614–629. Springer International Publishing, 2016. doi: 10.1007/978-3-319-46493-0_37. URL https://doi.org/10.1007/978-3-319-46493-0_37.

Yan-Shuo Liang and Wu-Jun Li. InfLoRA: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23638–23647, 2024. doi: 10.1109/CVPR52733.2024.02231. URL https://openaccess.thecvf.com/content/CVPR2024/html/Liang_InfLoRA_Interference-Free_Low-Rank_Adaptation_for_Continual_Learning_CVPR_2024_paper.html.

Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. TRGP: Trust region gradient projection for continual learning. In *International Conference on Learning Representations*, 2022. URL https://iclr.cc/virtual/2022/poster/6867.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. URL https://arxiv.org/abs/1907.11692.

Yuhang Liu, Minglai Shao, Zengyi Wo, Yunlong Chu, Bing Hao, Shengzhong Liu, Ruijie Wang, and Jianxin Li. Learning noise-resilient and transferable graph-text alignment via dynamic quality assessment. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1209–1220. ACM, 2026. doi: 10.1145/3805712.3809579. URL https://doi.org/10.1145/3805712.3809579.

Chaoxi Niu, Guansong Pang, Ling Chen, and Bing Liu. Replay-and-forget-free graph class-incremental learning: A task profiling and prompting approach. In *Advances in Neural Information Processing Systems*, volume 37, pp. 87978–88002, 2024. doi: 10.52202/079017-2793. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/a07e87ecfa8a651d62257571669b0150-Abstract-Conference.html.

Pingbo Pan, Siddharth Swaroop, Alexander Immer, Runa Eschenhagen, Richard E. Turner, and Mohammad Emtiyaz Khan. Continual deep learning by functional regularisation of memorable past. In *Advances in Neural Information Processing Systems*, volume 33, pp. 4453–4464, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/2f3bbb9730639e9ea48f309d9a79ff01-Abstract.html.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5533–5542, 2017. doi: 10.1109/CVPR.2017.587. URL https://doi.org/10.1109/CVPR.2017.587.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the International Conference on Machine Learning*, pp. 1889–1897, 2015. URL https://proceedings.mlr.press/v37/schulman15.html.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. GraphGPT: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*,

pp. 491–500. ACM, 2024. doi: 10.1145/3626772.3657775. URL https://doi.org/10.1145/3626772.3657775.

Michalis K. Titsias, Jonathan Schwarz, Alexander G. de G. Matthews, Razvan Pascanu, and Yee Whye Teh. Functional regularisation for continual learning with Gaussian processes. In *International Conference on Learning Representations*, 2020. URL https://iclr.cc/virtual/2020/poster/1650.

Qi-Wei Wang, Da-Wei Zhou, Yi-Kai Zhang, De-Chuan Zhan, and Han-Jia Ye. Few-shot class-incremental learning via training-free prototype calibration. In *Advances in Neural Information Processing Systems*, volume 36, pp. 15060–15076, 2023. doi: 10.52202/075280-0661. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/30dfe47a3ccbee68cffa0c19ccb1bc00-Abstract-Conference.html.

Yuhan Wang, Yibo Ding, Yutong Ye, Mufan Zhao, Wenbo Zhang, Ruijie Wang, and Jianxin Li. G^2LoRA: Gradient orthogonal low-rank adaptation framework for graph continual learning on text-attributed graphs. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 5184–5195. ACM, 2026a. doi: 10.1145/3770855.3817966. URL https://doi.org/10.1145/3770855.3817966.

Zekun Wang, Anant Gupta, and Christopher J. MacLellan. Trust region continual learning as an implicit meta-learner. *arXiv preprint arXiv:2602.02417*, 2026b. URL https://arxiv.org/abs/2602.02417.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 139–149, 2022. doi: 10.1109/CVPR52688.2022.00024. URL https://openaccess.thecvf.com/content/CVPR2022/html/Wang_Learning_To_Prompt_for_Continual_Learning_CVPR_2022_paper.html.

Yichen Wu, Hongming Piao, Long-Kai Huang, Renzhen Wang, Wanhua Li, Hanspeter Pfister, Deyu Meng, Kede Ma, and Ying Wei. SD-LoRA: Scalable decoupled low-rank adaptation for class incremental learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/92f43b1d33fae4aa1958f75317f0cec1-Abstract-Conference.html.

Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11943–11952, 2022. doi: 10.1109/CVPR52688.2022.01165. URL https://doi.org/10.1109/CVPR52688.2022.01165.

Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *International Journal of Computer Vision*, 133(3):1012–1032, 2025. doi: 10.1007/s11263-024-02218-0. URL https://doi.org/10.1007/s11263-024-02218-0.

Yun Zhu, Haizhou Shi, Xiaotang Wang, Yongchao Liu, Yaoke Wang, Boci Peng, Chuntao Hong, and Siliang Tang. GraphCLIP: Enhancing transferability in graph foundation models for text-attributed graphs. In *Proceedings of the ACM on Web Conference 2025*, pp. 2183–2197. ACM, 2025. doi: 10.1145/3696410.3714801. URL https://doi.org/10.1145/3696410.3714801.

A HiRP: Derivations and Implementation Details

A.1 Response design in context

Table 3 distinguishes the protected quantities and their input support. HiRP organizes semantic competition by class arrival without sample-level historical labels: the frozen classifier scores current class descriptions at task entry. Individual historical probabilities and collective current mass represent this structure, checked against current-iterate and task-start references.

Table 3: Response protection in related formulations. The comparison concerns the protected object and its inputs, not a claim that probability aggregation or functional regularization is unique to HiRP.

Method	Protected quantity	Inputs and enforcement
LwF	Historical-task outputs	Current-task inputs; distillation loss.
FRCL / FROMP	Function posteriors / past predictions	Inducing inputs / memorable examples; functional regularization.
MiB	Historical-class outputs and aggregated background/current-class probability	Segmentation inputs; background-aware distillation.
DKD	Target-versus-rest mass and non-target conditional distribution	Labeled distillation inputs; separately weighted losses.
HiRP	Historical discrimination and aggregate group competition, represented by individual historical probabilities and current-group mass	Bounded unlabeled features; local and task-level KL budgets on the same expanded candidates, with checked updates.

A.2 Semantic competition and historical prediction errors

Fix a task $t > 1$, a nonzero readout $\mathbf{W}\widetilde{\mathbf{h}}(x)$, and finite scores on the common candidate set $S = \mathcal{O}_t \cup \mathcal{N}_t$. Write $O = \mathcal{O}_t$, $N = \mathcal{N}_t$, and $\boldsymbol{\nu} = \mathbf{W}\widetilde{\mathbf{h}}/\|\mathbf{W}\widetilde{\mathbf{h}}\|$. The scale $\gamma > 0$ and unit class vectors $\mathbf{u}_c$ are fixed.

Theorem 2 (Semantic margins and predictive responses) *For every pair $i, j \in S$,*

$$\log \frac{p_i}{p_j} = \gamma(\mathbf{u}_i - \mathbf{u}_j)^\top \boldsymbol{\nu}. \tag{14}$$

If $\mathbf{u}_i \neq \mathbf{u}_j$, dividing this quantity by $\gamma\|\mathbf{u}_i - \mathbf{u}_j\|$ gives the signed distance from $\boldsymbol{\nu}$ to the pairwise decision hyperplane in the normalized feature coordinates. Furthermore, the historical-versus-current margin is

$$m = \max_{o \in O} s_o - \max_{n \in N} s_n = \log \frac{\max_{o \in O} p_o}{\max_{n \in N} p_n}. \tag{15}$$

Proof. The common softmax denominator cancels in p_i/p_j. Taking logarithms and substituting the score definition yields Equation (14). The decision hyperplane is $(\mathbf{u}_i - \mathbf{u}_j)^\top \boldsymbol{\nu} = 0$; normalization by its normal's length gives the distance. Since exponentiation preserves order, the same cancellation applies to the maxima in Equation (15). ∎

How competition contributes to historical errors. Consider any fixed distribution of labeled historical samples (x, y) with $y \in O$ and assume no score ties. Let $\hat{y}_O$ and $\hat{y}_S$ be the predictions on O and S, and let A_O, A_S be their accuracies for the same parameters. Then

$$A_O - A_S = \Pr\left\{ \hat{y}_O = y,\ \max_{n \in N} s_n > s_y \right\}. \tag{16}$$

Indeed, a sample correct under S is also correct under O. The remaining samples correct under O become incorrect precisely when a current class outranks y. Thus the competition gap in Equation (3) is the frequency of this event, and adding and subtracting the post-update A_O gives that decomposition exactly. For a sample correct at the reference, preserved historical ranking together with $m < 0$ after updating is a correct-to-incorrect change caused by the current group winning. Unconditional group crossings also include previously incorrect samples. On fixed candidates, the net accuracy drop equals the correct-to-incorrect rate minus the incorrect-to-correct rate; neither rate alone is the full-sequence signed AF. Candidate expansion can already introduce errors at task entry, which a subsequent drift constraint need not repair.

Which competition is retained by the hierarchy. The class-arrival partition is $\Pi_t = \{\{o\} : o \in O\} \cup \{N\}$. It organizes the existing semantic candidates into the competitive relationships retained by the response: each historical class has a separate coordinate, while current arrivals share one.

At the next task, current classes join the historical set and receive separate coordinates; subsequent arrivals form the aggregate group. The hierarchy concerns these response coordinates; the semantic embedding dimension remains fixed. The full probability vector determines all pairwise score margins and recovers scores up to a common additive offset. The hierarchical response intentionally retains less information. With $r_N(n) = p_n/q_N$ and $\mathrm{LSE}(\mathbf{s}_G) = \log \sum_{c\in G} e^{s_c}$,

$$\log \frac{q_O}{q_N} = \mathrm{LSE}(\mathbf{s}_O) - \mathrm{LSE}(\mathbf{s}_N), \tag{17}$$

$$m = \log \frac{q_O}{q_N} + \log \max_o r_O(o) - \log \max_n r_N(n). \tag{18}$$

Hence $\mathbf{r} = ((p_o)_{o\in O}, q_N)$ specifies historical conditional discrimination and aggregate group competition, while leaving the current-group conditional distribution free. These are identities for the model's scores; they neither require calibrated posteriors nor recover every aspect of feature geometry.

Historical-node margins in Figure 2. The figure evaluates the same 200 Photo test nodes (100 Film Photography and 100 Flashes) before and after learning Sports-Fitness at task 5 of the six-task setting, using the SGD readout. Both checkpoints score the same 35 historical classes and 13 Sports-Fitness classes. For true class y, the horizontal coordinate is $s_y - \max_{o\in\mathcal{O}_t\setminus\{y\}} s_o$ and the vertical coordinate is $s_y - \max_{n\in\mathcal{N}_t} s_n$. These are score margins computed directly from the logits, rather than a projection of node embeddings. The zero lines mark ties with the true class; the diagonal in the lower-left quadrant separates historical-class and Sports-Fitness errors when both beat the true class. Colors show the winning class group, and shapes identify the two Photo classes. After adaptation, 26 of these nodes retain a positive historical margin but are predicted as Sports-Fitness.

Continuous boundary trajectories. Figure 4 and Figure 8c,d use the same fixed 16-node training minibatch from CiteSeer, Task 2 of the four-task setting. Starting from the saved task-1 readout, we replay the first 120 task-2 updates in their original data order to obtain a state inside the protection boundary. The two update rules independently optimize the fixed minibatch from this state, sharing the support, response definition, and budgets. Their first 17 updates produce identical parameters, losses, and update norms. Relative step 0 is the next update, whose initial task KL is 0.004827, above 95% of its 0.005 budget. The local step-radius budget is 0.002 through step 0, then decreases linearly to 0.0002 over 32 further updates; the task-level KL budget remains fixed. All accepted updates satisfy both nonlinear KL checks. At the final point, training loss is 1.26672 for radial shrinking and 1.06853 for tangent correction. These curves measure optimization on the fixed training minibatch; task-level test results are reported separately below.

Single-step boundary diagnostic. We additionally probe a saved task-2 hierarchical/radial checkpoint from the four-task setting to compare individual feasible updates. The task KL is at its 0.005 limit. Both update rules use the same first 16-node training minibatch, support, gradient, and budgets. The raw proposal violates the task limit (0.005062); radial scaling yields an L2 step of 1.29×10^{-7}, whereas the feasible tangent step has length 0.00713 and lowers the batch cross-entropy from 1.782 to 1.741. The corrected step has local KL $1.39\times10^{-6} < 2\times10^{-6}$. Tangent correction reserves inward slack and is followed by nonlinear feasibility checks, so the accepted step need not lie exactly along the first-order tangent. This one-step diagnostic measures update feasibility and batch loss; the full-task comparison is reported separately in Section 6.6.

A.3 Hierarchical response and conditional prediction stability

Fix a task with nonempty historical and current class sets, and suppress the input and task indices. All probabilities below are strictly positive for finite logits. Expanding the KL of the hierarchical response gives

$$\begin{aligned} D_H &= \sum_{o\in\mathcal{O}_t} p_o^a \log \frac{p_o^a}{p_o} + q_N^a \log \frac{q_N^a}{q_N} \\ &= \sum_{o\in\mathcal{O}_t} q_O^a r_O^a(o) \left(\log \frac{q_O^a}{q_O} + \log \frac{r_O^a(o)}{r_O(o)} \right) + q_N^a \log \frac{q_N^a}{q_N} \\ &= D_{\mathrm{KL}}((q_O^a, q_N^a) \,\|\, (q_O, q_N)) + q_O^a D_{\mathrm{KL}}(\mathbf{r}_O^a \| \mathbf{r}_O). \end{aligned} \tag{19}$$

Theorem 1: statement and proof. For a reference class $c \in O$, define

$$\delta_R = p_c^a - \max\left\{ \max_{o \in O \setminus \{c\}} p_o^a,\ q_N^a \right\} > 0, \tag{20}$$

where the inner maximum is zero if $O = \{c\}$. If $D_H < \delta_R^2/2$, the updated unified prediction remains c, regardless of redistribution within N. When c is the true label, this excludes a correct-to-incorrect change at this input. *Proof.* Pinsker's inequality gives $\eta = \mathrm{TV}(\mathbf{r}^a, \mathbf{r}) \le \sqrt{D_H/2}$, so every response coordinate changes by at most η. For each other historical class o, $p_c - p_o \ge \delta_R - 2\eta > 0$. Likewise, $p_c - q_N \ge \delta_R - 2\eta > 0$. Since $p_n \le q_N$ for each $n \in N$, class c exceeds every candidate. ∎

This certificate is sufficient, not necessary: reference correctness alone does not imply $\delta_R > 0$, because the certificate compares one historical class against the entire current-group mass. At the response level, probabilities over historical A/B and current C/D can change from $(0.40, 0.10, 0.25, 0.25)$ to $(0.40, 0.10, 0.49, 0.01)$. The hierarchical response stays $(0.40, 0.10, 0.50)$ and $D_H = 0$, yet the winner changes from A to C. Thus preserving aggregate competition alone cannot guarantee a fixed strongest-competitor margin.

Separate conditional and group guarantees. In particular, $D_{\mathrm{KL}}(\mathbf{r}_O^a \| \mathbf{r}_O) \le D_H / q_O^a$. Suppose there are at least two historical classes and the reference conditional distribution has a unique maximizer c_1, with gap

$$\Delta_O = r_O^a(c_1) - \max_{o \ne c_1} r_O^a(o) > 0. \tag{21}$$

If $D_H < q_O^a \Delta_O^2/2$, then the highest-scoring historical class remains c_1. To see this, Pinsker's inequality gives $\mathrm{TV}(\mathbf{r}_O^a, \mathbf{r}_O) \le \sqrt{D_{\mathrm{KL}}(\mathbf{r}_O^a \| \mathbf{r}_O)/2}$. The probability of any single class changes by at most this total variation. Hence every pairwise gap between c_1 and another historical class remains positive when $2\,\mathrm{TV}(\mathbf{r}_O^a, \mathbf{r}_O) < \Delta_O$. This establishes stability of the historical maximizer, not correctness with respect to the true label or dominance over current classes.

One can also obtain a conservative condition for a historical class to remain globally dominant. Let $\epsilon_m = D_{\mathrm{KL}}((q_O^a, q_N^a) \| (q_O, q_N))$, $\epsilon_c = D_{\mathrm{KL}}(\mathbf{r}_O^a \| \mathbf{r}_O)$, $\eta_m = \sqrt{\epsilon_m/2}$, $\eta_c = \sqrt{\epsilon_c/2}$, and $r_*^a = \max_o r_O^a(o)$. Writing $[z]_+ = \max(z, 0)$, the condition

$$[q_O^a - \eta_m]_+ [r_*^a - \eta_c]_+ > q_N^a + \eta_m \tag{22}$$

ensures that a historical-class probability exceeds every current-class probability. The left-hand side lower-bounds the probability of the reference best historical class; the right-hand side upper-bounds total current mass and therefore its largest individual probability. Combining this condition with conditional ranking stability preserves the reference best historical class under unified prediction. The condition is sufficient and can be loose because it bounds the strongest current competitor by the entire group mass.

These statements are pointwise. For the empirical source-weighted distribution w used in training, an average bound $\mathbb{E}_w[D_H] \le \epsilon$ implies only

$$\Pr_{x \sim w}\{D_H(x) \ge \tau\} \le \epsilon/\tau, \qquad \tau > 0, \tag{23}$$

by Markov's inequality. It does not provide a test-distribution guarantee or uniform conditional protection when q_O^a is small.

A.4 Implicit Fisher products

All quantities in this subsection are evaluated at the current iterate $\boldsymbol{\theta}_k$. For support input x, let $\mathbf{z}(x) = (s_c(x))_{c \in \mathcal{C}_{\le t}}$, $\mathbf{y} = \mathbf{W}_k \widetilde{\mathbf{h}}$, $\boldsymbol{\nu} = \mathbf{y}/\|\mathbf{y}\|_2$, and let $\mathbf{U}$ have normalized class embeddings as rows. For a parameter direction $\mathbf{v} = \mathrm{vec}(\mathbf{V})$, the score Jacobian product is

$$\dot{\mathbf{z}} = \mathbf{J}_x \mathbf{v} = \gamma \mathbf{U} \frac{(\mathbf{I} - \boldsymbol{\nu}\boldsymbol{\nu}^\top)\mathbf{V}\widetilde{\mathbf{h}}}{\|\mathbf{y}\|_2}. \tag{24}$$

The implementation clamps the norm used in these derivative calculations below by 10^{-8}. For nonempty historical and current groups, write $\mathbf{r}_O$ and $\mathbf{r}_N$ for their respective conditional softmax

distributions and set $\mathbf{b} = [\mathbf{r}_O; -\mathbf{r}_N]$, with historical classes listed first. The response Fisher in logit coordinates is

$$\mathbf{H}_x = q_O q_N \mathbf{b}\mathbf{b}^\top + \begin{bmatrix} q_O\big(\operatorname{Diag}(\mathbf{r}_O) - \mathbf{r}_O\mathbf{r}_O^\top\big) & 0 \\ 0 & 0 \end{bmatrix}. \tag{25}$$

The first term measures local curvature of aggregate group competition; the second is the historical conditional Fisher weighted by q_O. Consequently,

$$\mathbf{G}_k\mathbf{v} = \frac{1}{K}\sum_{r=1}^{K}\frac{1}{|\mathcal{S}_r|}\sum_{x\in\mathcal{S}_r}\mathbf{J}_x^\top\mathbf{H}_x(\mathbf{J}_x\mathbf{v}) + \lambda\mathbf{v}. \tag{26}$$

The conditional curvature product is computed as $\mathbf{r}_O \odot \big(\dot{\mathbf{z}}_O - (\mathbf{r}_O^\top\dot{\mathbf{z}}_O)\mathbf{1}\big)$, without materializing its class-by-class matrix. Likewise, Jacobian products and their transposes are evaluated through the normalized readout. The source weights are $1/(K|\mathcal{S}_r|)$ per row. No dense parameter Fisher or Hessian is stored. When the historical group is empty, the response is constant and only damping remains.

A.5 Support construction

The initial support contains external unlabeled reference features, with at most 1024 retained rows. At the end of task t, a seeded random permutation of the current training features selects at most 512 rows without replacement. These form an additional historical source. Once any historical source exists, the external reference retains at most 512 rows. The remaining portion of the 1024-row budget is apportioned among historical sources in proportion to their currently retained row counts, using integer quotas and deterministic remainder allocation. When capacity permits, every historical source retains at least one row. Within a source, deterministic evenly spaced indices select the retained rows. Repeated compression can progressively reduce early-task coverage; the memory is not a balanced reservoir over all historical training samples.

Stored source metadata identifies the task, source, and candidate classes, but individual historical node labels are not used for supervision. Although checkpoints also retain response-related state, each task recomputes the constraints on surviving feature rows with its expanded candidates and task anchor. They do not compare distributions with mismatched candidate sets. The feature-row budget excludes class embeddings, reference responses, readout anchors, and temporary optimization buffers. Feature storage is $O(M(d+1))$, and the readout contains $m(d+1)$ parameters.

A.6 PROPOSAL CORRECTION AND FEASIBILITY SEARCH

Algorithm 1: HiRP training and inference

Inputs: frozen node features $\mathbf{h}$, class-text encoder, pretrained readout $\mathbf{W}$; current labeled task data; external unlabeled support; $M = 1024$, $\delta = 0.002$, $\rho = 0.1$, $\lambda = 0.01$.
Initialize: retain at most M external feature rows; set $L = \delta^2/2$, $B = \rho^2/2$.

1. **For each arriving task t, repeat steps 2–8:** encode newly arrived class descriptions; form $\mathcal{O}_t$, $\mathcal{N}_t$, and $\mathcal{C}_{\le t}$.
2. Freeze the task support $\mathcal{M}_t$ and set $\boldsymbol{\theta}_0 = \mathrm{vec}(\mathbf{W})$. Evaluate $\mathbf{r}_t(x; \boldsymbol{\theta}_0)$ on the expanded candidates for every support row.
3. **For each current-data minibatch in the three training epochs, repeat steps 4–7:** compute the label-smoothed current cross-entropy gradient $\mathbf{g}_k$; set $\mathbf{G}_k = \mathbf{F}_k + \lambda \mathbf{I}$ using the current-iterate response.
4. Check that $\widehat{D}_t(\boldsymbol{\theta}_0, \boldsymbol{\theta}_k) \le (1 + 10^{-5})B$; otherwise raise an error. Compute $\mathbf{v}_k \approx \mathbf{G}_k^{-1}\mathbf{g}_k$ by CG (at most 12 steps); form $\mathbf{d}_k$ using Equation (11), flooring $\mathbf{g}_k^\top \mathbf{v}_k$ at 10^{-20}. A zero gradient produces a zero proposal.
5. If $\widehat{D}_t(\boldsymbol{\theta}_0, \boldsymbol{\theta}_k) \ge 0.95B$, compute $\mathbf{a}_k$ and $\mathbf{u}_k \approx \mathbf{G}_k^{-1}\mathbf{a}_k$. If $\mathbf{a}_k^\top \mathbf{d}_k > 0$, $\mathbf{a}_k^\top \mathbf{u}_k > 10^{-20}$, and the projected norm exceeds 10^{-12}, form Equations (12) and (27) and rescale into the local metric ball. Replace $\mathbf{d}_k$ with this correction only when its inner product with $\mathbf{g}_k$ is negative.
6. Set $\boldsymbol{\theta}' = \boldsymbol{\theta}_k + \mathbf{d}_k$. If local KL exceeds L, shrink the displacement by 48 interval-search steps retaining an evaluated feasible endpoint. If task KL then exceeds B, repeat this search on the remaining displacement.
7. Recompute both nonlinear KL values. Commit $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}'$ only if finite and both pass relative tolerance 10^{-4}; otherwise raise an error without committing.
8. **At task end:** set $\mathbf{W}$ to the final iterate reshaped as a readout matrix; sample up to 512 current-training features without their supervision labels, append their source metadata, and compress to M rows by Appendix A.5.

Inference: use the shared readout and all arrived class descriptions in Equation (2); no task routing or score calibration.

Conjugate gradients start from zero and stop after 12 iterations or when the residual norm is at most 10^{-5} times its initial value. The proposal denominator uses $\mathbf{g}_k^\top \mathbf{v}_k$ with a numerical floor of 10^{-20}. No historical supervised loss is included in $\mathbf{g}_k$; the current minibatch cross-entropy uses label smoothing of 0.2.

Set $B = \rho^2/2$ and $L = \delta^2/2$. If the current task divergence is at least $0.95B$, compute its gradient $\mathbf{a}_k$ and the CG approximation $\mathbf{u}_k$ to $\mathbf{G}_k^{-1}\mathbf{a}_k$. The tangent correction is attempted only when $\mathbf{a}_k^\top \mathbf{d}_k > 0$ and $\mathbf{a}_k^\top \mathbf{u}_k > 10^{-20}$. After projection, a metric norm exceeding 10^{-12} is required before normalization. We normalize the tangent proposal and introduce inward slack using

$$\bar{\mathbf{d}}_k = \delta \frac{\mathbf{d}_k^{\tan}}{\|\mathbf{d}_k^{\tan}\|_{\mathbf{G}_k}}, \qquad \mathbf{d}_k^{\mathrm{in}} = \bar{\mathbf{d}}_k - \frac{\|\bar{\mathbf{d}}_k\|_{\mathbf{G}_k}^2}{\mathbf{a}_k^\top \mathbf{u}_k}\mathbf{u}_k. \tag{27}$$

The inward adjustment in Equation (27) is then rescaled by $\min(1, \delta/\|\mathbf{d}_k^{\mathrm{in}}\|_{\mathbf{G}_k})$. It replaces the original proposal only if its inner product with $\mathbf{g}_k$ is negative. With an exact inverse, the tangent term removes the linearized outward component, and the inward adjustment contributes $-\|\bar{\mathbf{d}}_k\|_{\mathbf{G}_k}^2$ to the task-divergence directional derivative. This supplies first-order inward slack, rather than an analytic bound on all higher-order terms.

The solver first evaluates the local KL of the proposed update. If it exceeds L, it searches a scalar multiplier in $[0, 1]$ for 48 bisection iterations, retaining an evaluated feasible lower endpoint. It next checks the task KL and, if necessary, applies the same procedure to the remaining displacement. Both divergences are recomputed for the final proposal. KL evaluations use double-precision log-probabilities; accepted updates must pass both budget checks within the implementation's relative tolerance of 10^{-4}. The final verification matters because the nonlinear KL level sets need not be convex along an arbitrary parameter ray. The interval search is a feasibility procedure and does not establish a maximal feasible step or a general convergence guarantee. If the final proposal is nonfinite or fails a budget check, the implementation raises an error before committing it. It also

rejects a current state already outside the task budget beyond its tolerance. A zero gradient gives a zero CG proposal through the denominator floor; no update is required in that case. The two-reference KL constraints and exact-inverse tangent projection specify the mathematical construction; the 95% trigger, inward adjustment, finite CG solve, and interval searches specify its numerical realization.

A.7 First-task optimization and attribution

At the first task, the response is constant and $\mathbf{F}_k = 0$. For a nonzero gradient and an exact inverse, Equation (11) reduces to

$$\mathbf{d}_k = -\frac{\delta}{\sqrt{\lambda}} \frac{\mathbf{g}_k}{\|\mathbf{g}_k\|_2}, \qquad \|\mathbf{d}_k\|_2 = 0.02. \tag{28}$$

This normalized step is the first-task specialization of the same update rule, before historical response constraints become active. Mean first-task accuracies are 81.49% for HiRP versus 78.84% for SGD in the four- and six-task settings, and 83.77% versus 79.68% in the ten-task setting; the four- and six-task settings share the same first task.

The final FA/AA comparisons evaluate the complete algorithms, including this optimization difference. The 32-step intervention in Section 6.6 starts both methods from the same later HiRP checkpoint and matches parameter-step norms, isolating direction effects at those selected states. The response/update factorial ablation holds the data, initialization, support rules, and numerical budgets fixed across its variants. Together these analyses distinguish full-system accuracy, direction effects at common states, and conditional response/solver effects.

Full-sequence controls from a common first-task checkpoint. In the ten-task setting, all methods start task 2 from the same HiRP task-1 checkpoint, with task-1 accuracy 83.77%, and complete tasks 2–10. They share frozen graph and text encoders, the trainable readout, training batches, three epochs per task, batch size 16, and label smoothing 0.2. The Fisher-proposal control retains the hierarchical response, feature support, damped Fisher metric, and normalized proposal rule, while removing both KL feasibility checks, radial shrinking, and tangent correction. SGD uses its standard learning rate of 5×10^{-4}.

Table 4: Full-sequence comparisons in the ten-task setting from a shared task-1 checkpoint. Zero-shot mean is the unweighted accuracy over History, Computer, and Instagram after task 10. Values are percentages; higher is better.

Method	AA ↑	Zero-shot mean ↑
SGD	50.71	56.82
Fisher proposal without KL constraints	56.84	55.20
HiRP (ours)	**69.74**	**58.27**

HiRP reaches 69.74% AA versus 56.84% for the Fisher-proposal control, a 12.91-point gain (Table 4). Restricting the final average to tasks 2–10 yields 69.11% versus 58.04%. For SGD, replacing its own task-1 checkpoint with HiRP's increases final AA from 49.83% to 50.71%, only 0.88 points. These results support an additional benefit from response constraints and their update corrections beyond the first-task starting point and the Fisher proposal alone.

Continual accuracy and zero-shot transfer with matched frozen encoders. We evaluate the same final checkpoints on History, Computer, and Instagram using each domain's complete class set, without target-domain adaptation. These domains contribute neither continual-training examples nor feature support and are not used for model selection; the reported zero-shot mean weights the three domains equally. Even with the same frozen encoders, trainable readout, and task-1 starting point, SGD is lower than HiRP in both AA (50.71% versus 69.74%) and mean zero-shot accuracy (56.82% versus 58.27%). Thus, in this matched comparison, HiRP achieves a better balance between learning arrived classes and retaining transfer to held-out graph domains.

Common-start, norm-matched intervention. We intervene at task 3 of the four-task setting and task 5 of the six- and ten-task settings. For each run, HiRP and SGD start from the same HiRP checkpoint and receive the same 32 current-training batches. Each SGD update is rescaled to match the corresponding HiRP update's Frobenius norm. For fixed candidate sets, define

$$m_t(x; \mathbf{W}) = \max_{o \in \mathcal{O}_t} s_o(x; \mathbf{W}) - \max_{n \in \mathcal{N}_t} s_n(x; \mathbf{W}). \tag{29}$$

The crossing rate is the fraction of all historical test samples with $m_t > 0$ before and $m_t < 0$ after the intervention. It measures a change in the winning group, not necessarily a correct-to-incorrect prediction. HiRP reduces this rate in all three streams; in the ten-task setting it falls from 2.627% to 0.254%, while current-class global accuracy changes from 91.86% to 91.71%. Historical-sample global accuracy improves by 0.56, 0.45, and 2.94 points in the four-, six-, and ten-task settings, respectively; current-sample accuracy decreases by 0.05, 0.18, and 0.15 points. Matching update norms rules out smaller per-step parameter displacement as the sole explanation at these common starting states.

Case-study diagnostics. Figure 8a,b uses all 1180 historical test nodes at task 5 of the ten-task setting and the same 32 batches from 129 current training nodes. The mass-only control shares the starting readout, support, and numerical KL budgets with HiRP; SGD* matches HiRP's step norm at each iteration. Mean old-only accuracy changes by -2.60, -0.62, and $+0.11$ points for SGD*, Mass, and HiRP, respectively. Their old-to-new crossing rates are 2.63%, 0.54%, and 0.25%; new-class accuracies are 91.86%, 91.76%, and 91.71%. Tangent correction is inactive during these short interventions, so panels a,b examine the response and its induced update rather than activation of the tangent correction.

Additional single-step and complete-task diagnostics use CiteSeer, Task 2 of the four-task setting. The single-step experiment uses the boundary checkpoint described in Appendix A.2; retained length is the accepted norm divided by the raw proposal norm, and loss decrease is relative to the initial batch loss. The raw proposal norm is 0.0076947, radial norm is 1.2910×10^{-7}, and tangent norm is 0.0071312. The initial loss 1.78164 becomes 1.78164 under radial shrinking and 1.74053 under tangent correction. Separately, complete task-2 runs of hierarchical radial and tangent updates yield new-class test accuracies of $64.47 \pm 0.90\%$ and $66.67 \pm 0.80\%$, and old-class accuracies of $81.18 \pm 1.76\%$ and $80.87 \pm 1.38\%$. The new-class accuracy comparison uses complete-task training.

A.8 What the four-cell ablation controls

Table 2 is a 2×2 factorial comparison. The response factor selects aggregate group competition alone or additionally preserves historical conditional discrimination; the solver factor selects scalar shrinking or tangent correction followed by feasibility checks. Within each paired comparison, all four variants share the pretrained readout, frozen feature cache, class order, current-data batches, support-selection rule, local/task radii, damping, and training epochs. Their first-task readout tensors are identical across variants in every paired run. Paired protocol and first-task equality records are supplied in `reproducibility/ablation_controls.json`.

Holding the solver fixed compares the two response designs; holding the response fixed compares the two solvers. For example, with tangent updates fixed, the hierarchical response improves FA over the binary response by 1.59, 1.22, and 13.12 points on whole4, whole6, and global10. With the hierarchical response fixed, tangent correction improves FA over radial shrinking by 4.57, 4.21, and 1.07 points. The gains interact: on global10, tangent minus radial is -2.18 points with binary mass but $+1.07$ with the hierarchical response. The interaction makes these conditional contrasts nonadditive.

The response change also changes the Fisher geometry and accepted displacement, so it measures the complete response-induced update under equal numerical budgets.

B Additional Experimental Details and Results

B.1 Data and protocols

Table 5 specifies the supplied splits used for adaptation and evaluation. Cora denotes the seven-class citation graph with 2,708 nodes; it is not Cora-Full. Photo denotes the twelve-class text-

Table 5: Exact dataset instances and prescribed splits. The single-graph schedule lists the classes introduced at each task; the two external graphs are used in the six-task setting. WikiCS retains its official partially assigned split.

Dataset	Nodes	Classes	Train	Valid.	Test	Single schedule
Cora	2,708	7	1,624	542	542	2, 2, 3
CiteSeer	3,186	6	1,911	637	638	2, 2, 2
WikiCS	11,701	10	580	1,769	5,847	2, 2, 2, 2, 2
Ele-Photo	48,362	12	29,017	9,672	9,673	2, 2, 2, 2, 2, 2
Sports-Fitness	173,055	13	34,611	17,305	121,139	–
Actor (HetGB)	4,416	5	2,116	1,411	889	–

attributed product graph used in our GraphCLIP pipeline. Classes from different graphs receive disjoint identifiers. The full training split is used within each task's class subset, and the train and test sample pools are disjoint. The configuration names `whole4`, `whole6`, and `global10` denote the 4-dataset/4-task, 6-dataset/6-task, and 4-dataset/10-task settings, respectively. Result files store FA under the key `Global`. The exact class-ID maps and per-task cumulative candidates are included in `reproducibility/protocols/`; `reproducibility/splits/` contains the four original-domain node-ID partitions. Global IDs 0–6, 7–12, 13–22, and 23–34 refer respectively to Cora, CiteSeer, WikiCS, and Photo; whole6 appends Sports-Fitness (35–47) and Actor (48–52). The ten tasks of global10 contain $4, 3, 3, 3, 4, 3, 3, 4, 4, 4$ classes in that order; the supplied TSV maps these ranges to the original class identities.

Cora and Photo use fixed NumPy default-generator seed-0 partitions; CiteSeer uses the saved NumPy RandomState seed-0 partition. WikiCS uses official mask column 0, with 3,505 nodes outside the supervised train/validation/test assignment. Sports-Fitness uses the supplied aligned `Fitness.csv/Fitness.pt` graph and a fixed 20/10/70 split generated with PyTorch seed 88. Actor uses the masks in the supplied HetGB `Actor.npz` archive. The source files and original-node split identities for both external graphs are described in `external_dataset_identity.json`; compact cache indices are not substituted for original node identities.

Graph context and label access. These are fixed, transductive graph instances with sequential label access. For the four original domains, the GraphCLIP pipeline uses cached root-node graphs derived from the supplied `target_data` edge lists, augmented with 32-dimensional random-walk positional encodings. For the two external domains, root-node graphs use walks of 256 steps (Sports-Fitness) or 64 steps (Actor), restart probability 0.8, and the same positional-encoding dimension. Unlabeled neighbors may belong to other tasks or splits: the graph context is not restricted to currently labeled nodes. The supervised loss uses only current-task training roots, and evaluation uses cumulative candidates for every test root. Historical training roots enter the bounded support only after their task; validation and test roots are not added as support rows. This setting therefore does not claim an inductive protocol in which future nodes, edges, or attributes are unavailable.

Frozen model and class semantics. The checkpoint is `pretrained_graphclip.pt`; its architecture and readout settings are recorded in `model_and_reference.json`. Its graph encoder has 12 GPS layers, hidden size 1024, and 8 attention heads; 384-dimensional node inputs and 32-dimensional positional encodings yield a 2048-dimensional feature by concatenating mean and center-node representations. The trainable affine readout is 384×2049, including its bias, with 786,816 parameters. The text encoder/tokenizer is `sentence-transformers/all-MiniLM-L6-v2`, with attention-mask mean pooling, maximum class-text length 512, and unit-normalized 384-dimensional class embeddings. We retain the checkpoint's fixed cosine-logit scale $\gamma = 14.7611256$. Encoders run in evaluation mode; the frozen feature cache is shared across methods that adapt the readout.

Class inputs concatenate the dataset template, class name, and supplied description. The templates are "this paper has a topic on {class}" (Cora), "good paper of {class}" (CiteSeer), "it belongs to {class} research area" (WikiCS), "this product belongs to {class}" (Photo/Sports-Fitness), and "this actor belongs to {class}" (Actor). The complete strings, including whitespace and descriptions, are

included in `class_texts.json`. These semantic prototypes are frozen; the active classifier at task t uses only the arrived candidate IDs. Precomputing the immutable text bank does not expose future-class labels or future learned model parameters to the task's loss.

External reference and feature memory. The initial external pool contains 512 root graphs: 256 from arXiv-2023 and 256 from PubMed, selected without replacement using seeds 20260721 and 20260722. It is the recorded `graph_zscl_reference_pool_512_rwpe32.pt` asset, not the separate five-source balanced pool used by other exploratory runs. The source datasets are distinct from the six target datasets; this is dataset/ID separation, not a claim of exhaustive semantic deduplication across corpora. The reference may share source domains with GraphCLIP pretraining. Its graph IDs, source counts, and feature-memory settings are included in `model_and_reference.json`. Actual initial support is 512 rows, with the 1024-row cap applied after historical training features are added. One fully occupied float32 feature support uses 8,392,704 bytes (about 8.00 MiB); the readout uses 3,147,264 bytes (about 3.00 MiB). Reference responses, anchors, class embeddings, and solver workspaces are additional state and are not included in those two counts.

B.2 Baseline implementations

GraphCLIP joint fine-tuning. GraphCLIP starts from the same pretrained checkpoint as HiRP and jointly updates the graph encoder, shared readout, text encoder, and logit scale. We use the bidirectional graph–text pairwise cross-entropy implementation released with ADAligner (Liu et al., 2026) in its public downstream code.[1] Optimization uses AdamW, learning rate 10^{-5}, weight decay 10^{-5}, batch size 4, five epochs per task, and gradient clipping at 1.0. Each task uses all prescribed current-task training examples and continues from the previous task's final model, with a fresh optimizer. The final epoch supplies the checkpoint; the method uses no LoRA, learned input prompt, replay, or distillation.

Held-out History evaluation. Figures 1 and 7 use the ten-task stream. History contributes neither training examples nor feature support and is not used for model selection. Before continual training ($t = 0$) and after each task, evaluation uses the same 8,311 History test nodes and its complete class set without target-domain adaptation. GraphCLIP recomputes class text embeddings using its updated text encoder, while HiRP uses its frozen text encoder and current readout.

Readout and prototype baselines. SGD updates the shared GraphCLIP readout using current-task cross-entropy. LwF adds distillation of historical-class responses on current inputs, with weight 1 and temperature 2. Both use a learning rate of 5×10^{-4}, three epochs, batch size 16, and label smoothing 0.2. Neither has a historical supervised replay loss. HiRP uses the same current-data training budget, replacing the ordinary readout update with its response-constrained rule. EWC, MAS (Aljundi et al., 2018), and OGD (Farajtabar et al., 2020) are parameter-regularization or projection baselines on the shared readout. For whole4 and global10, their saved checkpoints are reevaluated on the common fixed test split; whole6 is evaluated during its full training sequence. All use the same final metric definitions, while this checkpoint/evaluation provenance is retained in the reproducibility records. Cosine and TEEN use their own graph representations and prototype rules.

The LoRA comparison adapts GraphCLIP using rank-78 adapters in its final graph block and a shared readout, with 2,064,768 trainable parameters in total. It uses three epochs per task, batch size 32, learning rate 5×10^{-4}, and label smoothing 0.2. G2LoRA is a separate GraphCLIP adaptation of the published mechanisms, not a claim to reproduce the original paper's complete system or few-shot results. It uses rank-32 graph and text adapters, four epochs per task, batch size 32, learning rate 10^{-5}, weight decay 0.05, and unsmoothed cross-entropy. Both use all prescribed current-class training samples and cumulative test candidates. Consequently, comparisons share the data protocol but do not equate trainable parameter counts or optimization budgets.

The cross-graph and single-graph comparisons additionally use the corresponding graph/text implementations of GCN (Kipf & Welling, 2017), Cosine, TEEN (Wang et al., 2023), BERT (Devlin et al., 2019), and the explicitly identified variants of TPP (Niu et al., 2024) and SimpleCIL (Zhou

[1] `https://github.com/karmaisacat-13/ADAligner`, commit `5e4c7a3`, `transfer/eval_prompt.py`.

et al., 2025). GCN is the LLM4GCL (Cheng et al., 2025) `BareGNN` implementation instantiated as a two-layer GCN with hidden size 128 for the cross-graph runs; it trains on node input features and edges without the pretrained GraphCLIP graph encoder. BERT sequentially adapts its text model using the upstream LoRA configuration (rank 5, $\alpha = 16$, dropout 0.05). Cosine and TEEN retain their graph representations and prototype rules. SimpleCIL (BERT) adapts BERT with LoRA on the first task, then keeps its representations fixed and updates class prototypes; this differs from the RoBERTa configuration described in G2LoRA. Cross-graph prototype updates use all prescribed current-class training samples.

TPP-style denotes the LLM4GCL-derived adapter with graph-smoothed task profiles, per-node predicted routing, and independent linear classification heads on input features. The executed implementation omits the original TPP graph prompts and GNN classification path; its scores therefore characterize this variant, not a complete reproduction of the original method. Its task heads remain fixed after their respective tasks, but newly available task prototypes can change the routing of historical nodes. Inference never uses a ground-truth task identifier or ground-truth task grouping of test inputs. The adapters use the same node indices, class schedules, and cumulative-candidate evaluation as the readout comparisons; validation-based stopping follows each implementation's configuration. These comparisons retain differing pretraining resources and model capacities. In the six-task diagnostic, GCN's final accuracy averaged over the first five tasks is only 0.19%. The final task accounts for approximately 0.64% of test nodes, contributing to its low sample-weighted FA.

RoBERTa. RoBERTa (Liu et al., 2019) denotes sequential full fine-tuning of RoBERTa-large and its standard dense–tanh classification head on node text. The encoder and head are updated at every task using the complete current training split, without LoRA, prototype replacement, or historical supervised replay. Training cross-entropy and test prediction both use cumulative seen-class candidates; inference does not receive task identities. We use AdamW with learning rate 2×10^{-5} and weight decay 0.01, batch size 32, maximum text length 256, 6% linear warmup followed by linear decay, and gradient clipping at norm 1. Each task runs three to six epochs, with patience two; current validation accuracy selects the checkpoint, with validation cross-entropy breaking ties. Parameters and optimizer state are float32, with float16 autocast and gradient scaling; the full selected checkpoint is restored and checked tensor by tensor before evaluation. On whole6, Sports-Fitness accounts for 87.32% of test nodes: RoBERTa's 75.12% FA therefore coexists with 27.64% AA, underscoring the need to report both metrics.

InfLoRA. InfLoRA† (Liang & Li, 2024) implements the interference-reducing subspace mechanism on the final GraphCLIP GPS attention block, with rank-10 K/V adapters and task-wise linear class heads on 2,048-dimensional graph features. The preceding graph blocks are frozen; the pretrained semantic readout is not used. Current-task cross-entropy trains the new class rows and active low-rank factors; historical classifier rows stay fixed and inference considers all seen classes. Activation subspaces and frozen previous factors carry historical information without supervised replay. Each task uses all prescribed training roots, 20 epochs, batch size 32, Adam with learning rate 5×10^{-4} and zero weight decay, cosine decay, and validation-best checkpoint selection. This is a last-block GraphCLIP implementation of the mechanism, rather than the original ViT benchmark configuration; its layer scope and optimization budget differ from HiRP.

SD-LoRA. SD-LoRA (Wu et al., 2025) implements magnitude–direction decoupling on the final GraphCLIP GPS attention block, using rank-10 Q/V adapters and task-wise linear heads on 2,048-dimensional graph features. The preceding graph blocks are frozen and the pretrained semantic readout is not used. Each task learns its new low-rank factors and current classifier rows, retains previous low-rank directions, and updates their projection-specific magnitudes; historical classifier rows remain fixed. The implementation preserves the effective adapter function when normalizing directions at task boundaries. Training cross-entropy uses current-class candidates, while inference compares all seen classes without task identities or historical supervised replay. All runs use 20 epochs per task, batch size 32, SGD with learning rate 10^{-2}, momentum 0.9, zero weight decay, and current-validation-best checkpoint selection. To prevent overflow on Sports-Fitness, all settings use global gradient clipping at norm 1, float64 gradient-norm accumulation, and finite-value checks; no batches are skipped. Earlier uncorrected attempts are retained separately and excluded. The table uses the name SD-LoRA; these results describe its recorded last-block GraphCLIP implementation, not the original vision benchmark configuration.

GCN with LLM embeddings. $\text{GCN}_{\text{LLMEmb}}$ follows the LLM4GCL feature-enhancement baseline (Cheng et al., 2025). Frozen 4-bit Llama-3.2-3B provides 3,072-dimensional node features through attention-mask mean pooling, with maximum text length 512. A two-layer GCN with hidden size 128 and dropout 0.5 is trained sequentially on these features and graph edges. Each task uses the full current training split, batch size 256, learning rate 10^{-4}, weight decay 5×10^{-4}, and at most 300 epochs with validation-based checkpoint selection. Inference considers all arrived classes without task identities or historical supervised replay.

GraphGPT adaptation. GraphGPT† (Tang et al., 2024) matches multiple graph tokens to source descriptions, then continually tunes a linear projector with the GraphCLIP encoder and Llama-3.2-3B frozen; it uses no language-model LoRA. The shared alignment checkpoint is trained for three epochs on 22,532 matching groups, with 1,187 held-out groups, from the five recorded GraphCLIP source corpora. Downstream runs share this checkpoint and vary the training order. Each task uses full current training data, batch size 16, AdamW with learning rate 2×10^{-4} and zero weight decay, and validation-based selection after three to six epochs. Greedy generation uses the cumulative class names; unrecognized answers are counted as incorrect. The saved protocols record prompt construction and evaluation batches; this is an explicit backbone/source-corpus adaptation, not the official GraphGPT pretrained system.

LLaGA. LLaGA† (Chen et al., 2024) uses the neighborhood-detail (ND) architecture with MiniLM384 node inputs, two-hop sampling with ten neighbors per hop, and the upstream Laplacian positional encoding. A two-layer $495 \to 3072 \to 3072$ projector maps the graph representation to the frozen Llama-3.2-3B input space; no language-model LoRA or GraphCLIP graph encoder is used. The projector is randomly initialized and continually trained on all current training roots, using batch size 16, AdamW at 2×10^{-4} with zero weight decay, and validation-based selection over three to six epochs. Neighbor samples are cached per node within each run, rather than resampled per batch. Greedy decoding uses cumulative class names with unrecognized answers counted as incorrect; final test evaluation uses batch size 32 throughout, while validation retains its recorded per-run batch of 4 or 32. The records include checked projector save/restore events and the evaluation-batch policy. These results describe the ND architecture under this continual protocol and language-model choice, not the official pretrained LLaGA system.

B.3 Comparison conditions and computation

Shared splits, candidate masks, and test identities do not imply identical information access, representations, or optimization cost. Table 6 records these distinctions. Cross-graph LLM4GCL adapters use their upstream default configurations with no per-dataset best-setting override; full effective configurations are in `upstream_default_configs.json`. GCN/Cosine/TEEN use batches of 256 and at most 300 epochs, learning rate 10^{-4} and weight decay 5×10^{-4}. TPP-style trains its independent heads for at most 3000 epochs with batch size 200 and the same rate/decay; the unused original prompt/pretraining options in its YAML do not activate those mechanisms. BERT and SimpleCIL use `google-bert/bert-base-uncased`, learning rate 2×10^{-4}, weight decay 0.05, maximum text length 256, and at most 20 epochs, with batch sizes 10 and 30 respectively. The BERT revision and asset names are listed in the reproducibility records. Their stopping decisions use validation data; BERT trains sequentially, whereas SimpleCIL trains its representation only at the first task. For GCN, BERT, Cosine, TEEN, and SimpleCIL, saving and restoring the validation-selected checkpoint verifies every trainable tensor; empty checkpoints are rejected. Cross-graph prototype routines requesting fixed sample counts use all prescribed current-class training rows. The single-graph GCN, Cosine, TEEN, BERT, and SimpleCIL runs retain their recorded upstream per-dataset configuration overrides and prototype-sampling settings; their complete effective configurations accompany the results. For the native whole6 readout regularizers, EWC and MAS use weight 100, learning rate 5×10^{-4}, three epochs, and batch size 16; their retained low-rank state uses ranks 64 (EWC) and 512 (MAS/OGD). The recorded EWC recipe uses external-reference statistics, while MAS/OGD also retain task-derived statistics; whole4/global10 retain their archived recipe provenance rather than being attributed to this native training entry point.

HiRP performs 6,219, 13,110, and 6,228 current-data updates over three epochs on whole4, whole6, and global10. Each update includes Fisher-vector products, up to 12 CG iterations per solve, an optional second solve, and up to 48 scalar-search iterations per active KL constraint. Inference uses

Table 6: What is and is not shared across the compared implementations. All displayed methods use the prescribed current training roots and cumulative-candidate test protocol; history access and adaptation remain method-specific.

Method family	Updated representation	Historical information used in adaptation
HiRP / four-cell ablation	Shared readout on frozen GraphCLIP features	Bounded historical and external unlabeled features; response constraints.
SGD / LwF	Same readout and frozen feature cache	SGD: current CE only. LwF: previous-model responses on current inputs; no historical-node CE.
EWC / MAS / OGD	Shared readout	Stored importance/projection information under each saved recipe; not the HiRP response constraint.
LoRA / G2LoRA†	Graph adapters/readout; selected graph and text adapters	Published-mechanism adaptations with their own retained state and optimization budgets.
InfLoRA†	Last graph-attention K/V adapters and task-wise linear heads	Activation subspaces and frozen prior factors; no historical supervised replay.
SD-LoRA	Last graph-attention Q/V adapters and task-wise linear heads	Frozen prior low-rank directions with learned magnitudes; no historical supervised replay.
GCN / BERT	GCN and head / text LoRA and head	Sequential current-task training; distinct from the pretrained GraphCLIP readout.
$\text{GCN}_{\text{LLMEmb}}$	GCN and classifier on frozen LLM node embeddings	Sequential current-task training; no historical supervised replay.
RoBERTa	Full RoBERTa-large encoder and classification head	Current-task supervised fine-tuning; no historical supervised replay.
Cosine / TEEN / SimpleCIL	Method-specific representations and prototype rules	Retained class prototypes; SimpleCIL fixes BERT after first-task adaptation.
TPP-style	Independent input-feature linear heads	Retained heads and graph-derived task profiles; predicted per-node routing.
GraphGPT†	Linear projector; frozen GraphCLIP and Llama-3.2-3B	Shared source matching initialization; sequential current-task tuning without historical supervised replay.
LLaGA†	ND projector; MiniLM384 inputs and frozen Llama-3.2-3B	Per-seed cached neighbor samples; sequential projector tuning without historical supervised replay.

Table 7: Recorded training time per complete sequence in minutes (mean $\pm$ sample standard deviation). Feature precomputation, validation, and testing are excluded.

Method	4 datasets / 4 tasks	6 datasets / 6 tasks	4 datasets / 10 tasks
GraphGPT†	250.9 ± 0.8	742.2 ± 128.2	217.5 ± 26.4
LLaGA†	174.5 ± 59.7	523.9 ± 67.5	142.2 ± 19.6
InfLoRA†	19.4 ± 2.0	31.8 ± 3.6	27.2 ± 1.7
SD-LoRA	21.2 ± 0.3	37.2 ± 5.0	34.2 ± 2.6
HiRP (ours)	6.7 ± 1.1	16.2 ± 5.5	7.2 ± 2.5

the same frozen encoder and shared readout as its readout baselines, without the training support or solver. Table 7 compares recorded training times with GraphGPT, LLaGA, InfLoRA, and SD-LoRA under their documented configurations. Baseline times sum the recorded training epochs; HiRP uses task timers that also include support updates and checkpoint saving. HiRP has lower recorded training time in all three streams, with feature precomputation, validation, and testing excluded. Training budgets, adapted modules, and device loads differ across methods.

The recorded cached-feature environment is Python 3.10.20, PyTorch 2.2.2 with CUDA 12.1, PyG 2.7.0, Transformers 4.44.2, and V100 32-GB hardware; its version record is supplied with the reproducibility materials. All four-cell variants reset current-batch RNG to seed $+100000t$ and support selection to seed $+300000t$. HiRP uses fixed three-epoch final readouts, while heterogeneous base-

Table 8: Single-graph class-incremental classification. All entries use the same class splits and cumulative-candidate evaluation. FA and AA are percentages; bold denotes the largest mean among the displayed methods. The displayed methods retain their respective backbones.

Method	Cora		CiteSeer		WikiCS		Photo	
	FA ↑	AA ↑	FA ↑	AA ↑	FA ↑	AA ↑	FA ↑	AA ↑
GCN	54.61 ± 0.18	31.42 ± 0.11	32.45 ± 0.31	31.36 ± 0.30	43.76 ± 3.69	36.63 ± 4.14	14.58 ± 0.04	16.39 ± 0.05
Cosine	65.87 ± 1.39	62.54 ± 0.98	42.79 ± 4.27	38.87 ± 4.34	58.07 ± 1.38	51.58 ± 1.85	56.34 ± 1.86	60.00 ± 3.01
TEEN	59.04 ± 4.36	54.27 ± 3.70	54.49 ± 2.91	48.21 ± 2.67	59.76 ± 1.51	55.31 ± 0.60	49.75 ± 3.19	52.89 ± 0.99
TPP-style	68.20 ± 0.59	63.53 ± 0.47	70.64 ± 0.59	69.10 ± 0.55	58.10 ± 1.10	58.61 ± 1.23	50.61 ± 0.04	66.57 ± 0.09
SGD	67.65 ± 0.28	56.49 ± 0.70	63.85 ± 0.50	60.82 ± 0.49	62.22 ± 0.06	55.13 ± 0.06	28.68 ± 0.05	28.85 ± 0.07
LwF	68.88 ± 0.11	59.93 ± 0.26	64.94 ± 0.48	60.72 ± 0.52	62.46 ± 0.12	54.38 ± 0.12	28.65 ± 0.31	31.53 ± 0.11
LoRA	69.07 ± 0.38	59.86 ± 0.37	68.86 ± 0.24	65.97 ± 0.28	66.28 ± 0.25	60.07 ± 0.22	34.04 ± 0.03	37.92 ± 0.05
HiRP (ours)	$\mathbf{75.09 \pm 0.18}$	$\mathbf{72.57 \pm 0.42}$	$\mathbf{73.88 \pm 1.27}$	$\mathbf{70.71 \pm 1.79}$	$\mathbf{74.22 \pm 0.08}$	$\mathbf{72.18 \pm 0.12}$	$\mathbf{71.23 \pm 0.66}$	$\mathbf{75.45 \pm 1.70}$

Table 9: Additional language-model comparisons on CiteSeer and Photo. Entries are means and sample standard deviations, using the same evaluation splits as Table 8.

Method	CiteSeer		Photo	
	FA ↑	AA ↑	FA ↑	AA ↑
BERT	31.24 ± 0.36	30.20 ± 0.35	14.29 ± 0.01	16.06 ± 0.01
SimpleCIL (BERT)	57.58 ± 4.65	57.75 ± 4.23	31.98 ± 6.98	43.63 ± 5.02
HiRP (ours)	$\mathbf{73.88 \pm 1.27}$	$\mathbf{70.71 \pm 1.79}$	$\mathbf{71.23 \pm 0.66}$	$\mathbf{75.45 \pm 1.70}$

lines retain their documented stopping and adaptation rules. The supplied configurations and result records identify the evaluated implementations.

Reproducibility materials. The accompanying `reproducibility/` directory contains protocols, split identities, configurations, five-run main-table results, and derived diagnostic data; `tools/` provides figure reconstruction scripts. Its README distinguishes outputs reproducible from these records from training runs requiring the original feature caches, pretrained assets, and complete training/evaluation source. This is a record and analysis supplement, not a self-contained end-to-end training release.

Scope and limitations. The three streams test different domain/class schedules but share several datasets. Appendix B.9 examines the local and task radii, damping, and support capacity; sensitivity to other orders and backbones remains to be evaluated. The benchmark informed method development, and the documented baseline recipes need not be optimal for every method. Historical reverse-order runs add calibration or group supervision and are outside the evaluated HiRP configuration. At the method level, aggregate current mass permits within-group changes, bounded support can miss historical regions, and task-entry references can already contain candidate-insertion errors. These limitations separate empirical response preservation from an unconditional guarantee against forgetting.

B.4 Single-graph classification

HiRP attains the highest FA and AA among the methods in Table 8 on all four graphs. Its FA is 75.09%, 73.88%, 74.22%, and 71.23% on Cora, CiteSeer, WikiCS, and Photo, respectively. The same configuration is used without substituting dataset-specific exploratory variants. Table 9 additionally reports BERT and SimpleCIL (BERT) on CiteSeer and Photo using the same evaluation splits. These results extend the classification comparison to individual graphs.

Table 10: Single-graph certificate coverage. N_{ref} counts reference-correct historical sample–transition pairs on expanded candidates; Trans. counts transitions across runs. C_+ and C_{cert} are percentages (transition mean within each run, then mean $\pm$ sample SD). The last column reports pooled correct-to-incorrect changes among certified pairs.

Dataset	Trans.	N_{ref}	C_+ (%)	C_{cert} (%)	Forgotten / certified
Cora	6	724	97.63 ± 0.79	88.89 ± 2.54	0 / 642
CiteSeer	6	1,107	98.28 ± 1.80	91.82 ± 1.91	0 / 1,032
WikiCS	12	21,565	98.21 ± 0.16	96.42 ± 0.24	0 / 20,717
Photo	15	44,367	96.41 ± 0.18	80.86 ± 0.48	0 / 35,906

B.5 Empirical coverage of the prediction-stability certificate

We evaluate the practical coverage of Theorem 1 on the four single-graph HiRP settings in Table 8, covering 12 sequences and 39 task transitions. For each transition, we compare saved test logits from the task-entry and task-end readouts on identical historical nodes and the same expanded candidate set $\mathcal{O}_t \cup \mathcal{N}_t$. The reference-correct set $\mathcal{I}_{t,s}$ contains historical test nodes whose task-entry prediction over this expanded set equals their true label y. Correctness restricted to historical candidates is not sufficient for inclusion. We compute δ_R in Eq. (20) with $c = y$ and the pointwise D_H from the paired logits, retaining the original score scale without temperature adjustment. Ground-truth labels are used only for this offline diagnostic.

Coverage and aggregation. Writing $N_{\mathrm{ref}} = |\mathcal{I}_{t,s}|$, the two coverage rates are

$$C_+ = \frac{\sum_{x \in \mathcal{I}_{t,s}} \mathbf{1}\{\delta_R(x) > 0\}}{N_{\mathrm{ref}}}, \qquad C_{\mathrm{cert}} = \frac{\sum_{x \in \mathcal{I}_{t,s}} \mathbf{1}\{\delta_R(x) > 0,\ D_H(x) < \delta_R(x)^2/2\}}{N_{\mathrm{ref}}}. \tag{30}$$

Table 10 first averages each rate equally over transitions within a run, then reports the mean and sample standard deviation across runs. Its counts instead pool sample–transition pairs; a node may occur in multiple transitions, so these are not counts of independent nodes. We evaluate the strict inequalities in float64 using stable log-softmax/logsumexp and natural logarithms. No reference-correct pair lies within 10^{-12} of the zero-margin or certificate boundary; this tolerance is used only for auditing, not to relax either condition.

Observed retention. The complete certificate covers 80.86%–96.42% under the stated averaging rule. Across all datasets, 58,297 of 67,763 reference-correct sample–transition pairs satisfy both conditions, and none changes from correct to incorrect. Another 5,856 pairs remain correct without certification; all 3,610 observed correct-to-incorrect changes occur among uncertified pairs. Figure 9 plots all 66,396 reference-correct pairs with positive response margins; the remaining 1,367 pairs are excluded from the plot but remain in the coverage denominators. Every plotted point strictly below the diagonal retains its correct prediction, consistent with Theorem 1; retained points above the diagonal illustrate that the condition is sufficient rather than necessary.

Scope. Coverage is measured on the four single-graph settings using observed test-point margins and task-entry-to-task-end drift after training. It assesses a sufficient condition for individual prediction stability, distinct from full-sequence signed AF and causal attribution of algorithmic gains. The analysis is limited to these settings; an average training-support budget alone does not certify test points. Per-transition counts, derived point data, and plotting code accompany the reproducibility supplement.

B.6 Final accuracy and baseline retention

Table 11 separates final historical-task accuracy R from acquisition accuracy Q, with $\mathrm{AF} = R - Q$. The smaller forgetting of TPP-style does not arise solely from a lower acquisition reference: on whole4 it also has higher final historical accuracy than HiRP (68.28% versus 66.48%). HiRP instead achieves higher last-task accuracy (85.05% versus 72.22%), yielding higher overall FA and AA. On whole6 and global10, HiRP has both higher acquisition and higher final historical accuracy,

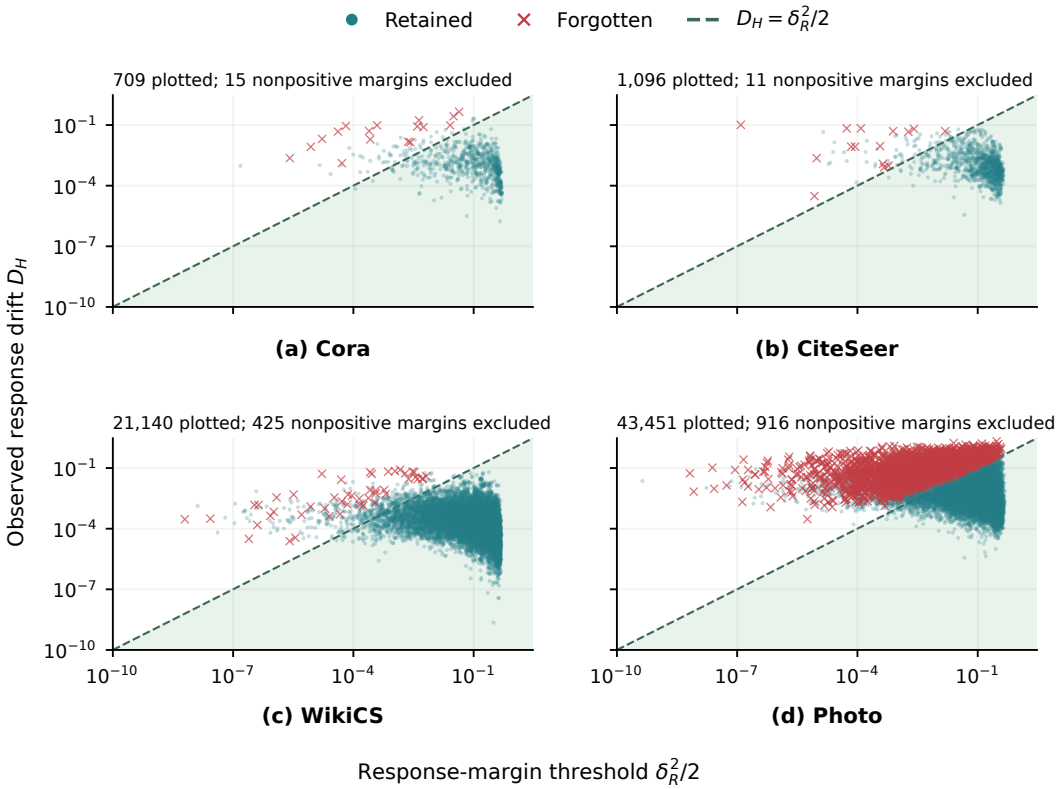


Figure 9: Single-graph coverage of Theorem 1. Each point is a reference-correct historical sample–transition pair with $\delta_R > 0$. Circles indicate retained predictions and crosses indicate correct-to-incorrect changes. The dashed line is $D_H = \delta_R^2/2$; the shaded region below it satisfies the strict certificate and contains no forgotten pairs. Panel annotations give the number of nonpositive-margin pairs excluded from the plot. Both axes use logarithmic scales.

Table 11: Historical-task retention for baseline variants and the hierarchical-response ablation. R is final historical-task accuracy and Q is accuracy when each historical task was first learned; the final task is excluded. Entries report means ± sample standard deviations in percentages or percentage points, and $\mathrm{AF} = R - Q$ before rounding. Bold marks the largest displayed mean.

Method	4 datasets / 4 tasks			6 datasets / 6 tasks			4 datasets / 10 tasks		
	$R\uparrow$	$Q\uparrow$	AF$\uparrow$	$R\uparrow$	$Q\uparrow$	AF$\uparrow$	$R\uparrow$	$Q\uparrow$	AF$\uparrow$
TPP-style	**68.28±0.34**	70.57±0.34	−2.28±0.04	70.79±0.06	72.19±0.05	−1.40±0.02	59.60±0.19	69.58±0.24	−9.98±0.09
SimpleCIL	59.83±1.15	61.59±1.19	**−1.76±0.05**	48.57±0.73	49.90±0.79	**−1.33±0.07**	50.75±0.95	58.51±0.87	**−7.77±0.09**
Hier. + radial	66.38±0.69	74.17±0.63	−7.79±0.41	71.53±0.33	76.70±0.26	−5.17±0.17	**67.65±0.16**	80.42±0.12	−12.76±0.27
HiRP (ours)	66.48±0.37	**74.91±0.57**	−8.43±0.32	**73.45±0.13**	**79.66±0.25**	−6.21±0.32	67.26±0.09	**82.00±0.11**	−14.74±0.15

alongside a larger decline from acquisition. These results characterize a learning–retention trade-off while retaining final classification performance as the primary objective.

Saved per-node predictions show that all historical correct-to-incorrect changes in the TPP-style runs coincide with changed routing; predictions remain unchanged when the selected head is unchanged. Its mean final task-routing accuracies are 98.94%, 96.46%, and 63.54% on whole4, whole6, and global10. These diagnostics characterize the routing-based TPP-style variant defined in Appendix B.2.

B.7 RETENTION AND ACQUISITION IN THE ABLATION

Let $R = (T-1)^{-1} \sum_{i<T} A_{T,i}$ be final historical-task accuracy and $Q = (T-1)^{-1} \sum_{i<T} A_{i,i}$ the corresponding acquisition accuracy. For paired configurations on the same tasks, $\Delta\mathrm{AF} = \Delta R - \Delta Q$. Table 11 reports both configurations; tangent-minus-radial differences are paired comparisons with the hierarchical response held fixed. Whole4 and whole6 improve final historical accuracy but increase acquisition accuracy by more, yielding lower AF. Global10 additionally incurs a small final-retention loss.

The mean KL budget controls the empirical support average, so changes concentrated on poorly covered historical regions can remain consequential for accuracy.

B.8 FUNCTIONAL PRESERVATION WITH IDENTICAL FEATURE SUPPORT

Table 12 compares HiRP with two distillation controls using the same historical and external feature support, alongside SGD and LwF.

Table 12: Additional distillation controls with identical feature support. Means ± sample standard deviations; FA/AA are accuracy (%) and AF is signed forgetting (points). Bold marks the highest mean.

Method	4 datasets / 4 tasks			6 datasets / 6 tasks			4 datasets / 10 tasks		
	FA ↑	AA ↑	AF ↑	FA ↑	AA ↑	AF ↑	FA ↑	AA ↑	AF ↑
SGD	75.79±0.18	58.35±0.22	−25.67±0.08	83.02±0.14	58.77±0.28	−22.70±0.10	50.16±0.15	49.96±0.15	−41.67±0.18
LwF	76.63±0.18	58.43±0.61	−25.24±0.69	84.14±0.07	61.55±0.26	−19.06±0.18	56.31±0.33	49.64±0.30	−41.23±0.27
Memory-LwF	77.13±0.17	59.37±0.63	−24.14±0.65	84.99±0.08	63.63±0.48	−16.75±0.36	54.47±0.11	47.39±0.39	−44.13±0.38
Memory-KD	78.18±0.13	68.14±0.22	−11.19±0.27	85.46±0.12	70.58±0.17	−7.63±0.19	60.71±0.12	58.65±0.42	−29.97±0.45
HiRP (ours)	**80.09±0.19**	**71.12±0.26**	**−8.43±0.32**	**87.04±0.42**	**73.22±0.23**	**−6.21±0.32**	**75.43±0.05**	**69.86±0.13**	**−14.74±0.15**

Setup. The teacher is each control's own task-entry readout $\boldsymbol{\theta}_0$, fixed within the task. Memory-LwF distills softmax probabilities over historical classes $\mathcal{O}_t$; Memory-KD uses all arrived classes $\mathcal{C}_{\leq t}$. Writing the temperature-τ distribution as $\boldsymbol{\pi}_{t,\tau}$, both minimize

$$\mathcal{L}_{\mathrm{CE}}(\mathcal{B}_t;\boldsymbol{\theta}) + \beta\tau^2 \widehat{\mathbb{E}}_{x\sim\mathcal{M}_t} D_{\mathrm{KL}}(\boldsymbol{\pi}_{t,\tau}(x;\boldsymbol{\theta}_0) \,\|\, \boldsymbol{\pi}_{t,\tau}(x;\boldsymbol{\theta}))\,, \tag{31}$$

where $\widehat{\mathbb{E}}$ averages all retained support rows with the source weights in Equation (8). The distillation term starts at task two. Both controls share HiRP's task-wise support (at most 1024 unlabeled rows), pretrained initialization, frozen features and class semantics, current training splits and batch order, three epochs, batch size 16, and label smoothing 0.2. They use ordinary SGD at 5×10^{-4} with zero momentum and weight decay, temperature $\tau = 2$, and weight $\beta = 1$. Final task readouts are evaluated with the same cumulative candidates and task-agnostic inference. The controls run on a T4 with PyTorch 2.5; detailed configurations accompany the supplemental records.

Results. Memory-KD improves over Memory-LwF and current-input LwF on all three streams. HiRP achieves higher FA, AA, and signed AF than both controls; its AA gains over Memory-KD are 2.98, 2.65, and 11.22 points on whole4, whole6, and global10, respectively. These additional comparisons support HiRP's effectiveness with historical feature access held fixed.

B.9 HYPERPARAMETER SENSITIVITY ANALYSIS

We vary the local radius $\delta \in \{0.001, 0.002, 0.004, 0.008\}$, task radius $\rho \in \{0.05, 0.1, 0.2, 0.4\}$, damping $\lambda \in \{0.003, 0.01, 0.03, 0.1\}$, and support capacity $M \in \{768, 1024, 1536, 2048\}$ on whole4, whole6, and global10. Each run changes one parameter while keeping the others at their defaults: $\delta = 0.002$, $\rho = 0.1$, $\lambda = 0.01$, and $M = 1024$. The study uses complete streams and the same three-epoch training protocol; the external-reference cap remains 512 rows throughout the capacity sweep.

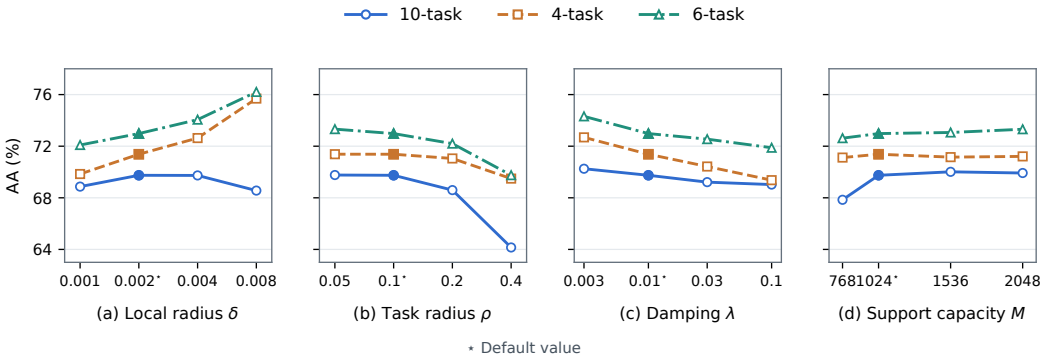


Figure 10: Sensitivity to four core parameters on three continual-learning streams. Curves show final AA; stars mark default values. The first three horizontal axes use logarithmic scales, and support capacity uses a linear scale.

Figure 10 shows that larger local steps improve AA on whole4 and whole6, whereas global10 peaks near the default local radius and declines at $\delta = 0.008$. Increasing the task radius from 0.05 to 0.4 reduces AA on all three streams, most strongly on global10, from 69.76% to 64.15%. Lower damping gives higher AA over the tested range. Increasing capacity from 768 to 1024 improves global10 AA from 67.85% to 69.74%, with limited further gains; whole4 changes little and whole6 improves modestly. These trends show that local step size, cumulative drift, and support coverage affect the streams differently.