WHEN EXPERTS ARE NOT EXPERTS: MULTIMODAL CONTINUAL INSTRUCTION TUNING WITH CERTIFIED CAPABILITIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal Continual Instruction Tuning (MCIT) aims to let Multimodal Large Language Models (MLLMs) acquire new skills after deployment without forgetting previous ones. Modern MCIT methods build on Mixture-of-Experts (MoE), adding lightweight experts and guarding them against interference. We revisit this design and find that the defenses tax plasticity without securing the expertise they protect. Learned routers barely distinguish tasks even when trained jointly, so the defenses guard a separation that never forms. Meanwhile, the difficulty of the stream concentrates in a few capabilities that a shared model fails to hold. Based on these observations, we introduce *Union-of-Capabilities* (UoC), a paradigm in which a generalist, retrained on the stored data at each refresh and frozen in between, grows only by certified capability modules. Building on UoC, we propose C2R (Certify to Route), which realizes the paradigm in two stages. (1) At training time, CERTIFY admits an expert only on verified gain over the generalist, and (2) at inference time, ROUTE dispatches a query only on certified evidence through a training-free certificate cascade, without a task label. Every installed module is trained once and then frozen, so what it has learned is never overwritten. On CoIN and UCIT, C2R sets a new state of the art, outperforming the strongest previous method with zero forgetting, fewer trainable parameters, up to 1.7× faster training at a single refresh, and 1.8× faster inference, and it keeps this lead when updated after every task, with forgetting within 0.2 points.

1 INTRODUCTION

Multimodal Large Language Models (MLLMs) (Bai et al., 2023; Liu et al., 2023; Team et al., 2023; Zhu et al., 2024) acquire broad vision-language abilities through multimodal instruction tuning. Once deployed, they face a continuous stream of new domains and instructions, and naive sequential fine-tuning on this stream suffers from *catastrophic forgetting* (French, 1999), in which the skills acquired on earlier tasks are overwritten as new ones arrive. Multimodal Continual Instruction Tuning (MCIT) aims to address this challenge by continually adapting the model to new tasks without forgetting previous skills. In real-world deployment, however, adaptation and forgetting are not the only constraints. Recent work on continual learning (CL) finds that, with storage now cheap, GPU compute has become the dominant constraint (Prabhu et al., 2023; Jeon et al., 2025; Cho et al., 2026). A practical MCIT system must therefore be *cost-effective*: it must adapt well to new tasks, preserve prior skills, and do both at low GPU cost, in training as well as in inference.

For effective adaptation, modern MCIT methods (Guo et al., 2025; Huai et al., 2025; Xie et al., 2026) have investigated the Mixture-of-Experts (MoE) paradigm (Shazeer et al., 2017), which extends the model with lightweight experts instead of retraining it. At each adapted layer, a learned router combines the experts with query-dependent weights, so every query passes through a mixture of experts rather than one. Because the experts are coupled through this mixture, training on a new task can overwrite what earlier tasks stored in them, and defenses such as constrained updates and selective freezing are added to limit this damage (Figure 1(a)). The defenses assume that *each expert holds task-level expertise*. Under this assumption, they have improved stability in the *stability-plasticity trade-off* (Grossberg, 1982), and their cost has been accepted as the price of stability. In this paper, we re-evaluate the assumption and find that *the experts do not always serve as experts* in MCIT. Pro-

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

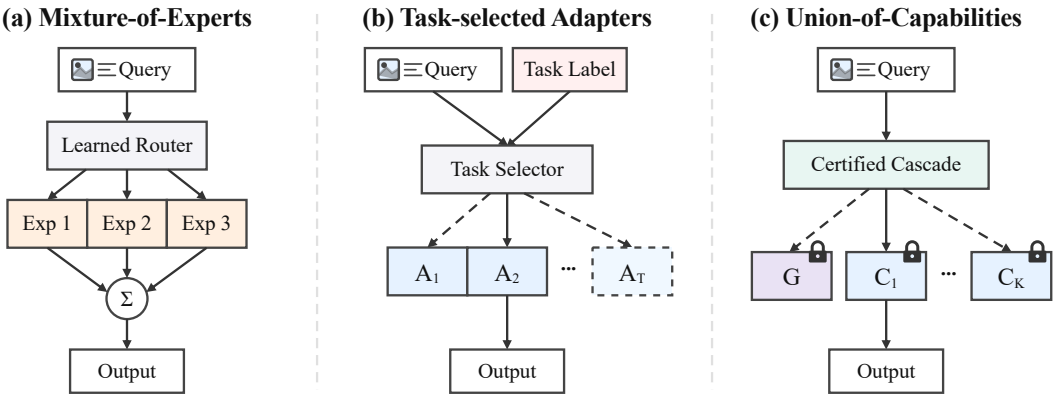

Figure 1: Three ways of organizing expertise. (a) Mixture-of-Experts with a learned router mixes all experts, so every query pays for the mixture and every update pays for defending it. (b) Task-selected adapters isolate tasks, but modules grow with the stream and selection needs task labels. (c) Union-of-Capabilities adds only certified capability modules C_k to a frozen generalist G.

tecting shared experts trades against adaptation. For example, a naive null-space projection removes most of each new task's gradient, and even more sophisticated defenses, such as the curvature-aware scaling of the strongest method in this family, remove a fixed fraction of its update no matter where in the stream the task arrives. What the defenses guard, however, never forms, since learned routers barely encode task identity even when trained on all tasks at once. Both the interference that defenses fight and the separation they fail to secure arise from sharing experts through a learned router, so these methods unavoidably pay a defense tax for a separation that does not exist.

The opposite design avoids the interference by keeping an isolated adapter per task (Figure 1(b)). Given test-time task labels, it serves as an oracle, since isolation removes forgetting. In deployment, however, queries rarely carry a task label and the task stream never ends, so the oracle must obtain a label for every query while its modules and their maintenance cost accumulate, until it becomes impractical to deploy. Neither direction is therefore cost-effective. The MoE family pays a growing defense cost for expertise that does not reliably form, and the isolation family pays a module cost that keeps growing with the stream.

Both limitations arise from organizing expertise by task. A task, however, is only a grouping of data. The knowledge it requires is better described by underlying *capabilities*, such as reading text in images, and several tasks rely on the same capability. Once expertise is organized by capability, it divides into two parts. A single general model, which we call the *generalist*, covers the capabilities that tasks share, and dedicated experts are needed only for the few capabilities it lacks. We formalize this division as *Union-of-Capabilities* (UoC), a paradigm in which expertise is *certified* rather than assumed and defended (Figure 1(c)). Its base is the frozen generalist, to which a capability module is admitted only when its measured gain over the generalist is certified. Every admitted module is trained once, then frozen, and shares no parameters with any other, so at inference each query is served by the generalist or by exactly one module. Forgetting is thereby removed for every admitted module, while the number of modules tracks certified capabilities, not the length of the stream.

Building on UoC, we propose C2R (Certify to Route), a practical MCIT system that realizes this paradigm in two stages. Following the abundant-memory regime of CL (Prabhu et al., 2023; Cho et al., 2026), tasks are stored rather than discarded after training. At each *refresh*, which is an offline pass over the stored data, C2R trains a generalist on the stored pool, profiles it to find the capabilities it lacks, and trains a candidate expert for each. Two decisions then remain, which candidates to install and which module answers each query, and C2R makes both with a certificate, a statistic computed on stored data and evaluated against a criterion fixed in advance. (1) CERTIFY certifies which candidates are installed at the refresh. It admits a candidate as an expert only on verified gain over the generalist, measured as its net effect under the deployed dispatcher, which weighs the queries correctly routed to it against those misrouted to it, and every admitted expert joins the union frozen. (2) ROUTE decides which module answers a query at inference. A training-free certificate cascade sends the query to the one installed expert whose certificate it satisfies and to the generalist

otherwise, so ROUTE learns no router, changes no module, and requires no task label. The three mechanisms thus have distinct roles. The refresh keeps the generalist current, CERTIFY covers the capabilities it lacks, and ROUTE decides which module answers. Every certificate is evaluated at the refresh, so inference applies fixed rules in one forward pass per query. Every installed expert stays frozen across refreshes, so what the system holds is decided by certificates rather than protected by defenses. A refresh proceeds while the previous system serves, adding no inference overhead, and when refreshes must be frequent, the generalist can instead be updated sequentially with no loss of accuracy (Section 5.3). Our contributions are as follows:

- We show that task-level expertise does not form in MoE-based MCIT. Learned routers barely encode task identity even when trained on all tasks at once, so the defenses guard a separation that never forms while taxing every new task's adaptation.
- We introduce Union-of-Capabilities (UoC), a paradigm that organizes expertise around certified capabilities instead of tasks, and realize it as C2R, which certifies which experts to install at the refresh and routes to them at inference without a learned router or a task label.
- C2R outperforms the strongest previous method on CoIN and UCIT by 4.24 and 10.62 points with zero forgetting, half its trainable parameters, up to 1.7× faster training, and 1.8× faster inference, and remains ahead when updated after every task, by 3.26 points under an equal replay budget, with forgetting within 0.2 points.

2 RELATED WORK

Multimodal continual instruction tuning. Early approaches to MCIT use prompts, replay, and regularization (Cao et al., 2024; Zeng et al., 2025; Chen et al., 2025; Li et al., 2026), and MoE-based methods (Shazeer et al., 2017) have recently shown strong performance. MoELoRA (Chen et al., 2024) splits a LoRA module into gated experts, and later methods protect them (Huai et al., 2025; Guo et al., 2025; Zhao et al., 2025a; Xie et al., 2026; Hou et al., 2026) with instance-level routers, stabilized routers, or expert subspaces. This family defends experts once created, whereas we measure whether the defended expertise exists and replace the defense with certification.

Expert specialization in MoE. Whether learned routing produces specialization is contested for MoE language models. A fixed hash matches a learned router (Roller et al., 2021), random experts stay competitive (Zuo et al., 2022), Mixtral shows no topic specialization (Jiang et al., 2024), and later studies diverge on how much experts specialize (Muennighoff et al., 2025; Ternovtsii & Bilak, 2026; Huang et al., 2026). They concern token-level routing in one jointly trained model, whereas we ask if *task-level* expert identity forms in adapter-based MCIT (Section 3).

Continual learning with abundant memory. GDumb (Prabhu et al., 2020) showed that retraining on stored data surpasses many CL methods, and recent studies reframe CL for abundant memory and scarce compute (Prabhu et al., 2023; Cho et al., 2026). Retraining alone, however, leaves capability gaps in MCIT (Section 3.3), for which the certificates of C2R install experts.

Parameter isolation. Separate parameters per task are standard in CL (Aljundi et al., 2017; Serrà et al., 2018), as is training modules independently and combining them (Li et al., 2022; Gururangan et al., 2022). In MCIT, ProgLoRA (Yu et al., 2025) adds a frozen LoRA block per task but fuses the blocks with a learned input-conditioned allocation, so its outputs still drift, and a recent system keeps an isolated adapter per task under hard routing (Zhao et al., 2025b), so that later tasks overwrite no adapter. C2R inherits the latter property and changes how the isolated modules are organized. Experts are indexed by certified capabilities rather than tasks, installed and reached by certificates rather than task labels or learned routers, and their count follows what the generalist lacks. Post-hoc routing over expert libraries (Muqeeth et al., 2024; Ostapenko et al., 2024; Yadav et al., 2025), built outside the continual setting, is contrasted with certified routing in Appendix G.

3 FROM TASK EXPERTS TO CERTIFIED CAPABILITIES

The defenses of the MoE-based family and their added cost rest on the assumption that *each expert holds task-level expertise*. We re-examine it with SAME (Xie et al., 2026), the strongest published member, on the CoIN stream (Chen et al., 2024) of eight tasks. We report three observations: (1) protecting shared experts taxes every new task's adaptation (Section 3.1), (2) learned routers do not hold

Table 1: A single generalist covers most of the stream. Per-task accuracy on CoIN. The generalist alone is trained *without* the data of the three underlined tasks, yet stays within 0.72 of SAME in mean accuracy. Generalist-joint and MoE-joint train the single adapter and the MoE architecture of SAME on every task's data, and the per-task oracle (Figure 1(b)) is the isolation reference.

	SciQA	TxtVQA	ImgNet	GQA	VizWiz	REC	VQAv2	OCR	Mean
SAME	78.35	60.69	90.21	61.70	54.13	59.87	66.04	63.59	66.82
Generalist alone	87.15	53.94	96.67	61.56	62.05	50.39	67.71	49.36	66.10
Generalist-joint	86.28	60.05	96.24	61.12	61.21	54.88	67.15	57.47	68.05
MoE-joint	84.67	61.88	95.81	60.40	59.29	58.48	66.63	59.00	68.27
Per-task oracle	87.57	64.52	97.33	62.94	62.58	66.06	68.66	62.36	71.50

task identity (Section 3.2), and (3) the difficulty of the stream concentrates in a few capabilities that a shared model fails to hold (Section 3.3). The paradigm of Section 3.4 follows from them. Additional analysis on the UCIT stream is given in Appendix D.

3.1 Observation 1: Protecting shared experts taxes adaptation

Because the experts are shared, a new task's gradient overlaps directions that earlier tasks rely on, so protecting them removes part of it. Figure 2 reports the fraction removed per position under two mechanisms. The first is a naive null-space projection, the standard way to block interference. The second is the curvature-aware scaling on the historical input covariance that SAME applies to its experts. The projection removes 78–85% of each new task's gradient, and SAME's scaling removes about 17%, far less but flat across positions, so the tax is structural. Consistent with this tax, SAME, trained on the same data, falls behind the per-task oracle (one isolated adapter per task, Figure 1(b)) on seven of the eight tasks (Table 1), and given the same replay budget it only matches a single adapter trained on every task's data (Section 5.3). The tax would vanish if routing assigned each task its own experts, since the protected directions would then be disjoint from the new gradient. We next ask whether routing can deliver this separation. SAME itself cannot answer this question, since it restricts router updates by design and a sequentially trained router can fail for ordering reasons.

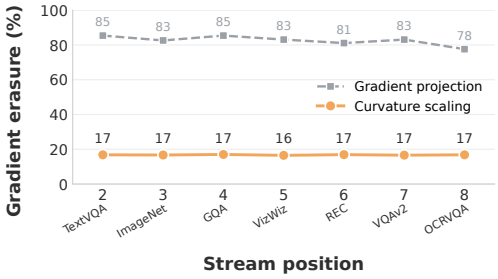

Figure 2: Protecting shared experts trades against adaptation. Fraction of each new task's gradient removed at each position of the CoIN stream by a naive null-space projection (gray, dashed) and by the curvature-aware scaling of SAME (orange, solid), a mild but structural tax.

3.2 Observation 2: Learned routers do not hold task identity

We therefore use *MoE-joint*, SAME's MoE architecture with a freely learned router, trained jointly on all eight tasks. Even in this favorable setting, routing is nearly task-agnostic, with task identity explaining almost none of the assignments at any gate (Figure 3). In the sequential setting, SAME's own diagnostics report drifting expert activations as tasks arrive (Xie et al., 2026), so the separation forms in neither regime. The cause is the training objective: a router minimizes the loss, which does not require separating tasks, and a sequentially trained router drifts as well. Both arise from *sharing experts through a learned router*, not from continual tuning. One question remains: *If expertise does not reside in the experts, where does the stream's difficulty lie?*

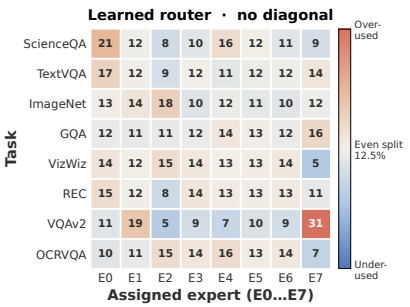

Figure 3: Learned routers do not encode task identity. Top-1 expert assignments of MoE-joint at its most task-aligned gate.

3.3 OBSERVATION 3: THE STREAM'S DIFFICULTY LIES AT THE CAPABILITY LEVEL

We hypothesize that it lies in capabilities. A capability such as reading text in images serves several tasks at once, so a task boundary groups the data rather than the knowledge, and the difficulty concentrates in the few capabilities that a shared model fails to hold. We test this by training a single adapter on all eight tasks (Generalist-joint in Table 1), the same rank-8 adapter as G but given every task's data, and comparing it with the per-task oracle. The joint adapter trails the oracle by 4.5, 11.2, and 4.9 points on TextVQA, REC, and OCRVQA, while staying within 1.9 of it on the other five. The certification sets show the same three gaps, 4.3, 10.4, and 7.8 points against at most 1.2 elsewhere (Appendix C). TextVQA and OCRVQA, which rely on the same reading-text capability, fall together although they are separate tasks. Sharing through experts does not close these gaps, since MoE-joint of Section 3.2, with twice the parameters of the C2R library, also trails the oracle on all three tasks, so shared training fails to hold them while isolated training does. The rest of the stream needs no dedicated expert. The generalist alone, the same adapter trained without the data of these three tasks (G in Section 4), matches or exceeds SAME on five of the eight tasks (GQA within 0.14) and falls short only on the three tasks whose data it was denied. Taken together, the three observations demonstrate that expertise in MCIT (1) is not secured by defended experts, (2) is not recovered by learned routing, and (3) is required only for a few capabilities that a shared model fails to hold, even at twice the parameters.

3.4 THE UNION-OF-CAPABILITIES PARADIGM

The three observations lead to three design principles, each answering one observation. (1) *Modules are frozen rather than defended.* Defenses run at every update to protect expertise that does not form, whereas a frozen module pays its cost once. (2) *Routing is certified rather than learned.* A learned router is not trained to recover task structure and drifts as tasks arrive, whereas measured criteria cannot drift, and their error rate is checked against a level declared in advance. (3) *Modules are indexed by capability rather than by task.* The generalist can cover most of the stream, and the remaining difficulty is a small set of capabilities it lacks. Based on these principles, we introduce *Union-of-Capabilities* (UoC, Figure 1(c)). A UoC system has the form $S = G \oplus C_1 \oplus \cdots \oplus C_K$, where G is a generalist frozen between refreshes over the frozen backbone, C_k ($k = 1, \ldots, K$) is a frozen capability module, and $\oplus$ composes modules that share no parameters. A module is admitted to the union only when its gain over G is certified, so K tracks the number of certified capability gaps rather than the length of the stream. On our streams the certified groups mostly coincide with tasks; the capability index matters where they do not, as with reading text. Since the admitted modules are frozen and share no parameters, the answer a module gives to any query is invariant to all subsequent stream content. The structure does not fix which queries reach each module or how the generalist behaves after refreshes; these are measured (Section 5.3 and Appendix E).

4 C2R: CERTIFY TO ROUTE

C2R turns principles (1)–(3) into an MCIT system with two stages (Figure 4). CERTIFY runs during an offline training pass that we call a *refresh* and decides which candidate experts join the system, and ROUTE runs at inference and decides which module answers a query. Both decisions are made by certificates. Every arriving task t is stored and divided into a training set $\mathcal{D}_t^{\text{train}}$ and a held-out *certification set* $\mathcal{D}_t^{\text{cert}}$. A *certificate* grants a privilege, installing an expert or routing a query to it, if and only if a statistic measured on the certification sets under the deployed dispatcher meets a criterion fixed in advance; it is not a guarantee. The two certificates are coupled: the install certificate measures an expert on exactly the queries that the route certificate sends to it.

A refresh incorporates every task that has arrived since the previous one and builds the system $S = G \oplus \{C_k\}_{k \in \mathcal{I}}$ of Section 3.4 ($K = |\mathcal{I}|$), where G is the generalist, $\mathcal{I}$ the set of installed groups, and C_k the module serving installed group k. A group is the unit of storage and certification, one per task; a module is what a certificate installs and may serve several groups sharing a capability or a certified slice of one (Appendix B). The refresh first trains G on a curated subset of the stored pool $\mathcal{D} = \bigcup_t \mathcal{D}_t^{\text{train}}$. It then screens the certification profile of G for capability gaps and trains a candidate expert per flagged group. Each candidate's route is fixed from stored data, and the install certificate decides whether its group joins $\mathcal{I}$. An installed module is kept frozen. Every step reads the unordered

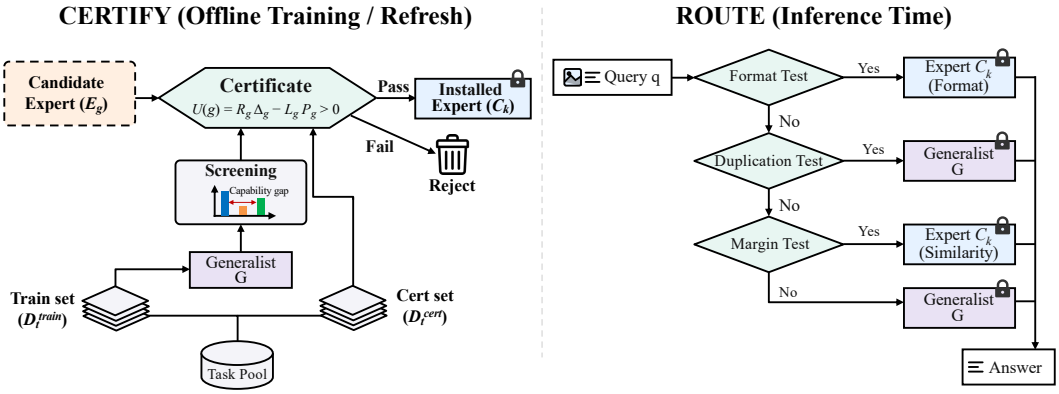

Figure 4: Overview of C2R. Left: CERTIFY (training time; *refresh*) screens G for capability gaps, trains a candidate E_g per gap, and installs it as a frozen module C_k only if it passes the certificate criterion $U(g) > 0$. Right: ROUTE (inference time) sends a query through a training-free cascade of certified tests (format, duplication, or similarity margin) to exactly one frozen module, the generalist G when no test passes.

pool, so the instance built from a pool is invariant to arrival order, and the final generalist is the same under any number of full-pool refreshes, since the last refresh reads the full pool (Appendix E).

4.1 CERTIFY: CERTIFIED EXPERTS

CERTIFY separates proposal from admission. A screen proposes candidate groups, and only the certificate admits an expert. Let $v(g) = \mathrm{acc}(G; \mathcal{D}_g^{\mathrm{cert}})$ be the accuracy of G on the certification set of group g, and v the profile over groups. The screen $\mathcal{F} = \sigma_{\mathrm{gap}}(v; \Delta_{\min})$ flags the groups below the median of v that are separated from the rest by the largest gap, provided the gap is at least a floor $\Delta_{\min}$ fixed in advance; if the groups use different metrics, it flags those whose candidates gain more than a floor δ over G (Appendix B). A group declared in advance must pass the same screen. A candidate trained on the pooled data of flagged groups sharing a capability is certified per group and allocated as in Appendix B. For every flagged group $g \in \mathcal{F}$, a candidate expert E_g is trained on the stored data of its capability and installed as its module C_k, so that g joins $\mathcal{I}$, if and only if

$$U(g) \;=\; \underbrace{R_g\,\Delta_g}_{\text{recall-weighted gain}} \;-\; \underbrace{L_g\,P_g}_{\text{leak-weighted damage}} \;>\; 0, \tag{1}$$

where all four terms are measured on the certification sets, with q drawn from their union and $q \in g$ denoting the queries of group g. Here, $R_g = \Pr[\pi(q) = E_g \mid q \in g]$ is the *recall* of the dispatcher π of Section 4.2 on the group, with the candidate's route in place, and $\Delta_g = \mathbb{E}[\mathrm{acc}(E_g; q) - \mathrm{acc}(G; q) \mid \pi(q) = E_g,\, q \in g]$ is the *gain* of E_g over G on the routed queries. $L_g = \Pr[\pi(q) = E_g \mid q \notin g]$ is the rate at which the other groups' queries *leak* to E_g, and $P_g = \mathbb{E}[\mathrm{acc}(G; q) - \mathrm{acc}(E_g; q) \mid \pi(q) = E_g,\, q \notin g]$ is the *penalty* those queries suffer, defined as 0 when no query leaks. Weighted by the group prior $p_g = \Pr[q \in g]$, $p_g R_g \Delta_g - (1 - p_g) L_g P_g$ is exactly the change in the expected accuracy of S from installing g when its routed queries would otherwise go to G. An expert is thus admitted by its net effect under the route that actually reaches it, not by its accuracy in isolation. The certificate omits p_g, since the deployed query mix is unknown in advance; the p_g-weighted criterion installs the same experts (Appendix C). If the certificate passes, E_g is frozen and becomes the installed module C_k of its group, with k its index in $\mathcal{I}$; otherwise the candidate is discarded and its group stays with G.

4.2 ROUTE: CERTIFIED ROUTES

ROUTE decides at inference which module answers a query. It trains no router, and every rule it applies is fixed at the refresh. Installed groups are of two kinds. A *format-defined* group $k \in \mathcal{I}_{\mathrm{fmt}}$ is identified by a deterministic *format test* $c_k^{\mathrm{fmt}}(q) \in \{0, 1\}$ with a verbatim substring extracted from its stored prompts and deployed only when its frequencies pass thresholds fixed in advance.

A *similarity-gated* group $k \in \mathcal{I}_{\text{sim}}$ is identified by a *margin* $m_k(q)$, the difference between the similarity of q to the stored data of k and its similarity to the pool of G, under a similarity measure fixed in advance (Appendix B). At the deployed threshold $\tau_k = 0$, q passes when it is nearer to k than to G. Its route certificate requires the leak of the group, measured on the certification sets, to stay below a level α declared in advance, so the error budget of a route is declared rather than tuned. A format-defined group also carries a margin test that catches a query whose prompt escapes its pattern, with τ_k set at the same leak level. A *duplication test* $c^{\text{dup}}(q) \in \{0, 1\}$ returns 1 when the question text of q recurs verbatim in the stored questions of G; such a query is in-distribution for G by construction and returns to it before any margin test can claim it. The dispatcher returns the module that answers, evaluating three stages in a fixed order,

$$\pi(q) \;=\; \begin{cases} C_k, & c_k^{\text{fmt}}(q) = 1 \text{ for some } k \in \mathcal{I}_{\text{fmt}} \quad \text{(format stage)} \\ G, & \text{else if } c^{\text{dup}}(q) = 1 \quad \text{(duplication stage)} \\ C_k, & \text{else if } m_k(q) > \tau_k \text{ for some } k \in \mathcal{I} \quad \text{(margin stage)} \\ G, & \text{otherwise,} \end{cases} \tag{2}$$

where, within a stage, the first satisfied test in a fixed order routes. A query thus reaches an installed module only under the certificate that defines its group.

4.3 Training and inference

The inference loop answers each query with the current instance by Equation (2), in one forward pass through G or one expert. Because every task is stored, how much of the pool G revisits at a refresh is a compute choice. By default a refresh retrains G on the full pool and reuses every installed expert, so the cost of a stream grows with the number of refreshes and, when refreshes are frequent, can exceed that of the sequential baselines (Section 5.3). In this regime, C2R updates the previous G sequentially instead of retraining it (denoted as C2R-seq). Each refresh trains G on the newly arrived tasks together with a replay budget of stored examples, one per new example, taken in equal shares from the earlier tasks G serves, and the accuracy of G on its earlier tasks is measured on the certification sets at each refresh. Frequent refreshes therefore sacrifice no accuracy, since C2R-seq matches the single-refresh instance under every schedule (Section 5.3). Algorithms 1 and 2 in Appendix A provide the pseudocode, and Appendix B the details.

5 Experiments

5.1 Experimental setup

Benchmarks and backbone. We evaluate on CoIN (Chen et al., 2024), a stream of eight tasks arriving in the order ScienceQA, TextVQA, ImageNet, GQA, VizWiz, REC, VQAv2, and OCRVQA. We also evaluate on UCIT (Guo et al., 2025), a six-task stream arriving in the order ImageNet-R, ArxivQA, VizWiz-Cap, IconQA, CLEVR, and Flickr30k. All methods use LLaVA-1.5-7B (Liu et al., 2024) as the backbone, following prior work.

Baselines. On both benchmarks, we compare against MoELoRA (Chen et al., 2024), Modal-Prompt (Zeng et al., 2025), HiDe-LLaVA (Guo et al., 2025), and SAME (Xie et al., 2026). For CoIN, we further compare against Continual-LLaVA (Cao et al., 2024), SEFE (Chen et al., 2025), ProgLoRA (Yu et al., 2025), and LLaVA-CMoE (Zhao et al., 2025a). For UCIT, we include the zero-shot backbone, FT-LoRA, Replay-LoRA (Guo et al., 2025), and CL-MoE (Huai et al., 2025). The baselines follow their original sequential protocols, with accuracies taken from their papers; we also rerun SAME, reproducing its published mean, and MoELoRA for the cost and replay comparisons (Section 5.3, Appendix E). For C2R, the main tables report a single refresh at the end of the stream.

Hardware and evaluation metrics. All experiments run on a single NVIDIA RTX 3090 GPU, with costs measured in wall-clock time. We report the final per-task accuracy, the mean accuracy across tasks (Tables 2 and 3), forgetting, and routing accuracy. Appendix B defines the metrics and the setup, and Appendix E adds results on forgetting, task ordering, and seeds.

Table 2: Results on CoIN: final per-task and mean accuracy after the eight-task stream. Best in **bold**.

Method	SciQA	TxtVQA	ImgNet	GQA	VizWiz	REC	VQAv2	OCR	Mean
MoELoRA	62.02	52.05	37.21	53.12	43.32	33.22	57.92	65.75	50.58
Continual-LLaVA	58.67	49.99	57.66	**62.53**	42.32	16.25	64.33	**74.91**	53.33
ModalPrompt	68.42	56.40	41.13	61.11	50.13	36.69	66.90	59.68	55.06
SEFE	75.35	58.66	83.10	54.25	48.85	16.75	65.35	66.25	58.57
ProgLoRA	74.84	51.83	83.90	49.93	53.87	31.19	62.71	64.44	59.09
LLaVA-CMoE	77.55	58.17	94.50	48.91	55.45	23.40	56.40	59.44	59.23
HiDe-LLaVA	73.20	56.92	69.28	61.33	50.76	59.18	67.12	64.76	63.95
SAME	78.35	60.69	90.21	61.70	54.13	59.87	66.04	63.59	66.82
C2R (ours)	**86.75**	**64.24**	**96.67**	61.56	**61.75**	**64.39**	**67.61**	65.49	**71.06**

Table 3: Results on UCIT: final per-task and mean accuracy after the six-task stream. Best in **bold**.

Method	ImageNet-R	ArxivQA	VizWiz-Cap	IconQA	CLEVR	Flickr30k	Mean
Zero-shot	18.88	52.62	38.75	21.25	21.12	41.44	32.34
FT-LoRA	29.33	55.30	45.51	26.13	13.07	58.07	37.90
MoELoRA	49.87	77.63	43.65	46.40	36.47	58.34	52.06
Replay-LoRA	76.93	87.07	54.31	56.43	36.40	55.94	61.18
CL-MoE	64.12	78.38	44.83	62.00	50.75	58.06	59.69
HiDe-LLaVA	80.50	89.83	48.78	62.90	47.97	55.15	64.19
ModalPrompt	81.50	90.62	50.17	62.34	51.41	57.09	65.52
SAME	83.83	91.40	51.33	65.27	53.50	57.43	67.12
C2R (ours)	**90.07**	**93.13**	**61.10**	**85.50**	**77.83**	**58.82**	**77.74**

5.2 Main results

C2R outperforms every previous method on both benchmarks (Tables 2 and 3), exceeding the strongest, SAME, by 4.24 points on CoIN and 10.62 on UCIT in mean accuracy; under the sequential regime with equal replay it still leads SAME by 3.26 on CoIN (Section 5.3). It is above SAME on thirteen of the fourteen columns and within 0.14 points on GQA. The installed experts serve TextVQA, REC, and OCRVQA on CoIN, declared from the certification-set profile and then certified, and IconQA and CLEVR on UCIT, selected by the screen alone, which also recovers the three CoIN groups without the declaration (Appendix C). The sources of the margin differ by stream. On UCIT a single generalist retrained on the stored pool already leads SAME by 9.17 points and MoE-joint by 1.94, and the two certified experts recover 1.45 of the remaining 1.65 points to the oracle (Appendix E). On CoIN, G alone is within 0.72 of SAME and the certified experts carry the margin, which SAME does not close even with the same replay budget (Section 5.3). The row of C2R is identical under every task order, and its forgetting under the metric of Xie et al. (2026) is 0.00 on every task, since expert-served tasks are frozen and G is rebuilt from the full pool (Appendix E).

5.3 Analysis

Refresh schedule. Table 4 reports the training cost and final mean of C2R on CoIN against the number of refreshes. The final instance is the same under every schedule (Section 4.3), and the cost, certification included, grows with the number of refreshes, to 236 GPU-hours when refreshed after every task, above SAME (228.0). C2R-seq, which updates G sequentially, reaches a final mean of 71.47, 71.36, and 71.42 for 2, 4, and 8 refreshes, at most 0.41 points above the single-refresh instance (71.06) and 4.54 to 4.65 points above SAME (66.82), with forgetting within 0.2 points under every schedule. Refreshed after every task, the regime of the sequential baselines, it still leads SAME by 4.60 points, and it leads SAME given the same replay budget (Appendix E) by 3.26 points at lower cost.

Table 4: Cost and final mean on CoIN against the number of refreshes. C2R retrains G at each refresh, and C2R-seq updates it sequentially with a replay budget equal to the new data.

Refreshes	1	2	4	8
C2R, GPU-h	133.81	166.64	210.56	235.62
C2R, mean	71.06	71.06	71.06	71.06
C2R-seq, GPU-h	133.81	152.15	162.58	180.76
C2R-seq, mean	71.06	71.47	71.36	71.42

Design validation. Table 5 compares MCIT systems under diverse regimes, every Δ measured against G alone (66.10). Adding the merged expert deltas to G in weight space lowers the mean by 3.22 points (uniform) and 1.73 (TIES (Yadav et al., 2023)), so the modules must stay disjoint and routed. G alone is within 0.72 of SAME, and SAME given the same replay budget matches the single adapter given every task's data, so neither the data regime nor the MoE defenses explain the margin; on UCIT that budget lifts SAME only to 68.85 (Appendix E). The certified experts add $+4.96$, within 0.44 points of the per-task oracle.

Table 5: Design validation on CoIN. The per-task oracle is granted test-time task labels.

MCIT system	Mean	Δ vs G
Merged library (uniform)	62.88	-3.22
Merged library (TIES)	64.37	-1.73
Generalist G alone	66.10	0.00
SAME	66.82	$+0.72$
Generalist-joint	68.05	$+1.95$
MoE-joint	68.27	$+2.17$
C2R	**71.06**	$+$**4.96**
Per-task oracle	71.50	$+5.40$

Routing effectiveness. ROUTE reads only the query, with no learned router and no test-time task label. Routing accuracy is the share of queries that reach their designated module (Table 6). Our certified cascade reaches 97.24% on CoIN and 98.68% on UCIT and beats the training-free alternatives, nearest-class-mean (NCM) classifiers and weight-space routing (Ostapenko et al., 2024), so certification makes the installed expertise reachable without a task label. Under paraphrased instructions that defeat every format pattern, the margin tests of the format-defined groups hold routing at 90.3% (64.2% without them) and the mean at 69.6, above SAME on the original prompts (Appendix C).

Table 6: Routing accuracy on CoIN against training-free alternatives and the task-label oracle.

Routing family	Accuracy (%)
Certified cascade (ours)	**97.24**
NCM prototype	67.34
NCM task-ID (8-way)	64.02
Weight-space	9.91
Task-label oracle	100.00

Stage contributions. Removing either stage returns the system to G alone, and label-routed experts are the oracle, not a deployable system. Table 7 therefore places C2R between G alone and the oracle on three axes: the mean, routing, and the three installed tasks, where the experts recover 97%, 89%, and 124% of the oracle's gain over G. On OCRVQA, C2R exceeds the per-task oracle because two adapters serve the task under a question-type test, an allocation by capability that one per-task adapter cannot express (Appendix C). On UCIT, G alone leaves 1.65 points to the oracle and C2R recovers 1.45 (Appendix E).

Table 7: C2R against G alone and the task-label oracle on CoIN.

Axis	G alone	Oracle	C2R
Mean accuracy	66.10	71.50	71.06
Routing accuracy (%)	–	100.00	97.24
TextVQA	53.94	64.52	64.24
REC	50.39	66.06	64.39
OCRVQA	49.36	62.36	65.49

Resource efficiency. Table 8 compares GPU costs. One refresh of C2R takes 133.8 GPU-hours including certification, or 170.4 with the screen alone (Appendix C), against 228.0 for SAME ($1.70\times$ and $1.34\times$), as the pool caps the examples per task (Appendix B) and no router or defense is trained. Inference is $1.81\times$ faster than SAME, since one adapter serves each query.

Table 8: GPU cost on CoIN. Best in **bold**.

MCIT system	Params (M)	GPU-h	Inference (min)
SAME	92.8	228.0	35.0
Per-task oracle	92.8	165.0	19.7
C2R	**46.4**	**133.8**	**19.3**

6 CONCLUSION

This paper shows that task-level expertise does not form in MoE-based MCIT and that the defenses built on this assumption tax every new task to guard a separation that never forms. Union-of-Capabilities (UoC) replaces the assumption with a certificate. A frozen generalist grows only by certified capability modules, and every route passes a test fixed in advance. C2R (Certify to Route) implements the paradigm in two stages, certifying experts at the refresh and routing to them at inference. C2R outperforms the strongest previous method on CoIN and UCIT with zero forgetting, fewer parameters, and lower GPU cost, and keeps its lead even when updated after every task. Expertise is thus better certified than defended. Limitations are discussed in Appendix F.

AI USE STATEMENT

In this work, we used generative AI tools for writing and editing assistance. We have not used generative AI tools for generating research ideas, designing the method, producing experimental results, or analyzing them. All AI-assisted text was reviewed by the authors, and every reported number was produced by our own measurement pipeline and checked against the source logs by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

We acknowledge and concur with the ICLR Code of Ethics. This work uses only publicly released benchmarks (CoIN and UCIT) under their original licenses and involves no human subjects and no new data collection. VizWiz contains photographs taken by blind users, and we use it strictly within its published terms without redistribution.

REPRODUCIBILITY STATEMENT

Appendix A gives the full training and inference procedures, and Appendix B specifies the architecture, the training recipe, the train/certification split, the evaluation metrics, and all declared constants, which are fixed in advance and never tuned per task or per benchmark. No certificate reads test data. Upon acceptance, we will release the implementation and the certification logs.

REFERENCES

Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *CVPR*, 2017.

Anastasios N. Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 2023.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv:2308.12966*, 2023.

Meng Cao, Yuyang Liu, Yingfei Liu, Tiancai Wang, Jiahua Dong, Henghui Ding, Xiangyu Zhang, Ian Reid, and Xiaodan Liang. Continual LLaVA: Continual instruction tuning in large vision-language models. *arXiv:2411.02564*, 2024.

Cheng Chen, Junchen Zhu, Xu Luo, Heng T Shen, Jingkuan Song, and Lianli Gao. CoIN: A benchmark of continual instruction tuning for multimodel large language models. In *NeurIPS*, 2024.

Jinpeng Chen, Runmin Cong, Yuzhi Zhao, Hongzheng Yang, Guangneng Hu, Horace Ho Shing Ip, and Sam Kwong. SEFE: Superficial and essential forgetting eliminator for multimodal continual instruction tuning. In *ICML*, 2025.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality. `https://lmsys.org/blog/2023-03-30-vicuna/`, 2023.

Dongkyu Cho, Taesup Moon, Rumi Chunara, Kyunghyun Cho, and Sungmin Cha. Forget forgetting: Continual learning in a world of abundant memory. In *ICLR*, 2026.

Jeremy Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. In *ICML*, 2019.

Adam Dahlgren Lindström and Savitha Sam Abraham. CLEVR-Math: A dataset for compositional language, visual and mathematical reasoning. In *International Workshop on Neural-Symbolic Learning and Reasoning (NeSy)*, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

William Fleshman and Benjamin Van Durme. SpectR: Dynamically composing LM experts with spectral routing. In *COLM*, 2025.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 1999.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017.

Stephen Grossberg. *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control*. D. Reidel Publishing Company, 1982.

Haiyang Guo, Fanhu Zeng, Ziwei Xiang, Fei Zhu, Da-Han Wang, Xu-Yao Zhang, and Cheng-Lin Liu. HiDe-LLaVA: Hierarchical decoupling for continual instruction tuning of multimodal large language model. In *ACL*, 2025.

Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. VizWiz grand challenge: Answering visual questions from blind people. In *CVPR*, 2018.

Danna Gurari, Yinan Zhao, Meng Zhang, and Nilavra Bhattacharya. Captioning images taken by people who are blind. In *ECCV*, 2020.

Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A. Smith, and Luke Zettlemoyer. DEMix layers: Disentangling domains for modular language modeling. In *NAACL*, 2022.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, 2021.

ZhiYan Hou, Haiyun Guo, Haokai Ma, Yandu Sun, Yonghui Yang, and Jinqiao Wang. PASs-MoE: Mitigating misaligned co-drift among router and experts via pathway activation subspaces for continual learning. In *ACL*, 2026.

Tianyu Huai, Jie Zhou, Xingjiao Wu, Qin Chen, Qingchun Bai, Ze Zhou, and Liang He. CL-MoE: Enhancing multimodal large language model with dual momentum mixture-of-experts for continual visual question answering. In *CVPR*, 2025.

Minbin Huang, Han Shi, Chuanyang Zheng, Yimeng Wu, Guoxuan Chen, Xintong Yu, Yichun Yin, and Hong Cheng. UniPool: A globally shared expert pool for mixture-of-experts. *arXiv:2605.06665*, 2026.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, 2019.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *ICLR*, 2023.

Sungho Jeon, Xinyue Ma, Kwang In Kim, and Myeongjae Jeon. REP: Resource-efficient prompting for rehearsal-free continual learning. In *NeurIPS*, 2025.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *arXiv:2401.04088*, 2024.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014.

Lei Li, Yuqi Wang, Runxin Xu, Peiyi Wang, Xiachong Feng, Lingpeng Kong, and Qi Liu. Multimodal ArXiv: A dataset for improving scientific comprehension of large vision-language models. In *ACL*, 2024.

Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. Branch-Train-Merge: Embarrassingly parallel training of expert language models. *arXiv:2208.03306*, 2022.

Songze Li, Mingyu Gao, Tonghua Su, Xu-Yao Zhang, and Zhongjie Wang. Multimodal continual instruction tuning with dynamic gradient guidance. In *CVPR*, 2026.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024.

Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. IconQA: A new benchmark for abstract diagram understanding and visual language reasoning. In *NeurIPS Datasets and Benchmarks Track*, 2021.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *NeurIPS*, 2022.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. OCR-VQA: Visual question answering by reading text in images. In *ICDAR*, 2019.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, Ali Farhadi, Noah A. Smith, Pang Wei Koh, Amanpreet Singh, and Hannaneh Hajishirzi. OLMoE: Open mixture-of-experts language models. In *ICLR*, 2025.

Mohammed Muqeeth, Haokun Liu, Yufan Liu, and Colin Raffel. Learning to route among specialized experts for zero-shot generalization. In *ICML*, 2024.

Oleksiy Ostapenko, Zhan Su, Edoardo Maria Ponti, Laurent Charlin, Nicolas Le Roux, Matheus Pereira, Lucas Caccia, and Alessandro Sordoni. Towards modular LLMs by building and reusing a library of LoRAs. In *ICML*, 2024.

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In *ECML*, 2002.

Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015.

Ameya Prabhu, Philip Torr, and Puneet Dokania. GDumb: A simple approach that questions our progress in continual learning. In *ECCV*, 2020.

Ameya Prabhu, Hasan Abed Al Kader Hammoud, Puneet Dokania, Philip HS Torr, Ser-Nam Lim, Bernard Ghanem, and Adel Bibi. Computationally budgeted continual learning: What does matter? In *CVPR*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *CVPR*, 2017.

Stephen Roller, Sainbayar Sukhbaatar, Arthur Szlam, and Jason Weston. Hash layers for large sparse models. In *NeurIPS*, 2021.

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *ICLR*, 2021.

Joan Serrà, Dídac Surís, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *ICML*, 2018.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR*, 2017.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *CVPR*, 2019.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv:2312.11805*, 2023.

Ivan Ternovtsii and Yurii Bilak. Equifinality in mixture of experts: Routing topology does not determine language modeling quality. *arXiv:2604.14419*, 2026.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance for continual learning. In *CVPR*, 2021.

Zhen-Hao Xie, Jun-Tao Tang, Yu-Cheng Shi, Han-Jia Ye, De-Chuan Zhan, and Da-Wei Zhou. SAME: Stabilized mixture-of-experts for multimodal continual instruction tuning. In *ICML*, 2026.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *NeurIPS*, 2023.

Prateek Yadav, Colin Raffel, Mohammed Muqeeth, Lucas Caccia, Haokun Liu, Tianlong Chen, Mohit Bansal, Leshem Choshen, and Alessandro Sordoni. A survey on model MoErging: Recycling and routing among specialized experts for collaborative learning. *TMLR*, 2025.

Yahan Yu, Duzhen Zhang, Yong Ren, Xuanle Zhao, Xiuyi Chen, and Chenhui Chu. Progressive LoRA for multimodal continual instruction tuning. In *Findings of ACL*, 2025.

Fanhu Zeng, Fei Zhu, Haiyang Guo, Xu-Yao Zhang, and Cheng-Lin Liu. ModalPrompt: Towards efficient multimodal continual instruction tuning with dual-modality guided prompt. In *EMNLP*, 2025.

Hengyuan Zhao, Ziqin Wang, Qixin Sun, Kaiyou Song, Yilin Li, Xiaolin Hu, Qingpei Guo, and Si Liu. LLaVA-CMoE: Towards continual mixture of experts for large vision-language models. *arXiv:2503.21227*, 2025a.

Hongbo Zhao, Fei Zhu, Haiyang Guo, Meng Wang, Rundong Wang, Gaofeng Meng, and Zhaoxiang Zhang. MLLM-CL: Continual learning for multimodal large language models. *arXiv:2506.05453*, 2025b.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. In *ICLR*, 2024.

Simiao Zuo, Xiaodong Liu, Jian Jiao, Young Jin Kim, Hany Hassan, Ruofei Zhang, Tuo Zhao, and Jianfeng Gao. Taming sparsely activated transformer with stochastic experts. In *ICLR*, 2022.

A Algorithm Details

Algorithms 1 and 2 summarize C2R. The refresh loop trains the generalist, screens its certification profile, trains a candidate expert for every flagged group, and installs the candidates that pass the install certificate. The inference loop applies the fixed cascade and serves each query with exactly one frozen module. Both loops read only the declared constants listed as the input of Algorithm 1, and the values used in our experiments are given in Appendix B. If the groups of a stream use different metrics, line 3 follows candidate training and flags those gaining more than δ over G (Appendix B). The pooled candidate for groups that share a capability, the question-type slices of a group, and the re-certification of previously installed routes follow the rules of Appendix B.

Algorithm 1 Refresh (CERTIFY)

Input: frozen backbone f_0, stored pool $\mathcal{D} = \bigcup_t \mathcal{D}_t^{\text{train}}$, certification sets $\{\mathcal{D}_t^{\text{cert}}\}$, declared constants κ, $\Delta_{\min}$, δ, pattern thresholds, α, and the level and margin of the slice test
Output: serving instance $S = G \oplus \{C_k\}_{k\in\mathcal{I}}$, format patterns, thresholds $\{\tau_k\}$

1: $G \leftarrow$ train from f_0 on the curated pool $\kappa(\mathcal{D})$, then freeze ▷ C2R-seq continues the previous G
2: $v(g) \leftarrow \text{acc}(G; \mathcal{D}_g^{\text{cert}})$ for every group g
3: $\mathcal{F} \leftarrow \sigma_{\text{gap}}(v; \Delta_{\min})$ ▷ a declared group must pass the same screen
4: re-check the route certificate (leak $\le \alpha$) of every installed $k \in \mathcal{I}_{\text{sim}}$, returning k to G if it fails
5: **for** each $g \in \mathcal{F}$, in a fixed order **do**
6: $E_g \leftarrow$ train on the stored data of g ▷ groups sharing a capability also get a pooled candidate
7: extract a format pattern for g from its stored prompts
8: **if** the pattern passes the pattern thresholds **then** g is format-defined with test c_g^{fmt}
9: **else** g is similarity-gated with $\tau_g \leftarrow 0$, and its leak must be $\le \alpha$
10: **end if**
11: for a format-defined g, $\tau_g \leftarrow$ the margin threshold with leak $\le \alpha$
12: measure R_g, Δ_g, L_g, and P_g of each candidate under the dispatcher π
13: $U(g) \leftarrow R_g\Delta_g - L_gP_g$ for each candidate
14: **if** $U(g) > 0$ for the designated candidate of g **then** install it as C_k, freeze it, $\mathcal{I} \leftarrow \mathcal{I} \cup \{g\}$
15: a pooled candidate takes a slice of g only if it beats C_k there under the slice test
16: **else** leave g served by G
17: **end if**
18: **end for**
19: **return** $S = G \oplus \{C_k\}_{k\in\mathcal{I}}$

Algorithm 2 Inference (ROUTE)

Input: query q, serving instance G, $\{C_k\}_{k\in\mathcal{I}}$, format patterns, thresholds $\{\tau_k\}$
Output: answer

if $c_k^{\text{fmt}}(q) = 1$ for some $k \in \mathcal{I}_{\text{fmt}}$ **then** $\pi(q) \leftarrow C_k$ ▷ format stage
else if $c^{\text{dup}}(q) = 1$ **then** $\pi(q) \leftarrow G$ ▷ duplication stage
else if $m_k(q) > \tau_k$ for some $k \in \mathcal{I}$, the first in a fixed order **then** $\pi(q) \leftarrow C_k$ ▷ margin stage
else $\pi(q) \leftarrow G$
end if
$M \leftarrow \pi(q)$
return $M(q)$ ▷ one forward pass

B Implementation Details

Datasets and backbone. CoIN consists of ScienceQA (Lu et al., 2022), TextVQA (Singh et al., 2019), ImageNet (Deng et al., 2009), GQA (Hudson & Manning, 2019), VizWiz (Gurari et al., 2018), REC (referring expression comprehension) (Kazemzadeh et al., 2014), VQAv2 (Goyal et al., 2017), and OCRVQA (Mishra et al., 2019), and UCIT of ImageNet-R (Hendrycks et al., 2021), ArxivQA (Li et al., 2024), VizWiz-Cap (Gurari et al., 2020), IconQA (Lu et al., 2021), CLEVR (Dahlgren Lindström & Abraham, 2022), and Flickr30k (Plummer et al., 2015). CoIN covers science questions, text reading, image classification, compositional questions, questions from

blind users, referring expressions, general questions, and questions about book covers. UCIT covers renditions of ImageNet classes, figure questions from arXiv papers, captions for blind users, icon questions, arithmetic on synthetic scenes, and image descriptions. LLaVA-1.5-7B pairs a CLIP ViT-L/14-336 vision encoder (Radford et al., 2021) with a Vicuna-7B language model (Chiang et al., 2023) and is the backbone of every method.

Adapters and training. The generalist and the experts are LoRA adapters of rank 8 with $\alpha_{\mathrm{LoRA}} = 16$, applied to the FFN gate, up, and down projections only. Each adapter, whether the generalist or a candidate expert, is trained for one epoch over its own training data with learning rate 2×10^{-4}, a cosine schedule, and a warmup ratio of 0.03. All training hyperparameters are selected on a random 20% sample of the UCIT training data and then fixed across tasks and benchmarks, so no training hyperparameter is tuned per task or per benchmark.

Certification sets and curation. Certification sets are held out when a task arrives and are never used for training. On UCIT each task's training data is split 90/10, a split declared in advance, which leaves 959 to 1,600 certification examples per task. On CoIN 4,000 examples per task are held out (2,500 for ScienceQA, limited by its smaller training split). The curation rule κ is fixed before any certification profile is read. Each task contributes at most 50,000 examples to the curated pool, which bounds the compute of a refresh and keeps large tasks from dominating, and the curated pool for CoIN holds 183,249 rows. Certification sets are drawn from each benchmark's training split, so C2R uses no data beyond it, and certification accuracies use a strict exact-match scorer, so they are not directly comparable with the test-protocol accuracies reported in the main tables.

Candidates and screen. On CoIN the candidate set is declared in advance as TextVQA, REC, and OCRVQA, the three tasks on which a generalist trained on every task trails its per-task candidate on the certification sets (Appendix C). Their data is reserved for the experts and excluded from the generalist's pool. The exclusion is a cost choice: the same gaps appear when the generalist trains on every task (Generalist-joint in Table 1), and excluding the three tasks raises G on every task it serves. On UCIT a candidate is trained for every group and the screen selects among them. The screen is the rule σ_{gap} of Section 4.1 with the floor $\Delta_{\min} = 8$: it keeps the groups below the median of G's per-group certification accuracies and flags those below the largest gap between neighboring values if that gap is at least $\Delta_{\min}$, and otherwise nothing is flagged. Because the tasks of UCIT are scored by different metrics, the screen there flags a candidate if and only if its held-out gain over G exceeds a floor $\delta = 2.0$, and the same rule flags the same three groups on CoIN. $\Delta_{\min}$, δ, and α were registered before any certification profile was read and are not tuned. Every flagged candidate is then admitted or rejected by the certificate of Equation (1); a group the screen does not flag stays with G, and so does a flagged group whose candidate fails. Because TextVQA and OCRVQA share the reading-text capability (Section 3.3), a candidate is also trained on their pooled data and certified per group. The cross-group probes of Appendix C, measured against G on each group's certification set, show that the OCRVQA candidate raises TextVQA accuracy by 3.95 points while the TextVQA candidate lowers OCRVQA accuracy by 4.3. TextVQA is therefore served by the pooled adapter, chosen by design as the module of the reading-text capability, and OCRVQA keeps its own adapter as its default. The pooled adapter replaces the default on a question-type slice of OCRVQA only where it beats the default on the certification set by at least 1.0 point and the difference is significant under a one-sided McNemar test at the 5% level with Holm correction across the seven question types (the slice test). Both constants are declared in advance. The test holds for yes/no and who questions alone (Appendix C), so pooling never lowers OCRVQA's certification accuracy on any slice. On the test set, the per-task TextVQA adapter is 0.28 points higher than the pooled one (Tables 1 and 2), a difference the capability-level allocation trades for one fewer module. The library on CoIN thus holds three adapters, for REC, for the pooled reading-text capability, and for the remaining OCRVQA question types, and on UCIT it holds the IconQA and CLEVR experts, which need no pooled candidate (Appendix C).

Routing signals. The similarity measure is a TF-IDF nearest-neighbor margin over question text on CoIN and a nearest-class-mean margin over visual features under the Mahalanobis distance on UCIT. Each uses the modality in which the groups are separated on the stored data, and both are fixed before any certification profile or test query is read. The leak level is declared in advance at $\alpha = 0.02$, and the deployed threshold of every similarity-gated group is the nearest-class-mean decision at 0. A format-defined group carries the same margin test, which serves only the prompts

that its pattern misses, with its threshold set at the leak level α on the certification sets. The feature mean and covariance that define the Mahalanobis distance on UCIT are stored with the library but are not a module. Format patterns are extracted from stored prompts (Appendix C) and deployed only when a pattern covers at least 98% of the group's prompts and at most 6% of every other task's prompts, and the deployed patterns are mutually exclusive. The duplication test compares question text against the stored questions of G and neither reads nor changes any answer.

Dispatch order and re-certification. Within each stage of Equation (2), groups are tested in a fixed order, so the first satisfied certificate routes. At each refresh the similarity measure is recomputed on the current pool, and the route certificate of every installed similarity-gated group is re-evaluated on certification queries disjoint from those used to compute the measure. The set of queries that reaches such a module may therefore change between instances, while its answers on the queries it serves do not. A group whose leak exceeds α is returned to G until a later refresh certifies it again; in our runs no installed group ever needed this.

Evaluation metrics. Let $A_{t,s}$ denote the accuracy on the test set of task s of the instance that serves after position t of a stream of T tasks, scored by each benchmark's own protocol (Chen et al., 2024; Guo et al., 2025). The main tables report the final accuracy $A_{T,s}$ of every task and its mean $\frac{1}{T}\sum_s A_{T,s}$. Forgetting is $F_s = A_{T,s} - A_{s,s}$ for each of the first $T-1$ tasks, averaged over them (Table 14), so a task served by the same frozen expert from the refresh that installs it has $F_s = 0$. Both definitions follow SAME (Xie et al., 2026). Under a schedule that refreshes less often than every task, $A_{s,s}$ is taken at the first refresh that incorporates task s. Routing accuracy is the fraction of test queries that Equation (2) sends to the module designated for their task, with G the designated module of every group outside $\mathcal{I}$. Training cost is the wall-clock GPU time of a refresh, and inference cost that of answering the CoIN test set. On CoIN one refresh takes 133.8 GPU-hours: about 50 for training G on the curated pool, about 80 for the candidate experts, trained once and then frozen, 3.9 for certification, and 0.2 of launch overhead. Certification is inference on the certification sets and grows to 17.7 GPU-hours over eight refreshes. In the from-scratch schedule of Table 4, G is retrained on each growing prefix of the pool, which excludes the declared groups and is capped at 50,000 examples per task, and a refresh whose pool does not grow reuses the previous G, so the cost grows sub-linearly to 236 GPU-hours.

C Certificate Analysis

This appendix documents the certificates behind the installed experts.

Dispatch by group. Table 9 reports, for every group of CoIN, the share of test queries that reach the designated module and the share reaching another group's expert. The format-defined groups are dispatched almost perfectly. TextVQA and REC reach their adapters on every query and OCRVQA on 99.76%, with the rest falling to G and none reaching another expert. Of the groups served by G, ImageNet loses no query to an expert pattern, ScienceQA and VizWiz about 1%, and GQA and VQAv2 1.76% and 4.88%. VQAv2, the largest share of the test set, accounts for most of them. Overall, 97.24% of the CoIN test queries reach their designated module, and 98.68% on UCIT, where both installed groups are similarity-gated.

Table 9: Dispatch on CoIN by group (%). Share of test queries that reach the designated module and share that reach an expert of another group.

Group	Designated module	Other expert
TextVQA	100.00	0.00
REC	100.00	0.00
OCRVQA	99.76	0.00
ImageNet	100.00	0.00
ScienceQA	99.01	0.99
VizWiz	98.87	1.13
GQA	98.24	1.76
VQAv2	95.12	4.88
All queries	97.24	

Dispatch by stage. Table 10 reports the share of test queries that each stage of Equation (2) resolves. On CoIN, every installed group is format-defined and the margin stage resolves no original query: the format stage resolves 36.1% of the queries, the duplication stage 18.9%, and the remaining 45.0% fall to G by default. On UCIT, no group is format-defined, so the format stage resolves nothing, the duplication stage resolves 3.0%, the margin stage 30.7%, and the default 66.3%.

Table 10: Share of test queries resolved by each stage of Equation (2) (%).

Benchmark	Format	Duplication	Margin	Default (G)
CoIN	36.1	18.9	0.0	45.0
UCIT	0.0	3.0	30.7	66.3

Format patterns. Each deployed format pattern is a verbatim substring of the query prompt, extracted at the refresh from an installed group's stored prompts, after removing instruction suffixes shared across tasks, as the phrase most frequent in the group's prompts and least frequent in every other task's. A pattern is deployed only when it appears in at least 98% of the group's prompts and at most 6% of every other task's; a group without such a pattern is similarity-gated instead. On CoIN all three installed groups pass, with the patterns 'bounding box coordinate' (REC), 'Reference OCR token' (TextVQA), and the word 'book' (OCRVQA); each appears in at least 99.8% of its group's prompts and in at most 5.1% of the other tasks'. The question-type test inside OCRVQA is a format test of the same kind, typing a question by keywords (title, author, genre, edition, type of book) and opening words ('is', 'who wrote'). The mapping is fixed by the stored prompts alone; re-extracting on the held-out certification prompts recovers the same patterns at nearly equal frequencies.

Screen outcomes. On CoIN, a generalist trained on every task trails its per-task candidate on the certification sets by 4.3, 10.4, and 7.8 points on TextVQA, REC, and OCRVQA, against at most 1.2 elsewhere, which is the basis of the declared candidate set (Appendix B). The screen confirms it: the three groups lie 14.05 points below the rest of the certification profile of G. This gap is well above the floor $\Delta_{\min} = 8$; any floor up to 14.05 selects the same set and a higher floor selects none, since the rule flags the groups below the largest gap. Every declared candidate passes its certificate. On UCIT the installed candidates show held-out gains of at least $+3.80$ over G, against at most $+0.98$ for the rest, so any floor δ between them installs the same two. The rules also reject candidates: on UCIT the four other candidates are not installed and their groups stay with G. Cross-group probes, measured against G on the certification set of the target group, corroborate the assignment: on CoIN, applying the TextVQA candidate to OCRVQA queries lowers accuracy by 4.3 points and the REC candidate lowers it by 26.7, whereas the OCRVQA candidate raises TextVQA accuracy by 3.95 and the REC candidate by 3.32, so OCRVQA keeps its own adapter while TextVQA is served by the pooled one. On UCIT the IconQA and CLEVR candidates change each other's certification accuracy by only -1.6 and $+0.8$, so no pooled candidate is needed there.

Screen-only run on CoIN. To check that the declaration adds nothing the screen would not find, we also run C2R on CoIN without it: G is trained on the curated pool of all eight tasks, the screen flags the groups below the largest gap of its certification profile, a candidate is trained for each, and the certificates decide the installs. The screen flags the same three groups, all three candidates pass, and the final mean is 70.84 at 170.43 GPU-hours (Table 11). The declaration therefore changes only the cost, since it keeps the three groups' data out of the generalist's pool.

Table 11: C2R on CoIN with the declared candidate set and with the screen alone.

Variant	Installed groups	Mean	GPU-h
Declared set (main tables)	TextVQA, REC, OCRVQA	71.06	133.81
Screen only	TextVQA, REC, OCRVQA	70.84	170.43

Install certificates. Table 12 reports the recall-weighted gain $R_g\Delta_g$, the leak-weighted damage L_gP_g, and their difference $U(g)$ for every installed expert on both benchmarks. On CoIN the installed groups are format-defined, so the REC and TextVQA adapters reach their groups fully and leak nothing, and their certificates reduce to the gain over G, $+12.6$ and $+20.1$ points. The REC adapter trains on 30,000 of the 55,885 stored REC examples (a capped split without the certification hold-out), hence its 64.39 against the oracle's 66.06 (Table 7) despite full routing. The OCRVQA gap is served by two adapters under a question-type test, each certified on its own slice, with the OCRVQA adapter as the group's default. The reading-text adapter beats it on the certification set significantly only on yes/no and who questions ($+3.07$ points, above the declared 1.0-point margin; one-sided McNemar test, $p < 0.001$ after Holm correction across the seven question types), so it is

adopted on that slice. No other type passes both tests, with differences from -25.0 on edition questions to $+2.78$ on other questions, so the default stays. The OCRVQA adapter thus reaches 52% of the group's queries, gains $+6.11$ on them, and leaks 0.2% of the other groups' queries at no measurable damage, for $U = +3.2$; the reading-text adapter passes on the remaining slice with $U = +3.5$ after 0.2 points of damage. On the test set, the split scores 3.1 points above the OCRVQA adapter serving the whole group (65.49 against the per-task oracle's 62.36; Tables 1 and 2). On UCIT both installed groups are similarity-gated and reached fully. The leak of IconQA, measured on certification queries disjoint from those that define the similarity measure, is 0.004, below $\alpha = 0.02$, and although the leaked queries are answered poorly ($P_g = 63.5$, both modules scored by the same rule on the 63 leaked queries), the damage term is 0.27 points. Because every installed group leaks at most 0.004, any level $\alpha \geq 0.005$ certifies the same routes, so the installed set does not depend on the exact α. The certificate values of $+5.0$ and $+3.8$ stay close to the gains and exceed the screen's floor $\delta = 2.0$, and on the test set the two experts add 5.03 and 5.26 points on IconQA and CLEVR (Table 15), consistent with their certified gains. A one-sided 95% lower confidence bound on $U(g)$ exceeds $+7$ points for the REC and TextVQA adapters and stays positive for both UCIT experts ($+1.7$ and $+0.5$), so the installs do not rest on sampling error. Equation (1) omits the group prior p_g. Requiring the exact change in expected accuracy, $p_g R_g \Delta_g - (1 - p_g) L_g P_g$ (Section 4.1), to be positive weighs the damage term by $(1 - p_g)/p_g$, more heavily than $U(g)$ whenever the group holds less than half of the stream. With the terms of Table 12 and the group shares of the certification sets, this prior-weighted criterion also passes for every installed expert; it would fail only for a group with positive damage holding about five percent of the stream or less, which none does.

Table 12: Install certificate of every installed adapter on the certification sets under the deployed dispatcher: the recall-weighted gain $R_g \Delta_g$, the leak-weighted damage $L_g P_g$, and their difference $U(g)$ of Equation (1). An adapter that serves several groups reports one row per slice; certification-set sizes are given in Appendix B.

Installed group (adapter)	Gain $R_g \Delta_g$	Damage $L_g P_g$	$U(g)$
CoIN-REC	12.6	0.0	+12.6
CoIN-TextVQA (reading-text adapter)	20.1	0.0	+20.1
CoIN-OCRVQA, yes/no and who questions (reading-text adapter)	3.7	0.2	+3.5
CoIN-OCRVQA, other question types (OCRVQA adapter)	3.2	0.0	+3.2
UCIT-IconQA	5.3	0.3	+5.0
UCIT-CLEVR	3.8	0.0	+3.8

Choice of the similarity measure. The similarity measure is fixed per benchmark before any certification profile is read (Appendix B), and the route certificate decides whether a route built on it may be deployed. To check that this choice is not a hidden tuning, we evaluate the route certificate of the installed groups under both measures on both benchmarks at $\tau_k = 0$ (Table 13). On CoIN the declared text measure certifies all three installed groups with a leak of at most 0.004, whereas the visual measure leaks 0.131 and is rejected. At $\tau_k = 0$ the text measure is conservative on CoIN, with a recall of 0.49 on its hardest group, so the margin tests of the format-defined groups set their thresholds at the declared leak level α instead, raising recall within the same budget. On UCIT both measures separate the installed groups, with leaks of 0.002 and 0.004 and full recall, since IconQA and CLEVR differ from the other tasks in their prompts as well as their images, so the declared visual measure passes and the text measure would too. The route certificate therefore rejects a measure that does not separate the groups and admits any measure that does, so the choice is decided by the declared level α rather than by hand.

Table 13: Route certificate under both similarity measures: largest leak and smallest recall over each benchmark's installed groups at $\tau_k = 0$; a measure passes when every leak is at most $\alpha = 0.02$.

Benchmark	Similarity measure	Leak	Recall	Passes
CoIN	text (TF-IDF nearest neighbor)	0.004	0.49	yes
CoIN	visual (Mahalanobis nearest class mean)	0.131	0.36	no
UCIT	text (TF-IDF nearest neighbor)	0.002	1.00	yes
UCIT	visual (Mahalanobis nearest class mean)	0.004	1.00	yes

Robustness to paraphrased prompts. We paraphrase every CoIN test instruction using a deterministic synonym dictionary and word swaps, with no model, so that no deployed format pattern survives, keeping images, answers, and answer formats unchanged, and re-run dispatch with the thresholds set on the original certification sets. With the format stage alone, routing accuracy falls from 97.24 to 64.2 and the mean from 71.06 to 65.5, since the format-defined groups fall back to G, while the paraphrase itself moves G alone by -0.4. The margin tests of the format-defined groups, set at the leak level α on the certification sets (Appendix B), restore routing to 90.3 and the mean to 69.6, still above the SAME result on the original prompts (66.82). The format stage is thus a fast path and the margin test covers what it misses, recovering 26.1 of the 33.0 routing points and 4.1 of the 5.6 mean points that the paraphrase removes. On the original prompts every query of a format-defined group is resolved by its pattern, so the margin tests change no result there (97.24, 71.06) and cost a similarity lookup only on the prompts the format stage misses.

D Extended Analysis of the Observations

How the defenses embody the assumption. The devices of the MoE-based family are built to keep task-specific experts intact, as their papers describe them. SAME freezes selected experts during training to limit cross-task interference and stabilizes its router along task-relevant subspaces (Xie et al., 2026), CL-MoE keeps a task-level router beside an instance-level one and updates experts with momentum so that the experts of earlier tasks are retained (Huai et al., 2025), and PASs-MoE stabilizes the rank directions that earlier tasks activate (Hou et al., 2026). Each device presumes that an expert holds what a task needs, the assumption that Section 3.2 tests.

Routing on UCIT. MoE-joint, with SAME's 92.8M trainable parameters, twice those of the C2R library, is trained jointly on all six tasks with no load-balancing or auxiliary router loss, and Figure 5 reports its top-1 expert assignments at the most task-aligned gate. Every task spreads over all experts, and every cell lies between 15% and 21% against the even split of 16.7%, as flat a profile as on CoIN. The normalized mutual information between the task and the top-1 expert, maximized over all 96 gates, is 0.02 on CoIN and 0.04 on UCIT, so task identity explains almost none of the assignments.

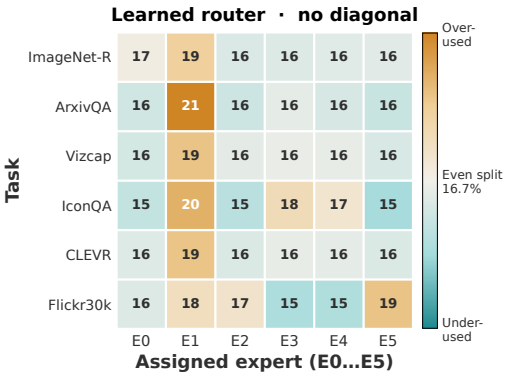

Figure 5: Learned routers do not encode task identity on UCIT. Top-1 expert assignments of MoE-joint at its most task-aligned gate.

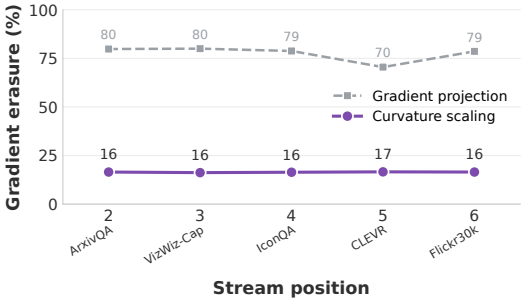

Figure 6: Protection tax on UCIT. Share of each new task's gradient removed per position by a naive null-space projection (gray) and by SAME's curvature-aware scaling (purple).

Protection tax on UCIT. The naive projection removes the components of a new task's update that lie in the span of the layer inputs of earlier tasks, the mechanism that null-space methods such as Adam-NSCL (Wang et al., 2021) and GPM (Saha et al., 2021) use to block interference. On UCIT it removes 70.5–80.0% of each new task's gradient (mean 77.5%), and the curvature-aware scaling of SAME removes 16–17% at every position (Figure 6), the same profile as Figure 2 shows on CoIN, where the two mechanisms remove 78–85% and about 17%. The scaling curve is flat because SAME

clamps its scaling factors to a fixed range, so the fraction it removes hardly depends on the task or its position. The tax is thus a property of sharing experts under a defense, not an artifact of one stream.

E Additional Results

Forgetting. Table 14 reports forgetting under the metric of Xie et al. (2026). Every C2R entry is zero: a single refresh builds the only instance and expert-served tasks are frozen thereafter. C2R-seq averages -0.19, $+0.01$, and $+0.09$ for two, four, and eight refreshes, against -4.23 for SAME, so it keeps forgetting within 0.2 points.

Table 14: Forgetting analysis on CoIN under $F_s = A_{T,s} - A_{s,s}$, averaged over the first seven tasks.

Method	SciQA	TxtVQA	ImgNet	GQA	VizWiz	REC	VQAv2	Avg $\bar{F}$
MoELoRA	-11.91	-12.85	-60.79	-14.41	-15.30	-18.01	-9.62	-20.41
HiDe-LLaVA	-1.43	-8.95	-13.42	-6.34	-6.29	-5.16	-3.12	-6.38
SAME	-2.01	-4.69	-2.21	-5.89	-4.25	-5.71	-4.83	-4.23
C2R	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**

Design validation on UCIT. Table 15 repeats the design validation on UCIT. Every Δ is measured against G alone (76.29), which exceeds SAME, with or without replay, on every task and even the per-task oracle on three of the four tasks it serves, so isolation is not the ceiling: G on its four tasks and the per-task adapters on IconQA and CLEVR would reach 78.10 under task labels. MoE-joint still exceeds the published SAME, so its deficit is not an artifact of our control, and with task-agnostic routing (Appendix D) the gated split buys no specialization. Adding to G a module merged from the experts' weight-space deltas (Appendix G) lowers every task below G alone, by 3.79 points with a uniform average and 2.62 with TIES (Yadav et al., 2023), so the modules must stay disjoint and routed. C2R recovers 1.45 of the 1.65 points to the oracle with two certified modules and no task labels. The 1.32% of test queries that the cascade misroutes in either direction cost 0.36 mean points against label routing (C2R trails G alone by 1.10, 0.14, 0.31, and 0.00 on the G-served tasks and the per-task adapters by 0.30 on IconQA and CLEVR), a rate that stays below the declared level $\alpha = 0.02$ although it exceeds the 0.004 leak measured on the certification sets.

Table 15: Design validation on UCIT: per-task accuracy, mean, and Δ against G alone.

MCIT system	ImgNet-R	ArxivQA	VizWiz-Cap	IconQA	CLEVR	Flickr30k	Mean	Δ vs G
SAME	83.83	91.40	51.33	65.27	53.50	57.43	67.12	-9.17
SAME + replay	84.50	91.60	55.40	69.10	54.80	57.70	68.85	-7.44
Merged (uniform)	88.50	90.10	58.20	74.50	68.30	55.40	72.50	-3.79
Merged (TIES)	89.20	91.00	59.10	76.20	70.10	56.40	73.67	-2.62
MoE-joint	90.72	92.95	56.81	74.67	72.30	58.65	74.35	-1.94
Generalist G alone	91.17	93.27	61.41	80.47	72.57	58.82	76.29	0.00
C2R	90.07	93.13	61.10	85.50	77.83	58.82	**77.74**	**+1.45**
Per-task oracle	90.27	93.23	61.20	85.80	78.13	59.01	77.94	+1.65

Mid-stream accuracy. Under a schedule that refreshes after every second task, we construct the instance after position t by retraining G on the first t tasks' pool and serving each installed group with an expert trained on its own arrived data (Table 16); TextVQA keeps its per-task adapter until the certified reading-text adapter replaces it at $t = 8$. The mean accuracy over the tasks seen by position t is 75.21, 77.51, 72.65, and 71.06 at $t = 2, 4, 6, 8$; under the metric of Table 14 the first seven tasks average -0.02, with TextVQA -0.28 at the switch and the five G-served tasks $+0.02$. C2R-seq reaches a final mean of 71.47, 71.36, and 71.42 with two, four, and eight refreshes, at 152.15, 162.58, and 180.76 GPU-hours (Table 4), so its final accuracy is nearly schedule-independent, within 0.41 points of the single-refresh instance.

Seeds and task order. We rerun the full pipeline under three fresh seeds. Tables 17 and 18 give the per-task mean and standard deviation, and every conclusion is unchanged. Tables 19 and 20 report the per-task results under the two permuted task orders of SAME (Xie et al., 2026). Baseline rows

Table 16: Accuracy matrix $A_{t,s}$ of C2R on CoIN, refreshing after every second task. Rows $t < 8$ are built from the data available by t (G retrained on the first t tasks, each installed group served by an expert trained on its own data, queries sent to their modules); row $t = 8$ is deployed.

Instance	SciQA	TxtVQA	ImgNet	GQA	VizWiz	REC	VQAv2	OCR
$t = 2$	85.90	64.52	–	–	–	–	–	–
$t = 4$	85.85	64.52	97.09	62.56	–	–	–	–
$t = 6$	86.28	64.52	97.01	62.66	61.06	64.39	–	–
$t = 8$	86.75	64.24	96.67	61.56	61.75	64.39	67.61	65.49

are from that study at one decimal, C2R rows from our final instance. C2R's results are identical in every order; MoELoRA swings by 7.1 points and drops to 0.0 on ScienceQA under Order 2.

Table 17: 3-seed mean ± s.d. of C2R on CoIN.

	SciQA	TxtVQA	ImgNet	GQA	VizWiz	REC	VQAv2	OCR	Mean
C2R	86.83±0.24	64.17±0.17	96.62±0.14	61.70±0.22	61.77±0.38	64.50±0.25	67.64±0.13	65.51±0.29	71.09±0.09

Table 18: 3-seed mean ± s.d. of C2R on UCIT.

	ImageNet-R	ArxivQA	VizWiz-Cap	IconQA	CLEVR	Flickr30k	Mean
C2R	90.10±0.26	93.37±0.11	61.48±0.42	85.53±0.22	77.83±0.27	58.91±0.18	77.87±0.10

Table 19: Results on CoIN under Order 2.

Method	SciQA	ImgNet	GQA	OCR	REC	VQAv2	VizWiz	TxtVQA	Mean
MoELoRA	0.0	36.1	55.8	55.9	63.1	62.8	45.5	58.5	47.2
SAME	78.1	89.7	60.8	63.4	60.1	66.1	57.2	61.1	67.0
C2R	86.8	96.7	61.6	65.5	64.4	67.6	61.8	64.2	**71.1**

Table 20: Results on CoIN under Order 3.

Method	OCR	VQAv2	REC	VizWiz	GQA	ImgNet	TxtVQA	SciQA	Mean
MoELoRA	29.9	62.2	59.1	47.4	56.3	45.3	56.9	77.7	54.3
SAME	62.8	65.7	58.6	53.9	61.3	90.1	62.4	80.1	66.8
C2R	65.5	67.6	64.4	61.8	61.6	96.7	64.2	86.8	**71.1**

Sequential baselines with replay. We rerun SAME and MoELoRA on CoIN with the replay budget of C2R-seq (Section 4.3; Table 21). Replay raises MoELoRA to 58.37 and SAME to 68.16 and cuts SAME's forgetting to -1.87 at 263.15 GPU-hours. C2R-seq refreshed after every task stays 3.26 points above SAME with replay at 82 fewer GPU-hours, certification included, with forgetting of $+0.09$, since the installed experts are frozen and replay is spent only on the tasks G serves. The replay generalist alone reaches 66.41 after eight refreshes (on the three expert tasks we score the single-refresh G as a proxy, since both generalists exclude their data), so the experts and their routes add 5.01 of the 71.42. The margin thus comes from the certified experts, not from a larger storage budget. On UCIT the same replay budget lifts SAME to 68.85 (forgetting -2.31, 194.25 GPU-hours), 7.44 below G alone and 8.89 below C2R (Table 15).

Table 21: Sequential baselines given the replay budget of C2R-seq on CoIN.

Method	Mean	Avg $\bar{F}$	GPU-h
MoELoRA	50.58	-20.41	217.00
MoELoRA + replay	58.37	-9.84	249.18
SAME	66.82	-4.23	228.00
SAME + replay	68.16	-1.87	263.15
C2R-seq (8 refreshes)	71.42	$+0.09$	180.76

F LIMITATIONS

The guarantee of Section 3.4 covers only queries served by an installed expert, and the leak of a route is certified on the stored stream, so a new task's queries are covered only after the next refresh. Because ROUTE learns no router, a misrouted query is answered by that module and is not corrected at inference. On CoIN this affects 2.76% of the test queries (Appendix C), most of them VQAv2 queries that match an expert pattern. The format and duplication stages exploit regularities of the benchmark prompts (verbatim templates, repeated questions) that a deployment may lack; the margin stage is the general mechanism and recovers most of the loss under paraphrase (Appendix C). The library is specific to its backbone and its stream: every certificate measures one generalist on the stored data, so a new backbone or stream requires a fresh refresh and may certify different gaps; all experiments use one backbone, LLaVA-1.5-7B, as in prior MCIT work, and transfer to others is untested. The streams hold six and eight tasks, so a cascade with many installed patterns is untested. The screen flags groups but does not propose which share a capability; the pooled reading-text candidate is declared from the Section 3.3 profile and then certified. C2R also assumes stored task data, following the abundant-memory regime of CL; the stored pool, 183,249 examples on CoIN, is a storage cost that the parameter counts of Table 8 omit. Where storing every task is impractical, the sequential update of Section 4.3 trains the generalist on the new tasks and an equal replay budget, while the experts and certificates still read their groups' stored data, and its retention is measured rather than guaranteed (Appendix E). Two directions remain open: operation without stored data, and stronger backbones or higher-rank generalists, which should leave fewer gaps to certify.

G ADDITIONAL RELATED WORK

Post-hoc routing over expert libraries. Outside the continual setting, a line of work reuses trained experts under post-hoc routing, surveyed as MoErging (Yadav et al., 2025). PHATGOOSE (Muqeeth et al., 2024) learns a lightweight gate over independently trained experts; Arrow (Ostapenko et al., 2024) and SpectR (Fleshman & Van Durme, 2025) route among library modules along directions extracted from the module weights, without training a router; we adapt Arrow as the weight-space baseline of Table 6. In these works, the set of experts is chosen by clustering or by hand rather than by measured gain, and the routing error is not measured against a declared level; C2R decides both by certification and inherits its continual guarantee from frozen, disjoint modules. The merged-library rows of Tables 5 and 15 add to G a single module merged from the installed experts' weight-space deltas $\Delta W = BA$, not their LoRA factors, by task arithmetic (Ilharco et al., 2023) with a uniform average or TIES (Yadav et al., 2023), the routing-free alternative.

Isolated adapters under hard routing. MLLM-CL (Zhao et al., 2025b), the closest design to C2R, trains a fresh LoRA per task and selects one at inference with a router that is itself an MLLM tuned on a few examples of each task; it is evaluated only on its own domain and ability streams, so it is absent from our tables. C2R installs a module only for a certified capability, so the count follows the generalist's gaps rather than the number of tasks, and its routes are certified rather than learned.

Nearest-class-mean routing. The margin stage follows the nearest-class-mean rule of iCaRL (Rebuffi et al., 2017), a standard training-free classifier in CL: C2R applies it to groups rather than classes, compares a group's mean with G's pool, and certifies the route by its measured leak instead of assuming it. The NCM prototype baseline of Table 6 is a whitened nearest-class-mean with cosine similarity over frozen CLIP-CLS features, without a certificate. Both baselines choose among the same deployed modules as the cascade and are scored by the same metric. The task-ID row of Table 6 applies the same classifier at task granularity, the router a per-task library needs.

Relation to conformal prediction. The leak level α of the route certificate is the marginal error budget that inductive conformal prediction (Vovk et al., 2005; Papadopoulos et al., 2002; Angelopoulos & Bates, 2023) controls by calibrating a threshold on held-out data. C2R deploys the nearest-class-mean threshold and reports the leak against α on certification queries not used to compute the measure, without claiming the guarantee, because test and stored queries need not be exchangeable. A certificate in this paper is thus a statistic measured on held-out certification data and compared with a criterion fixed in advance, unrelated to the certified robustness of Cohen et al. (2019), which bounds a classifier's output under input perturbations.