Merge confusable experts, split only separable ones

W. Schulz

Nov 11, 2026

Abstract

A literature synthesis and experiment plan on when to merge, split or spawn experts in continual learning with frozen or isolated experts. Under hard selection both operations change selection error first and capacity second. Re-read as a merge-and-split experiment, the lab's Split-CIFAR-100 task-count sweep shows going from 10 to 5 experts raised system accuracy from 0.342 to 0.422, and going to 20 cut it to 0.208. Merge pairs the router confuses, keep pairs it separates well, and split only at a plateau when the router can separate the children. Nine experiments (MS0–MS8) are pre-registered. The first, MS0 (10 seeds), confirms that a single merge helps behind max-softmax routing and hurts behind a strong shared-space router, that router confusion predicts which merge pays, and that keeping the parents beside the merged expert beat replacing them.

W. Schulz, NeuroGenesys, November 2026. A literature synthesis and experiment plan, not an experiment. Cited numbers are the sources' own and were not reproduced here; the lab's results carry their ledger IDs, and the one result outside the ledger (a single-seed router pilot) is labelled. The T-sweep reading in the first section is a re-reading of existing ledger results (8.18), not a new experiment. One source (MoRAM) was read only at snippet level and is flagged where it appears. Experiments MS0–MS8 are pre-registered; MS0 has run (appendix and ledger 14.8–14.12), the others have no results yet. The comparisons with NeuroGenesys results refer to the router architecture and forgetting-as-selection papers and the findings ledger. Research notes: `research_notes/Merging and splitting MoE experts/` *in the NeuroGenesys repository.*

In continual learning with frozen or isolated experts, merging and splitting change selection error first and capacity second, so the case for each depends on the router. The lab's own Split-CIFAR-100 libraries already contain the answer for the max-softmax regime. Going from 10 tasks to 5 is exactly a merge of adjacent expert pairs, retrained on the union of their classes: the R0 cache's $T = 5$ class lists are the unions of consecutive $T = 10$ pairs for every seed checked. That merge raised system accuracy from 0.342 to 0.422. Selection accuracy gained 0.110 and within-task accuracy lost only 0.056. Going from 10 to 20 tasks is the matching uniform split, and it cut accuracy to 0.208, because ε rose by +0.144 while within-task accuracy gained only +0.020 (ledger 8.18). The general rule follows from the identity $\text{acc} = \text{A_sel} \times \text{A_within}$, from REAP's merge-error analysis and from splitting-steepest-descent theory. Merge a pair when the router confuses the two experts often, or when it barely distinguishes them, because the merge erases a distinction the router was not using well. Keep a pair separate when the router reliably tells two functionally different experts apart, because merging them is forgetting. Split an expert only when its within-task loss has plateaued, its routed data is multimodal in a way that changes the loss, and the router can be shown to separate the children. Spawn a new expert only from a buffered, coherent novelty stream, behind a gate on held-out old-task ε. The literature has measured neither the forgetting caused by a merge event itself, nor selection error before and after a split, nor any head-to-head comparison of grow, reuse and merge policies. A single-seed router pilot also shows the calculus flipping once the shared feature space is strong. With ImageNet-ViT features, ε fell to about 0.18, and

the router's own 100-way head beat routing to the experts. Seven experiments, the first of which is a zero-training reanalysis of caches already on disk, would settle the open questions.

The lab's task-count sweep is already a merge-and-split experiment

The decision to merge or split looks like a capacity question. With frozen experts it is mostly a question about the router. The lab's identity acc = A_sel × A_within holds exactly for per-task experts with disjoint label sets (Optimal router architecture). A merge removes one candidate from the router's choice and asks one expert to cover two label sets. A split does the reverse. Ledger entry 8.18 measured both directions under max-softmax selection: ε went 0.534 → 0.644 → 0.788 at T = 5, 10 and 20, with step changes of +0.109 [+0.099, +0.119] and +0.144 [+0.136, +0.153]. Over the same steps, within-task accuracy went 0.905 → 0.961 → 0.981 and system accuracy 0.422 → 0.342 → 0.208 (lib/findings.ts). The R0 caching run reuses the Phase 1 harness unchanged. In its cache under `~/.cache/neurogenesys-router/`, the T = 5 class lists equal the unions of consecutive T = 10 task pairs for seeds 0, 1 and 2, and both libraries are scored on the same 10,000 test images (in a different order, so they align by image hash). The T-sweep is therefore a controlled uniform merge, with each merged expert retrained from scratch on its pair's data. The T = 20 libraries now being cached will provide the matching uniform split.

In log terms the trade is lopsided. Merging pairs raises ln A_sel by +0.27 and lowers ln A_within by only 0.06, so the selection gain is about 4.5 times the capacity loss. Splitting every expert lowers ln A_sel by 0.52 and raises ln A_within by 0.02, a 25-fold loss. In this regime the marginal expert is a liability: each added candidate costs more in routing than it returns in specialization. The same pattern appears outside the lab. CN-DPM's classifiers barely forget (Split-CIFAR-100 per-task accuracy 55.42 at initialization, 55.24 at the end). Its generative gate, however, identifies the right expert only 48.18% of the time on 5-task Split-CIFAR-10 and 31.14% on 20-task Split-CIFAR-100, which leaves end-to-end accuracy at 20.10 against 73.80 for iid-offline training (CN-DPM). SEED sees diminishing returns past five experts (SEED). MoRAM names "routing ambiguity" as the residual failure of LoRA-MoE methods that add isolated, frozen experts (preprint, snippet level) (MoRAM).

Two caveats bound this reading. First, the T = 5 experts are oracle merges, trained jointly on both tasks' data. A post-hoc merge in an exemplar-free stream has no such data and can only do worse within-task. Whether it keeps the selection gain is the first experiment below. Second, the regime depends on the router. In a single-seed validation pilot at T = 10 (seed 100, not in the ledger), RanPAC on stream-legal first-session features reached ε ≈ 0.63–0.66, no better than max-softmax's 0.653. RanPAC on ImageNet ViT-B/16 features reached ε ≈ 0.18. With that router, routed accuracy was about 0.657, and the router's own 100-way readout was about 0.798 (pilot files in `research_notes/Optimal router architecture/pilot/`). Once the shared space is strong, the experts' within-task accuracy becomes the binding term. A single shared head, the limit case of merging everything, then beats the expert library. ImageNet pretraining overlaps CIFAR-100 classes, so this shows the direction of the effect, not its size. The pilot also exposes a measurement trap that matters for split decisions. Under max-softmax routing the experts' conditional within-task accuracy was 0.950, against 0.865 under RanPAC on first-session features, because a self-report router sends each expert mainly the inputs it already finds easy. Split triggers should use task-given accuracy, not accuracy conditional on the router.

Merge when the router is not really choosing, and never where it chooses well

The sharpest merge criterion comes from REAP's analysis of summed-gate merging in sparse MoEs. Suppose experts i and j are replaced by one expert whose gate is the sum of theirs. The best fixed

mixture still leaves an irreducible error of approximately $E[(g_i+g_j)^2] \cdot \mathrm{Var}[r] \cdot E\|f_i - f_j\|^2$, where $r = g_i/(g_i+g_j)$ is the router's input-dependent preference between the two. "No fixed α removes this error" while the router's preference varies and the experts differ. Pruning instead has error bounded by the pruned expert's gate mass, independent of how the router's policy varies (REAP). The formula says a merge is safe only when the router was not really choosing between the two experts ($\mathrm{Var}[r] \approx 0$) or when choosing made no difference ($f_i \approx f_j$). In a stream, the distinction a merge erases is exactly what later shows up as forgetting.

The empirical record supports this on generative tasks. At 50% expert reduction on Qwen3-30B-A3B, REAP pruning kept the coding average at 0.541 against a 0.558 baseline. Merging methods fell to 0.364 (HC-SMoE) and 0.397 (M-SMoE), and HC-SMoE collapsed to 0.008 on WildBench. Merging contracted the expert output manifold by up to 100× in late layers of fine-grained models (REAP). Merging stays competitive at 25% compression, on multiple-choice scoring and on coarse-grained models. At 50% on Qwen3-30B, HC-SMoE's multiple-choice score of 0.542 beat REAP's 0.503 (REAP). Shared-subspace merging narrows the gap. Sub-MoE keeps Qwen3-30B-A3B at 0.57 at 128 → 64 experts, where HC-SMoE scores 0.38 (Sub-MoE). ConMoE does not fuse weights at all: it remaps each retired expert's router slot to its nearest surviving prototype, and it matches or beats REAP on multiple choice at 25% reduction (ConMoE). Remapping is pruning plus redirection. It is also the only consolidation that is fully reversible, provided the retired weights are archived.

The signal used to choose merge candidates has moved from router logits toward functional similarity. In fine-tuned Switch-T5, router-logit similarity was the best grouping signal (COPA 68.00, against 60.00 for expert features) (MC-SMoE). In retraining-free decoder LLMs, expert-output similarity won (0.5459) and router logits came last (0.3153) under average linkage (HC-SMoE). For importance, usage frequency is a poor proxy. On Mixtral, frequency-based pruning scored below random (60.77 against 63.04 at six of eight experts) (NAEE). On Qwen3-Coder-480B at 50%, frequency pruning collapsed coding to 0.010, while REAP's gate-weighted activation norm held 0.619 (REAP). Weight similarity and functional redundancy also come apart. Grok-1's experts have near-zero weight cosine but high output similarity. Mixtral's weight similarity of 0.2–0.4 is an artefact of upcycling from Mistral-7B, which makes it a poor testbed for natural redundancy (Lo et al.). A defensible merge statistic therefore needs both router terms and output terms. The router-only terms ($E[(g_i+g_j)^2]$, $\mathrm{Var}[r]$, and the responsibility overlap that SMEM used to rank merge candidates, $J_{merge} = P_i^\top P_j$) can be kept as running averages (SMEM). The output gap G_{ij} needs both experts evaluated on a sample of the same inputs.

Per-task continual learning adds a twist that REAP's token-level setting lacks. There, experts share an output space. Per-task experts own disjoint label sets, so every confusion between them is consequential: a wrong-task expert is never right. Merging two such experts does not average them. It builds one expert with a joint head over both label sets, which moves the discrimination between the two tasks out of the router's cross-expert comparison of incomparable self-report scores and into a single jointly trained softmax. A jointly trained head is far better at that discrimination. At $T = 10$, Joint's implicit task-selection accuracy was 0.576, against max-softmax's 0.356 (Optimal router architecture). The $T = 10 \rightarrow 5$ gain is this effect. The table puts the two settings together.

Pair state	Router rarely confuses i and j	Router often confuses i and j
Similar functions (low G_{ij})	Redundant. Merge or remap only when memory binds; accuracy barely moves either way	Harmless confusion. Merge: it frees memory and removes a candidate almost for free

Pair state	Router rarely confuses i and j	Router often confuses i and j
Different functions (high G_ij)	Healthy specialization. Keep both; merging erases a distinction the router uses, which is forgetting (REAP's Var[r] term)	Consequential confusion. With disjoint-label experts, merge by joint training or distillation so a calibrated head makes the call. With shared-output token experts, fix the router (shared-space scorer, shortlist or hierarchy) rather than merge

The bottom-right cell is where the lab's thesis bites hardest, and the remedy depends on the setting. When a stronger router is available, it beats any merge. The lab's hierarchical MoE report shows that a group-then-expert cascade helps only if its group-level score is better estimated than the flat score. Merging the confusable pair is the cheapest way to achieve that.

Merge in function space for scratch experts, in weight space only with a shared origin

How to merge depends on whether the experts share an origin. REAP reports that weight-space distances between clustered experts "far exceed" those between fine-tuned variants, so expert merging is harder than ordinary model merging (REAP). Networks trained from a shared early checkpoint become linearly mode-connected (Frankle et al.). Independently trained ones are connected, if at all, only after permutation alignment (Git Re-Basin). Inside fine-tuned Switch models, alignment added 0.7–2.0 points, and frequency-weighted averaging beat uniform averaging by up to 4 COPA points (MC-SMoE). In retraining-free LLM merging, uniform, frequency-weighted and ZipIt-style merges landed within 0.3 points of each other (HC-SMoE). The lab's SlimResNet18 task experts are trained from scratch with independent initializations. Averaging their weights should fail without alignment, and probably with it. The defensible merge for them is in function space: distil both teachers into one student over the union of labels, or retrain on the union. Low-rank adapters on a shared frozen backbone are the other case. They share an origin, and task-vector methods apply directly. MagMax fine-tunes tasks sequentially, then keeps each parameter's maximum-magnitude update. With no exemplars it reached 84.16 on CIFAR-100/5 and 76.75 on CIFAR-100/50, and fine-tuning sequentially before merging rather than independently added up to about 12 points by reducing sign conflicts (MagMax). Merging works best when the experts were never truly isolated. That is the opposite of the lab's frozen-expert design, and it argues for building future expert libraries as sparse deltas on a shared generalist, as the hierarchical report recommended for other reasons.

The router needs handling too. Almost every retraining-free method leaves the router untouched and redirects tokens bound for any group member to the merged expert, summing their gates (HC-SMoE; ConMoE). No paper found compares that with a brief router-only recalibration. A statistics-keyed router of the kind the lab favours makes the router side of a merge exact and free. Its class means and Gram matrix are additive, so a merged expert's routing statistics are just the union of its parents' class statistics. Nothing about other experts changes. Calibration data is the hidden cost. With the same pruning rule, Mixtral kept GSM8K at 37.07 when calibrated on MATH but only 24.87 when calibrated on C4 (NAEE). A merge chosen on new-task data alone will forget old tasks. In exemplar-free continual learning, old-task calibration has to come from stored statistics, a small buffer or a generative replay. That buffer is the same memory the router report found the contrast signal needs.

What a merge costs in forgetting is largely unmeasured. MagMax cut first-task forgetting from 25.7% to 9.8% and second-task forgetting from 22.3% to 1.7%, relative to sequential fine-tuning. Its accuracy still fell monotonically with the number of merged tasks, 84.16 → 80.41 → 78.49 → 76.75 at T = 5/10/20/50 (MagMax). That decline is interference, not selection error, and it gives a slope to compare against the lab's ε growth. A frozen-expert library loses accuracy to routing, a merged model loses it to interference, and the cheaper slope should win. No continual-learning paper reports accuracy on old tasks immediately before and after a merge event, separately from training drift. CN-DPM's only merge step pruned redundant experts above a 0.9 similarity threshold (7 → 5 on Split-MNIST) and was judged on expert count, not on gating accuracy (CN-DPM).

Split only at a plateau, and only if the router can separate the children

Splitting has one principled criterion. Splitting steepest descent treats a unit as splittable when its "splitting matrix" $S(\theta) = E[\Phi'(\sigma)\, \nabla^2_\theta\, \sigma]$ has a negative minimum eigenvalue at a point where ordinary gradient descent has stalled. The optimal split is into exactly two equal-weight children displaced along $\pm v_{min}$, which reduces the loss by $-\varepsilon^2 \lambda_{min}/2$. Splitting into more than two children never helps to first order, and units can be ranked independently because their splitting matrices have no cross terms (Liu et al. 2019). The criterion has two important properties. The split is function-preserving at birth, so it costs nothing immediately. And it pays only at a plateau, because before that gradient descent still gains $O(\varepsilon)$ per step while a split gains only $O(\varepsilon^2)$. The index has been validated for neurons and convolutional filters. On keyword spotting, a split-grown net reached 95.36% with 282.6K parameters, against DS-CNN's 94.85% with 413.7K (Liu et al. 2019). Nobody has computed it for MoE experts. Exact evaluation costs $O(n \cdot d^3)$, which is infeasible for million-parameter experts without Lanczos or Hessian-vector approximations.

The other split and spawn signals come from mixture models and growing networks, and none has been validated as a trigger for MoE experts. SMEM splits the component that fits its responsibility-weighted local data worst (SMEM). G-means splits a cluster when an Anderson–Darling test rejects Gaussianity along the axis joining its two candidate children (Hamerly & Elkan). Growing Neural Gas inserts a unit next to the one with the largest decayed accumulated error (Fritzke). For spawning, DP-means opens a new cluster when a point lies farther than λ from every centre (Kulis & Jordan). CN-DPM buffers improbable samples in a short-term memory and trains a new expert only when that memory fills (CN-DPM). DynMoE adds an expert when tokens activate none and removes unused ones (DynMoE). SEMA expands a layer only when all of its autoencoder descriptors flag an input as novel. Its sparse growth deployed 0.55–0.65M parameters, against 0.65–1.90M for expanding every task, and still led every baseline, for example 74.53 on 20-task ImageNet-R (SEMA). Two warnings come with these signals. Load is not a split signal. DeepSeek-V3 answers load by replicating hot experts, which changes throughput, not function (DeepSeek-V3). A hot expert may be a valuable generalist, and the hierarchical report found that removing DeepSeekMoE's shared expert raised Pile loss from 1.808 to 2.414. And multimodal routed data is not sufficient on its own. A nonlinear expert can serve two clusters well, and theory shows nonlinear experts are exactly what lets an MoE recover cluster structure (Chen et al. 2022). A split needs multimodality that changes the loss: different per-cluster error, or conflicting per-cluster gradients.

Initialization decides whether the children ever diverge, and the two working recipes sit at opposite extremes. When the trained router is reused, tiny noise is enough. Orthogonal Growth copies experts with $\alpha = 0.01$ Gaussian noise and reuses the router; $\alpha = 0$ cost about 1% downstream, and random initialization of new experts was "significantly worse" (Orthogonal Growth). When the router is new, the children must differ substantially. Naive upcycling sent tokens to about two experts across all lay-

ers, and Drop-Upcycling's partial re-initialization at r = 0.5 was the best long-run setting, reaching 40.3 against 34.5 from scratch at 8×1.5B over 500B tokens (Drop-Upcycling). Lifelong-MoE found that random initialization of expanded experts "leads to poor performance". It also found that freezing only old experts, or only old gating dimensions, was "not effective" and that only freezing both worked, which is the LLM-scale version of protecting the selector (Lifelong-MoE). Its effect sizes are small and from a single run (TriviaQA F1 between 5.8 and 7.1). The measured gains from growth mostly come from reusing sunk compute. Orthogonal Growth beat from-scratch training by 6.18 points at matched extra FLOPs but was only "comparable or slightly superior" at a fixed total budget (Orthogonal Growth). Fine-grained models also sit on the flat part of the granularity curve. OLMoE gained about 10% going from 8 to 32 experts and only 1–2% more at 64 (OLMoE). Post-hoc splits there can at most add targeted capacity for an underserved domain or task. They cannot add general quality.

The selection cost of a split is easy to predict. By construction the two children are the most confusable pair in the library, competing for exactly the inputs their parent won. Under top-k ≥ 2 with a shared output space that is mild. Under hard top-1 selection among frozen, disjoint-label experts it is severe, and the lab's $T = 10 \rightarrow 20$ step measures it: -0.134 accuracy. Function preservation helps only at first. Right after the split the twins compute the same function, so confusing them costs nothing. That cost grows as they diverge, and it is the consequential selection error Σ P(route j | target i)·G_ij. A split should therefore be accepted provisionally and confirmed only if the children's loss falls faster than their consequential confusion rises. That is SMEM's accept-or-reject test applied with a delay (SMEM). The routing margin between siblings, which the lab's margin-abstention router already computes, measures divergence and selection risk at once.

Price every expert by the selection error it adds

Together these results give one cost–benefit test per operation. The Bayesian views of model selection already have this form: split or spawn if and only if the likelihood gain exceeds a complexity price, set by BIC's (p/2) log n, DP-means' λ or a Dirichlet concentration α (Kulis & Jordan; Jain & Neal). The forgetting-as-selection thesis adds the term those priors omit. An expert's price is its memory and compute plus the expected increase in consequential selection error from adding a competitor to the router. Priced that way, the lab's T-sweep shows that experts holding 5 CIFAR classes each cost more than they return under max-softmax. A RanPAC router with ε near 0.18 would cut that price by more than two thirds and could make splits pay. The operations then differ along four axes.

Operation	Forgetting	Plasticity	Memory	Routing and selection error
Weight-space merge (average, TIES, MagMax)	Interference grows with merged count (MagMax −7.4 pts from T = 5 to 50); erases any distinction the router used (REAP Var[r] term)	Low; sequential fine-tuning before merging helps by up to ~12 pts	Constant in tasks	Removes candidates; ε term disappears for a single merged model

Operation	Forgetting	Plasticity	Memory	Routing and selection error
Function-space merge (distil or retrain on union)	Needs old-task data or statistics; oracle version cost 0.056 within-task at T = 10 → 5	Moderate	Halves expert count per merge	Moves the pair's discrimination into a joint head; +0.110 A_sel in the oracle
Prune	Removed expert's capability is lost; next-best expert substitutes (REAP 0.541 vs 0.558 code at 50%)	Unchanged	Large saving (Mixtral 89.9 → 46.9 GB at 4 of 8 experts, NAEE)	Router renormalises over survivors; no averaging collapse
Remap (ConMoE)	Reversible if weights are archived	Unchanged	Logical saving; physical saving needs a prototype runtime	Same as prune, without deleting anything
Function-preserving split	None at birth	Gain $-\varepsilon^2\lambda_min/2$ only at a plateau	+1 expert	Adds the most confusable sibling; cost rises as the twins diverge
Spawn from a novelty buffer	Near zero if old experts and gates are frozen	High	+1 expert per accepted cluster	Novelty score is a weak router: SEMA's descriptor routing 50.36 vs 53.32 learned on ImageNet-A; CN-DPM gate 31% at 20 tasks

The decision rules follow. Merge pair (i, j) when the REAP-style risk $W_ij \cdot V_ij \cdot G_ij$ is small, or, for disjoint-label experts, when their mutual confusion under the deployed router is high and a function-space merge can be calibrated on old-task statistics. Rank candidates by confusion or overlap, as SMEM ranks by J_merge, but gate on the risk term, not on similarity alone. Split expert k only when its task-given loss has plateaued, a splitting-index proxy or a multimodality test that changes the loss fires, and a held-out check shows the router separating the children. Initialize the children function-preservingly, with noise scaled to whether the router is reused. Spawn only when a buffered novelty stream forms a coherent cluster, and admit the new expert only if held-out ε on old tasks rises by less than a pre-set margin. That last gate is the admission rule the router report already proposes; the graded-succession design supplies its evaluator. All of this is synthesis. No published system implements these rules, and none of their thresholds has been calibrated.

The steelman against this framing is that co-trained MoEs behave differently. In continual pre-training on a 400B + 200B token stream, penalty-balanced routers recovered from the distribution shift within

about 500 steps, and the MoE forgot less than a dense model (FineWeb loss increase 0.037 against 0.066 with 40% replay) (Thérien et al.). With soft, co-trained routing over a shared output space, replay and few broad shifts, selection error is small and splits mainly buy capacity. The thesis applies to the regime the lab actually runs: many narrow tasks, frozen experts and hard selection. It would weaken as routing becomes soft and replay becomes cheap. Experiment MS3 below tests where that boundary lies.

Seven experiments settle what the literature leaves open

Everything in this section is a proposal. Thresholds are placeholders to fix at pre-registration, and compute figures are estimates from R0's measured timings unless marked unmeasured. The design follows the Phase 1 rules: paired seeds, 10,000-resample bootstrap intervals with seed as the unit, Holm correction within each primary family, a test split never used for tuning, and kill criteria with a deviation log (preregistration). Every arm should write `selection/v1` events so `lib/reliability` can audit them. The plan avoids duplicating existing studies. Admission of new experts stays in router experiment R5. Co-activation structure stays in HT1, which is blocked because the Modal workspace refuses GPU functions. MS5 below therefore runs locally.

ID	Question	Design	Cost (estimate)	Primary hypotheses and kill rule
MS0	Which single merges and splits help under each router?	Offline on the R0 cache. Build hybrid libraries: swap one adjacent $T = 10$ pair for its $T = 5$ parent (5 pairs $\times$ 10 seeds), and, once $T = 20$ is cached, swap one $T = 10$ expert for its two $T = 20$ children. Align test images by hash. Score under max-softmax, energy, RanPAC (first-session) and RanPAC (ViT). Predictors from validation: pairwise confusion, prototype overlap, feature CKA	CPU minutes	H0a: under max-softmax, single-pair merges raise accuracy (sign test). H0b: Δacc rises with pre-merge confusion (Spearman $\rho > 0.3$). H0c: under RanPAC-ViT, merge gains shrink and some splits turn positive. Kill: ρ's interval includes 0, so confusion-ranked merging is no better than arbitrary pairing

ID	Question	Design	Cost (estimate)	Primary hypotheses and kill rule
MS1	Can a post-hoc merge without old data recover the oracle gain?	For the most-confused, a random and the least-confused non-adjacent pair per seed: retrain on the union (oracle), two-teacher distillation with a 2,000-image buffer (matching `mem_size`), distillation on the newer task's data only, and weight averaging with and without permutation alignment (loss barrier reported)	$\approx$ 150 s per merged expert from R0 timings (772 s for ten experts); $\approx$ 7 h MPS for 4 pairs $\times$ 4 methods $\times$ 10 seeds; waits for R0 to release the GPU	H1a: confusion-ranked merges beat random ones by $\geq$ 0.02 accuracy. H1b: buffered distillation recovers $\geq$ 50% of the oracle gain. H1c: unaligned averaging collapses. Kill: buffered distillation recovers $<$ 25%, so merges must be decided before freezing (CN-DPM-style delayed commitment)
MS2	Does any diagnostic predict which expert is worth splitting?	Rank T = 10 experts by task-given error, two-means silhouette and Anderson–Darling on routed 160-d features, per-sample gradient covariance, a Lanczos λ_min proxy, load and a random draw; use T = 20 children as oracle splits, plus trained splits (Net2Net copy, class partition chosen by within-expert confusion vs random partition)	Oracle part offline; trained splits $\approx$ 4 h MPS (unmeasured)	H2a: under max-softmax no single split raises accuracy. H2b: under RanPAC-ViT, trigger-ranked splits beat random ones. Kill: no trigger beats random under any router, so splitting lacks a usable diagnostic at this scale

ID	Question	Design	Cost (estimate)	Primary hypotheses and kill rule
MS3	Does REAP's risk term predict forgetting caused by the merge event itself?	Co-trained MoE of MLP experts over cached ViT features, class-incremental $T = 20$, learned softmax router and shared head; merge single pairs mid-stream and measure old-task accuracy just before and after; compare $W \cdot V \cdot G$ against C_{ij}, weight cosine and random. Split arms: copy + $\alpha = 0.01$, Drop $r = 0.5$, $\pm v_{min}$, cluster partition; track sibling routing margin and divergence of routed sets	CPU minutes per run (unmeasured)	H3a: $W \cdot V \cdot G$ predicts merge-induced old-task loss better than C_{ij} (paired difference in ρ). H3b: with a reused router, copy + $\alpha = 0.01$ diverges no slower than Drop $r = 0.5$ at lower loss. Kill: no statistic beats weight cosine

ID	Question	Design	Cost (estimate)	Primary hypotheses and kill rule
MS4	Which lifecycle policy wins on one stream, at matched memory?	Same experts and stream under grow-only, grow + confusion merge at budget $K \in \{5, 10\}$, grow + REAP prune, remap, SEED-style fixed-K reuse, single MagMax model, and novelty spawn with a buffer; ε and task-given A_within logged separately, plus per-expert utilisation and redundancy curves per step	Offline to a few MPS hours	H4: grow + merge at $K = 10$ beats grow-only at $T = 20$ using half the expert memory. Also fit the MagMax accuracy(T) slope against the frozen library's A_sel(T)·A_within(T) to locate the crossover T*. Kill: grow-only wins at every K, so the marginal expert is not a liability once the router is fixed
MS5	Does the merge-risk term hold in a real LLM MoE?	OLMoE-1B-7B (13.8 GB bf16) locally on MPS: per-layer C, V, W, G on C4 + code calibration with 3 seeds; a table of near-duplicate pairs by output similarity; single-pair summed-gate merges and single prunes for sampled pairs, scored by perplexity and a small generative set; repeat at intermediate checkpoints to track redundancy over training	Unmeasured; HT1's smoke test gives the throughput; disk has $\approx$ 59 GiB free	H5: W·V·G predicts Δloss across pairs better than router-logit or weight cosine. Kill: no statistic predicts beyond frequency, so merge choice in LLM MoEs needs direct evaluation

ID	Question	Design	Cost (estimate)	Primary hypotheses and kill rule
MS6	Do the rules transfer to a continual LoRA-MoE on language tasks?	Qwen2.5-1.5B LoRA-MoE (harness from the weight-editing study) on a TRACE subset with a fixed expert budget; consolidate by frequency, output-similarity merge, REAP prune, remap or MagMax; relate retention to pre-merge risk and sign conflict	Hours on MPS (unmeasured)	H6: risk-gated consolidation retains more old-task accuracy than frequency-based consolidation. Run only if MS3 and MS5 support the risk term

The order is MS0 → MS2 (offline part) → MS3 → MS1 → MS4 → MS5 → MS6. MS0 costs nothing, and its merge half can run today, because the $T = 5$ and $T = 10$ libraries are complete for seeds 0–9. The decision logic is fixed in advance. If H0a and H0b hold and H1b passes, the lab should adopt confusion-driven consolidation with buffered distillation as the default lifecycle operation for frozen-expert libraries. If H0c and H2b hold, splitting earns a place only behind a strong shared-space router, and the router study's R1 becomes the gate for any split policy. If MS1's kill fires, merges must happen before experts are frozen, by grouping tasks in a short-term buffer and training one expert per group. That is a different architecture, and the evidence points toward it.

Conclusion

The question assumes merging trades plasticity for memory and splitting trades memory for capacity. With frozen experts and hard selection, both operations are really router operations. A merge succeeds when it removes a choice the router was making badly, and it fails, in exactly the way forgetting fails, when it removes a choice the router was making well. A split succeeds only when the router can make one more choice reliably. That makes router quality the master variable. The lab's own T-sweep shows that under self-report routing an expert holding 5 classes is worth less than nothing. The ViT pilot hints that with a strong shared space, even the 10-class library is worth less than one shared head. In both regimes the expert count is set by the router's resolving power, not by the data's diversity.

The most useful reframing for NeuroGenesys is that "how many experts" is not a quantity to fix in advance. It is an equilibrium to maintain: consolidate where the router is confused, divide where it is confident and the experts are strained, and audit both on held-out probes the router never trained on. Building future libraries as deltas on a shared generalist would make every merge in this report cheaper, by moving it from distillation into weight space. That is the second place where this question and the hierarchical-MoE work converge on the same design.

Experiments

The experiments proposed above are pre-registered as MS0–MS8, with code and the queue in `research_notes/Merging and splitting MoE experts/experiments/`. The plan adds MS7 (a learned predictor of which edits pay) and MS8 (where to cut: arbitrary vs human-taxonomy vs data-driven expert boundaries). The claim above that T = 5 tasks are unions of adjacent T = 10 pairs was checked for all ten seeds and holds. The first experiment, MS0, has run; its results are summarised in the appendix (ledger 14.8–14.12). The appendix also lists the nine experiments and their status.

Appendix: the MS0–MS8 experiment plan

Nine experiments were pre-registered from this synthesis on 2026-11-11. MS0 is complete (results below); the others have no results yet. MS1–MS4, MS7 and MS8 are pre-registered, smoke-tested and queued. MS5 and MS6 are pre-registered as design freezes and are not queued. One check is already done: T = 5 tasks are the unions of adjacent T = 10 tasks for all 10 seeds (5 of 5 pairs each; no other pairing matches).

ID	Question	Where	Status
MS0	Which single consolidations, fusions, retirements and splits of a frozen T = 10 library raise system accuracy under each router, and does pre-merge confusion predict which?	Local CPU, no training	Complete (2026-11-12, 10 seeds; 5 of 6 hypotheses supported, H0f falsified)
MS8	*Where to cut.* At equal K and parameters, is the arbitrary, human, embedding, confusion, competence, gradient-conflict, objective-searched or online partition best, and does that depend on the router?	Local CPU	Pre-registered, queued
MS2	Does any diagnostic (task loss, multimodality, load, gradient interference, Lanczos λ_min) pick the expert whose split pays? Does the partition inside the expert matter?	Local CPU, then MPS	Pre-registered, queued

ID	Question	Where	Status
MS3	In a co-trained MoE, does REAP's W·V·G predict the forgetting caused by the merge event itself better than co-routing or weight cosine? Which split initialisation diverges?	Local CPU	Pre-registered, queued
MS4	Lifecycle A–D at matched replay memory: (A) one model, (B) frozen specialists, (C) + probationary creation and splitting, (D) + budgeted consolidation. Plus an MDL view of population size	Local CPU	Pre-registered, queued
MS1	Can a post-hoc merge without old data (buffered or new-data-only distillation, weight averaging with or without alignment) recover the oracle merge's gain?	Local MPS (CPU fallback)	Pre-registered, queued
MS7	Can a simple model on logged edit features predict $\Delta U > 0$ out of seed better than the best single diagnostic?	Local CPU	Pre-registered, queued (last step)
MS5	Does the risk term predict single-merge damage in OLMoE-1B-7B?	Local MPS	Pre-registered (design freeze); not queued
MS6	Do risk-gated consolidation rules transfer to a continual Qwen2.5-1.5B LoRA-MoE?	Local MPS	Pre-registered (design freeze); not queued

The queue runs MS0 merge → MS8 pilot → MS0 split → MS2 offline → MS8 → MS3 → MS4 → MS1 → MS2 trained → MS7, with MS5 and MS6 run manually behind their gates. This is the report's order with two changes: MS8 is inserted high, and MS4 runs before MS1 because MS4 is CPU-only on cached

features while MS1 needs the GPU.

Every edit is scored against the original routed arrangement under the same router, never against either expert alone, with acc = A_sel × A_within and $\varepsilon = 1 - \text{A_sel}$ reported throughout. An edit is accepted only on a utility rule $\Delta U = \Delta Q + \lambda \Delta C - \mu R$ (change in correctness, memory change as a fraction of experts, negative flips) whose 95% bootstrap lower bound must exceed 0 on one fixed half of the test set, with negative flips ≤ 0.01 and no edited class losing more than 0.20; realised utility is read on the other half. Seed is the unit (n = 10), with 10,000-resample percentile bootstraps, one-sided tests in the predicted direction, Holm correction within each experiment's family, fixed kill criteria and a deviation log.

The results will drive five decisions. Router first: MS0 H0c, MS4 H4b and MS8 H8d test whether structural edits are mainly a selection fix. Consolidate confused pairs: MS0 H0a/H0b plus MS1 H1b decide whether confusion-driven consolidation with buffered distillation becomes the default; if MS1's kill fires, merges must be decided before experts are frozen. Where to cut: MS8 decides whether expert boundaries should be learned, inherited from human categories or left arbitrary. Splits: MS0 and MS2 decide whether splits are ever worth it at this scale, and which trigger proposes them. LLM transfer: MS3, then MS5 and MS6, decide whether REAP's risk term is a usable merge gate in shared-output MoEs.

MS0 results (pre-registered, run once on test seeds 0–9)

MS0 edits one expert or one pair of a frozen T = 10 Split-CIFAR-100 library, using the benchmark's nested T = 5/10/20 libraries (each merged or split expert is retrained from scratch on the union or half of the data). It scores every edit against the original routed library, not against either parent, under four routers. Baseline system accuracy is 0.332 under max-softmax and 0.671 under a confidence-free RanPAC router on ViT features (ε 0.178).

Hypothesis	Result	Verdict
H0a: one consolidation helps under self-report routing	Δacc +0.0058 [+0.0033, +0.0084], 35 of 49 events positive	Supported
H0b: pre-merge router confusion predicts the gain ($\rho \geq 0.30$)	ρ +0.645 [+0.518, +0.790]; feature CKA +0.02	Supported
H0c: the merge gain shrinks once the router is fixed	interaction −0.0125; under RanPAC-ViT a merge costs −0.0068	Supported
H0f: fusion (keep parents, add the union expert) pays a selection price	fusion +0.0115 vs consolidation +0.0058 under max-softmax	Falsified
H0d: one split hurts under self-report routing	Δacc −0.0171 [−0.0197, −0.0150], 99 of 100 negative	Supported
H0e: a fixed router makes some splits pay	79% of splits positive under RanPAC-ViT vs 1% under max-softmax	Supported

Three readings follow. First, merging and splitting are selection operations whose sign depends on the

router: behind weak self-report routing, merge confused pairs and do not split; behind a strong shared-space router, the signs flip. Second, the merge signal is router confusion, not representation similarity. Third, consolidation's mean gain hid class-level losses (worst class −0.434, negative flips 0.036), while fusion gained more with almost none (−0.001, 0.007); keeping the parents next to the merged expert was the safer edit. Among split triggers only task-given error tracked a split's value; multimodality tests and load did not. The pre-registered utility rule accepted almost nothing and needs loosening before MS4. Effects per edit are small (0.5–1.7 points) but consistent over ten seeds; the fusion and sign-flip mechanisms are interpretations, not tests.

Published by NeuroGenesys (AW3 Technology, Inc.). Canonical version: https://neurogenesys.si/research/merging-splitting-moe-experts