Route on shared statistics, never expert confidence: the router architecture for frozen experts

W. Schulz

Nov 12, 2026

Abstract

A literature synthesis and experiment plan on the router that picks which frozen expert, model, skill or generation handles each input. Theory, MoE gating, LLM routers, continual-learning task inference and the lab's own results agree that a router should not read the experts' own confidence; the best-supported design scores experts in one shared feature space with a closed-form rule and abstains on its own margin. Pre-registered tests on a cached Split-CIFAR-100 library of 30 expert sets then confirmed five of six predictions. A shared space beat self-report, slightly without pretraining and by 45 points of selection error with ImageNet ViT features, and it held selection error nearly flat from 5 to 20 experts. Margin-only abstention did not transfer, and a direct readout of the shared features beat routing to the experts by 13 points.

W. Schulz, NeuroGenesys, November 2026. A literature synthesis and experiment plan, updated 12 November 2026 with the pre-registered results of R0–R5 (see Results). Cited numbers are the sources' own; the lab's results carry their ledger IDs, and results outside the ledger are labelled. Several sources are 2025–26 preprints, marked as such. R6 (LLM skill routing) is queued.

The best-supported router for NeuroGenesys does not ask experts whether an input belongs to them. It scores every candidate expert in one shared, fixed feature space with a closed-form, confidence-free rule (random-feature ridge prototypes in the RanPAC style, or shrunk Mahalanobis). It takes its contrast signal from accumulated statistics of the other experts' data, not from a penalty term. It picks one expert per input by default, and combines outputs only inside a shortlist that a confidence-free stage has already produced. It abstains on its own top-1 minus top-2 margin. It admits new experts by adding their statistics. It is judged on held-out routing probes by an evaluator independent of both router and experts. That design rests on three kinds of evidence. Theory says the Bayes-optimal router routes to the expert with the highest expected correctness, and that confidence-based deferral is provably suboptimal when the experts are specialists. Literature results from continual learning, LLM routing and MoE gating all reward shared-space statistics and penalise readouts of expert self-report. The lab's own record is the third leg. On Split-CIFAR-100, frozen experts lose almost all their accuracy to selection: selection error ε rose from 0.534 to 0.788 as tasks went from 5 to 20, while within-task accuracy rose (ledger 8.18). On Split-MNIST, a confidence-free RanPAC selector cut ε from 0.213 to 0.067 (9.1). The critical caveat is that the confidence-free win has been measured only on MNIST. CIFAR-100 shows only which readouts fail. Hierarchy, top-k mixing, replay-trained routers and new-expert admission have no lab measurement at all. The seven experiments proposed below (R0–R6) would settle the design in roughly 10 GPU-hours plus a day of offline fitting, because one caching run (R0) turns every later router comparison into a cheap offline fit. Update, 12 November 2026: R0–R5 have run. On Split-CIFAR-100 they support the shared space and the confidence-free rule, and they do not support margin-only abstention. With a strong shared space, a direct readout beat the routed experts by 13 points (see Results).

Selection error sets the ceiling, and theory says what the router must estimate

The premise behind this question, that frozen experts make selection the dominant loss, is confirmed at L1 scale and only partly holds for LLM skills. In the pre-registered Split-CIFAR-100 study (10 seeds, SlimResNet18 per-task experts, frozen), selection error exceeded within-task error by +0.604 [+0.598, +0.610] at $T = 10$. Selection carried $S = 0.91$ of between-method variance in log accuracy across Naive, GDumb, DER++ and BiC (ledger 8.15, confirmatory; 0.68 without Naive, 8.17 post hoc) (lib/findings.ts; results.md). With max-softmax selection, ε went $0.534 \to 0.644 \to 0.788$ at $T = 5/10/20$, within-task accuracy went $0.905 \to 0.961 \to 0.981$, and system accuracy fell $0.422 \to 0.342 \to 0.208$. The whole decline is routing (8.18, Holm-adjusted $p < 0.0001$). The study's README draws the design conclusion directly: "the selector, not the frozen-expert design, is what limits the approach" (README). Even jointly trained Joint, the strongest shared-head learner, only reaches an implicit task-selection accuracy of 0.576 at $T = 10$. Part of the error is therefore genuine confusability between CIFAR-100 tasks (interim analysis).

LLM skill routing is a different regime. With Llama 3.1 8B routing in context among 64 synthetic skills, ε was only 0.030 and explained 6% of the routed accuracy drop (8.1). Router quality there tracked model size far more than documentation quality: ε was 0.170 for Llama 3.2 3B, 0.030 for 8B and 0.003 for Qwen3-VL 30B-A3B (8.10). Richer skill descriptions left 8B ε unchanged at 12/400 and only moved which pairs were confused (8.11). Scale is what turns routing into the binding constraint. The lab's L3 ledger entry records that wrong-skill selection explains about 38%, 57% and 67% of the pass-rate drop at 52, 102 and 202 skills on frontier models (Song & Wei 2026, 8.5). In the lab's own lexical ToolRet benchmark (pre-registered, run once, not in the ledger), BM25 ε@1 rose from 18.1% at 10 candidates to 78.2% at the full 44,453-tool corpus. The fitted slope was logit $\varepsilon = a + 0.341 \cdot \ln K$ (R^2 0.995) (ToolRet README). Across every lab testbed, ε grows with the number of candidates. The router question is therefore about scaling: what keeps ε flat as the library grows.

Theory says exactly what the router has to estimate. For frozen experts, the two-stage multi-expert learning-to-defer Bayes rule picks the expert with the lowest expected cost, $\text{argmin}_j \Sigma_y p(y|x) \cdot c_j(x,y)$. Mao, Mohri and Zhong prove realizable H-consistency and Bayes-consistency for this rule (ICML 2025) (arXiv 2506.20650; PMLR). When experts own disjoint label sets, a wrong-task expert is never right, and the rule reduces to route to $\text{argmax}_t P(T = t \mid x) \cdot P(E_t \text{ correct} \mid x, T = t)$. That is a task posterior weighted by in-task competence. Cross-expert confidences do not enter it. This decomposition is the research notes' derivation, not a published theorem, but it is why the lab's identity $\text{acc} = A_{\text{sel}} \times A_{\text{within}}$ holds exactly. Two further results explain why the obvious estimator fails. Jitkrittum et al. (NeurIPS 2023) show that confidence-based deferral is notably beaten by post-hoc rules when downstream models are specialists that do well only on a subset of inputs (arXiv 2307.02764, abstract only). Per-task experts are that case. Mozannar & Sontag (ICML 2020) give a consistent surrogate that beat confidence deferral by 7.89 points on CIFAR-10. They also show the older Madras-style mixture-of-experts loss is inconsistent and "learns never to defer" (arXiv 2006.01862). The theory leaves one hard practical problem: every consistent surrogate needs labelled samples of each expert's correctness drawn from the deployment mixture. In exemplar-free continual learning, that cross-task data is exactly what the system lacks.

A shared feature space and a confidence-free score beat every readout of self-report

The most consistent finding across all four evidence streams is negative, and it is well established: expert self-report is the wrong routing signal across separately trained experts. In the lab's MNIST campaign, experts were calibrated on their own data (own-task ECE 0.011) and badly miscalibrated on mixed inputs (0.697). Only 34% of raw inputs scored ≥ 0.95 belonged to the scoring expert's task. A shared

temperature left ε at 0.367 → 0.368, because rescaling cannot change an argmax (9.2) (lib/findings.ts; selector-abstention paper). Making experts more diverse did not move ε (0.388 → 0.397). Consensus over 1 to 15 experts per task saturated at N* ≈ 2 (SE1, SE3, Split-MNIST, run once, not in the ledger) (forgetting-as-selection paper). On CIFAR-100, energy improved on max-softmax by only 0.020 ε at T = 10. A pixel autoencoder gate was worst at every T and raised ε by 0.145. Both are pre-registered secondary analyses with unadjusted CIs, recorded in the interim analysis rather than as separate ledger entries (interim analysis). That 0.020 gain is small next to a 0.604 selection gap. Changing which self-report you read barely helps. Jaeger et al. (ICLR 2023) found plain max-softmax the best *within-model* failure detector across a broad benchmark (arXiv 2211.15259). That does not contradict the lab: MSP ranks one model's own errors well, but its scale is not comparable across experts trained on different data. The practical split is to use MSP inside an expert, to decide keep or abstain, and never across experts to pick the argmax.

The positive evidence is that a single shared space with a closed-form scorer works, but at the lab's own scale it has been shown only on MNIST. On streaming-PCA features, the confidence-free selectors ranked RanPAC (random ReLU expansion, M = 2,000, plus ridge) at ACC 0.926, ε 0.067, then shrunk Mahalanobis at 0.892, nearest-class-mean at 0.857, and the expert-confidence selector at 0.786. ε over T = 2..5 rose only 0.016 → 0.067 for RanPAC, against 0.071 → 0.213 for MSP (9.1). Letting the router pick the task and the expert pick the class came within 0.006 of the router's own 100-way readout. The experts therefore stay useful for within-task prediction even when they are useless for routing. Two caveats travel with 9.1: the PCA basis was fit on the whole unlabelled stream, and ridge λ was chosen on a labelled calibration split. The campaign also ran about 40 arms once, without multiplicity correction. The feature-space result is starker still. Moving experts onto shared streaming-PCA features cut ε from 0.40 to 0.279, and plain global PCA-NCM (0.871) beat explicit per-task experts plus consensus (0.788). By contrast, distance to an expert's own training centroid *in that expert's own ReLU space* anti-selected at ε 0.963 (SE5, SE2, not in the ledger). A post-hoc "not mine" head on frozen expert features was also worse than a shared space, reaching ε 0.303 at 100 images per task (9.5).

The literature points the same way at CIFAR and LLM scale. RanPAC reaches 92.2% on CIFAR-100, against 84.1–86.3% for prompt-selection methods. HiDe-Prompt's task-identity inference tops out at 67.74% (ledger 8.3, L3). FeCAM's shrunk Mahalanobis stays nearly flat from T = 5 to 20 (70.9 → 69.4), while NCM falls 64.8 → 61.5 (L3, via the lab review, not in the ledger) (Stopping confident wrong expert votes). Newer methods that check whether an expert *owns* the predicted class, rather than how confident it is, reach high task-ID accuracy. MoTE's class-scope filter gives 89.43% task identity on CIFAR-100 (arXiv preprint, 2506.11038). DPTA reports that raw frozen-ViT prototypes give top-5 class accuracy of 98–100% even where top-1 is far lower (arXiv preprint; figure caption only, 2411.17766). CPrompt's key matching reaches only 19.44% task accuracy with one key per task on fine-grained StanfordCars-10, and 39.83% with per-class keys (CVPR 2024, arXiv 2403.08568). At model level, encoder choice barely matters. LLMRouterBench found "no significant" difference when it replaced gte-Qwen2-7B with all-MiniLM-L6 (arXiv 2601.07206; the lab lists it as ICLR 2026, venue not re-verified) (arXiv 2601.07206). A kNN router with k = 100 over frozen BERT embeddings came within about 0.5 AUC of the best learned router on fixed pools and lost the least under shift (2.63 AUC) (arXiv 2505.12601, preprint). MoE gates warn against the opposite approach. Mixtral's token routing shows almost no domain specialisation and follows syntax (Mixtral), and forcing semantic clustering in hash routing worsened perplexity from 23.22 to 23.90 (Hash Layers). A per-token gate is no evidence that a learned gate identifies domains. Selecting whole experts should use sequence- or query-level representations.

Two counterweights keep this from being a closed case. First, on fresh multi-domain queries in RouterArena (ICLR 2026 per its listing), kNN and MLP routers trained on RouterBench reached only 58.7%

and 61.6% accuracy, against 66.9% for the IRT-structured MIRT-BERT at about 1/28 of kNN's cost (arXiv 2510.00202). That gap probably reflects mismatched training data and cost terms rather than a structural weakness of kNN, but a factorised P(correct | query, model) predictor clearly earns a place once experts overlap in competence. Second, the lab's MNIST win used a feature space with no hard representation problem. On CIFAR-100 without pretraining, the shared space may itself be the bottleneck: Joint's A_sel of 0.576 hints at that. The open question is whether feature quality, not the scoring rule, binds at realistic scale. Experiment R1 below is built to answer it.

The contrast signal must come from other experts' data; balancing is only a guard

Theory says a router needs cross-task correctness labels. The lab's record shows how to get the information they carry without storing data. A penalty trained only on an expert's own data was inert or harmful: a Deep Gamblers abstain output changed ε by -2% at $t = 0.5$ and $+54\%$ at $t = 0.75$ (9.4). Rotated or flipped own-task images as a synthetic third class helped raw pixels by +12.7 points but helped PCA features by only +2.98, just under the kill threshold of 3.00 (9.6). Real other-task data helped substantially. A "not mine" output trained on a past-task buffer raised raw ACC from 0.633 to 0.871 at 100 images per task. Its kill criterion still fired, because ε never fell below buffer-free RanPAC's 0.067 (9.5). Per-expert affine calibration on a 200-per-task buffer raised max-logit and energy selection by 5.4 points but lowered MSP (9.7). The conclusion is that the contrast must involve other tasks' statistics. On MNIST, the cheapest form, accumulated class means and Gram matrices that need no stored images, beat a replay-trained discriminative router at the memory tested. The literature agrees that cross-task contrast is the active ingredient. Kim et al.'s factorization bounds class-incremental loss by within-task loss plus task-prediction loss. The selectors that beat replay on CIFAR-100 (HAT+CSI 63.3, TPL 76.53) use replay data or synthetic negatives (8.2, L3). CPrompt's Classifier Consistency Learning trains each new prompt to look unconfident to old heads, adding +3.36 Last-accuracy with the correct prompt given (arXiv 2403.08568). Whether a replay-trained router overtakes accumulated statistics at CIFAR scale, as TPL's numbers suggest it might, is untested (R2).

Among model-level routers, the training signal matters far more than the architecture, and this finding is well supported. In RouteLLM (ICLR 2025), four architectures trained on Arena preference labels alone were near random. MT-Bench APGR ran 0.39–0.61, and GSM8K came in below random. With LLM-judge-augmented data, matrix factorisation reached 0.802, and the winning architecture changed by benchmark (arXiv 2406.18665). Near-ties are the specific failure. About 94.9% of RouterBench queries have a near-zero top-2 margin, so scalar-score routers drift toward the strongest model as budget grows. A pairwise ranking loss fixes most of it: EquiRouter reaches GPT-4-level performance at about 77% of cost, and removing the ranking loss erases most of the gain (arXiv 2602.03478, preprint, read via the Hugging Face papers page) (HF papers). RouterEval (Findings of EMNLP 2025) reports that existing routers mostly sit at 0.95–1.04× the best single model, against an oracle of 1.34–2.00×, and that classification-style routers collapse onto a few candidates (arXiv 2503.10657). The lab's own literature log records the same LLMRouterBench conclusion: recent routers do not outperform "Best Single" in the performance setting (Orchestrated specialists).

Load balancing is where MoE practice transfers least. Co-trained MoE layers need balancing to avoid collapse. Without a balancing loss, OLMoE's first-layer tokens collapsed onto one expert (OLMoE). DeepSeek's auxiliary-loss-free bias controller holds balance far better than a loss term, MaxVio 0.04 against 0.52–0.72, at slightly lower perplexity (9.50 vs 9.56 at 1B) (arXiv 2408.15664; numbers verified in text, and the paper itself reports internal inconsistencies in token counts) (Wang et al.). But selection among separately trained experts differs, because those experts are not equal. In Branch-Train-MiX, routing among domain experts without balancing left the Code expert "effectively dead" while the av-

erage score barely moved (34.6 vs 34.7). Balancing shifted per-task results instead: GSM8K 29.8 → 34.6 without it, HumanEval 27.4 → 19.5 (BTX). Equal usage is the wrong target when competence differs. What transfers is the *mechanism*: a non-gradient per-expert additive prior adjusted from observed selection rates, used as an anti-collapse guard rather than a usage quota. The *metric* also transfers: report the most-selected expert's share alongside accuracy, to detect the collapse RouterEval and EquiRouter describe. No source tested bias balancing over heterogeneous-quality experts, so this component is a design inference. Two further MoE lessons transfer cleanly. Routers converge early (60–80% overlap with the final assignment after 1–40% of OLMoE training) (OLMoE), and StableMoE distils then freezes the router to stop routing fluctuation (arXiv 2204.08396, abstract only). Both support a frozen or closed-form router over a continually retrained one. The z-loss stability claim in ST-MoE is unverified from recall. OLMoE independently confirms that z-loss reduces loss spikes at about 2% throughput cost.

Choose one expert, combine only inside a verified shortlist, and abstain on the router's margin

The evidence on combination versus choice splits cleanly by whether experts share a label space. When per-task experts own disjoint classes, the Bayes rule is argmax, and averaging only adds probability mass from wrong-task heads. TSBN reports both of its metrics declining slightly as soft aggregation goes from $k = 1$ to $k = 5$ task heads, because "additional task heads tend to introduce noise from irrelevant tasks" (arXiv 2411.00430, preprint, figure only) (arXiv 2411.00430). The label structure also makes plurality voting across such experts degenerate: two experts from different tasks never vote for the same class. The lab's Best-of-N result, plurality over the top-scored half beating argmax by +0.03 to +0.17 in all 6 math cells (8.14), therefore cannot transfer directly to class-incremental routing. Votes from one exploitable signal are also correlated. Experts of the same wrong task agree on which inputs they steal (phi 0.77, 9.10), and more voters saturate at $N^* \approx 2$ (SE3, not in the ledger). Where combination does pay, it is restricted to a candidate set that a confidence-free stage has already produced. MoTE filters by class scope, then fuses the survivors, and beats unfiltered dense combination on all five datasets: VTAB average accuracy goes 88.99 → 93.16. DPTA uses prototype top-K to load only K adapters (MoTE; DPTA, both preprints). When experts overlap in competence, as LLMs do, set-valued routing helps. RACER's conformal model sets with a null model add +1.2 to +3.6 points over single-choice base routers and reach 62.1% against 57.1% for the best single model, with empirical risk at or below $\alpha = 0.1$ in 100 trials (arXiv 2603.06616, preprint) (arXiv 2603.06616). BTX's top-2 (34.6) and soft routing (35.8) over top-1 Switch routing (24.7) is often cited for combination. It concerns token-level gates inside one model, so it supports soft mixing for overlapping, co-fine-tuned experts rather than for disjoint frozen ones (BTX). One lab lever dominates all of these when deployment allows it: pooling over a same-task session. Density-selector ε fell 0.095 → 0.011 → 0.002 → 0.000 at batch sizes 1, 4, 8 and 16 (9.9). This depends on a deployment guarantee, not on router design.

Abstention has the cleanest lab result: it works only on a score independent of the experts. On MNIST, a density/FeCAM margin reached AURC 0.025 against 0.098 for MSP, and selective risk at 70% coverage was 0.023 against 0.123 (9.3). In the QI succession toy, filtering by self-confidence lowered accuracy from 0.759 to 0.725 while naive succession held at 0.761 (not in the ledger) (GCCL lab record). Theory supplies the threshold. Chow's rule ($t^* = \lambda/(1+\lambda)$) and the Mozannar–Sontag rejector both defer when the best option's expected correctness falls below the cost-adjusted alternative. MoE practice supplies the analogue of variable k. Confidence-threshold routing gains +0.7% over top-2 with under 90% of activated parameters (ACL 2024), and AdaMoE's null experts cut Mixtral FLOPs by 14.5% while raising ARC-C accuracy by 1.69% (both abstract-level) (arXiv 2403.07652; arXiv 2406.13233). A low router margin should send the input to a fallback, such as top-k mixing within the shortlist, a shared 100-way head, or an escalation model, rather than to a hard refusal. The lab has not measured which fallback

pays (R4). Evaluation matters here too. Traub et al. (NeurIPS 2024) show AURC is ill-suited to aggregating over thresholds and propose AUGRC, which changed metric rankings on 5 of 6 datasets (arXiv 2407.01032). The lab's SA2 result should be re-checked on AUGRC before it is generalised.

Hierarchy is the least supported component. The literature evidence is suggestive but indirect. CaRE's bi-level router keeps final-layer top-2 router recall at 92.3% → 85.1% → 80.6% from 10 to 301 tasks, and its single-router ablation loses 24 Last-accuracy points at 100 tasks, 68.27 → 43.96 (arXiv 2602.03473, preprint) (arXiv 2602.03473). At model level, Avengers-Pro's training-free clustering, which picks the best model per query cluster, nearly dominates LLMRouterBench's cost–performance frontier (arXiv 2601.07206). No study compares hierarchical and flat routing on the same experts, and the lab has no measurement. The research notes add a structural argument. Over max-type scores (MSP, energy), a group-then-expert cascade whose group score is the max over members *is* the flat argmax. Hierarchy can only help if level one uses a better-estimated score, such as pooled group density, than the flat router does. Hierarchy's clear benefit is cost. A prototype top-K shortlist keeps expert forward passes at O(K) rather than O(T). That is the design reason to build one, independent of any accuracy gain. In the research notes' extrapolation from three methods, which is not a law, ε grows roughly with ln T in good shared features and much faster under fine-grained key matching. The lab's ToolRet slope ($b \approx 0.34$ per ln K, BM25) is the only fitted version of that relation the lab has.

New experts and independent evaluation turn a router into an institution

Admitting a new expert favours statistics-keyed routers by construction. A prototype or Gram-matrix router admits an expert by adding that expert's class statistics. Old-task statistics do not change and nothing is retrained. A learned router must be retrained, and MoE continual-learning papers name the failure mode: routing drift, where the expanded router sends old-task inputs to new experts even when old experts are frozen. Proposed fixes include zero-initialised router columns, frozen old routers, and training new experts only on inputs that clearly differ from earlier tasks. These come from search snippets and an aggregator page, not opened papers (LLaVA-DyMoE, StaR-MoE, CMoE-Adapter, LEGO) (LLaVA-DyMoE; StaR-MoE). At model level, cold start works best when each model has a *representation* rather than a per-model head. IRT-Router predicted a held-out Claude 3.5 Haiku from a text profile at 0.67 accuracy against 0.34 for a RouterBench-style predictor (ACL 2025) (arXiv 2506.01048). RouteLLM routers kept 0.74–0.77 APGR when swapped to unseen model pairs, because they learn query difficulty (arXiv 2406.18665). The lab's L3 review records that self-keyed LoRA routers (PHATGOOSE, Arrow, LoraRetriever) tolerate additions. It also records that no study holds a router fixed while an existing expert changes, and that PHATGOOSE's post-hoc routers fall further behind the oracle from 36 to 166 experts (Orchestrated specialists). The lab has measured neither "stealing" by a new expert nor router cold start with few samples. For admission and retirement, the graded-succession design already in this programme applies unchanged. A new expert or model generation starts at zero authority and gains traffic per context only as a paired e-process on held-out outcomes crosses pre-set thresholds. That design is a proposal, tested by its own experiments S1–S5 (Graded succession by earned authority). The router's contribution is to make admission cheap: statistics in, an admission gate on old-task ε, and nothing retrained.

Independent evaluation is the component every source agrees on, and the one the lab is best tooled for. The graded-succession report requires that "the evaluator must be independent of both generations". It cites the lab's finding that weighting whole sources by reliability backfired while per-example weighting fixed it (2.3) (Graded succession by earned authority). For routers, that means ε is measured on labelled routing probes disjoint from everything used to fit or calibrate the router. A same-context critic missed 12/12 planted errors while a fresh-context critic missed 0/12 ($n = 24$ micro-pilot, not in

the ledger), which illustrates why that disjointness matters (orchestrated lab record). Router papers add the baselines an independent evaluator must always run. The hash baseline: fixed random hashing matched or beat learned Switch routing at 64–128 experts (Reddit perplexity 22.95 vs 23.58 at 1×128), so a learned router has to beat a non-learned one before it earns credit (Hash Layers). Best Single and kNN (k ≈ 100), from the model-router benchmarks above. The strongest model's call share, to detect collapse. ID, shifted and perturbed splits, since RouterArena found BERT-based routers' selections unstable under surface perturbation (arXiv 2510.00202). The lab's `lib/reliability` module already provides Wilson and Newcombe intervals, prediction-powered inference, confidence sequences and an ALLOW/BLOCK/NEED-MORE-DATA gate. Its `selection/v1` event schema covers router, adapter and model selectors (stats.ts). None of it has yet been applied to a router study.

Which components are evidence-backed and which are design hypotheses

Component	Recommended choice	Status	Strongest support	What would overturn it
Scoring signal	Not expert confidence across experts; keep MSP inside an expert	Evidence-backed (lab + theory + literature)	8.15–8.18, 9.2, 9.3; Jitkrittum 2023; Mozannar & Sontag 2020	—
Feature space	One shared, fixed space (pretrained or stream-legal), query/sequence-level	Evidence-backed on MNIST and in literature; hypothesis at CIFAR scale in the lab	9.1, SE5 (not in ledger); RanPAC/FeCAM (L3); LLMRouterBench encoder swap	R1 kill K1
Scoring rule	RanPAC ridge prototypes; shrunk Mahalanobis runner-up; factorised P(correct) when competence overlaps	Lab MNIST evidence; ranking at scale is a hypothesis	9.1; RouterArena IRT; kNN preprint	R1
Training signal	Accumulated other-task statistics; buffer router only if it wins at matched memory; ranking loss for overlapping models	Mixed: contrast-is-necessary is evidence-backed; buffer-free-wins is MNIST-only	9.4, 9.5, 9.7; RouteLLM; EquiRouter (preprint)	R2 H2b

Component	Recommended choice	Status	Strongest support	What would overturn it
Load balancing	Non-gradient per-expert prior as anti-collapse guard; report top-expert share; no equal-usage target	Design inference from MoE	Loss-free balancing (preprint); BTX	Untested anywhere on heterogeneous experts
Combination	Argmax by default; combine only within a confidence-free shortlist or for overlapping experts; session pooling where guaranteed	Partly evidence-backed (TSBN, MoTE, RACER preprints; 9.9); top-k soft mixing untested in lab	9.9; MoTE; RACER	R3
Abstention	Threshold on router's own top-1 − top-2 margin; fall back rather than refuse	Lab MNIST evidence + theory	9.3; Chow; L2D	R4 kill
Hierarchy	Prototype top-K shortlist for cost; accuracy benefit unproven	Hypothesis	CaRE (preprint); Avengers-Pro	R5 H5a
New experts	Add statistics; gate admission on held-out old-task ε; graded authority	Hypothesis (literature L3, unverified snippets)	IRT cold start; graded-succession design	R5 H5b–c, R6 admission arm
Independent evaluation	Disjoint labelled probes, intervals, hash/Best-Single/kNN baselines, shift splits	Evidence-backed as practice; tooling exists, unused on routers	Hash Layers; RouterArena; lib/reliability	—

Seven proposed experiments would identify the router in about ten GPU-hours

Everything in this section is a proposal, taken from the lab-record research notes (lab record and experiments). Thresholds are placeholders to be fixed at pre-registration. Compute figures are estimates from measured run times unless marked unmeasured. One engineering fact drives the plan: the CIFAR harness saved no checkpoints, logits or features. `run.py` writes only a JSON summary, and the cache directory holds only data and the venv. Every new CIFAR selector therefore needs the experts retrained. R0 does this once, and R1–R5 become offline fits on cached tensors. The common design copies the

Phase 1 discipline: test split never used for tuning, pairing by seed, 10,000-resample bootstrap CIs with seed as the unit, Holm within each primary family, kill criteria and a deviation log (preregistration). Each task's training data is split 90/10. Experts train on the 90%, and the 10% is the only labelled pool a learned router may use. Max-softmax and energy are re-measured within each run as controls, because MPS non-determinism means a retrained expert set will not reproduce 8.15–8.18 bit-for-bit ("single-seed differences under ~6 pp at this scale are noise", 7.8). Every arm writes `selection/v1` events. Confirmatory seeds 10–19 avoid reusing class orders already examined. Seeds 0–9 would allow per-seed pairing with the ledger instead, and that choice is left to the freeze. Seed 100 on the validation split pilots and freezes hyperparameters.

ID	Question	Prerequisites	Cost (estimate)	Primary hypothesis and kill rule
R0	Cache the expert library (engineering)	New `phase2/code/cache_experts.py` importing the frozen `run.py` unchanged; storage under `~/.cache/neurogenesis-cl/experts/`, never `/tmp`	≈ 9.7 h MPS at measured means (≈ 5.2 h at minima) for 30 (T, seed) runs; ≈ 7 GB float16 plus disk-space checkpoints; ViT-B/16 extraction over 60k images unmeasured	Sanity gate: within-run max-softmax ε at T = 10 within the 8.18 seed range; > 0.03 off means stop and log a deviation

ID	Question	Prerequisites	Cost (estimate)	Primary hypothesis and kill rule
R1	Does a confidence-free router beat self-report on CIFAR-100, with a flatter $\varepsilon(T)$?	R0; cached ViT-B/16 weights (on disk); first-session backbone from R0	Offline; plausibly a few CPU hours for all arms (unmeasured; $M = 10{,}000$ needs a $10k \times 10k$ Gram solve)	H1a stream-legal features (streaming PCA, first-session backbone) beat energy ε at $T = 10$; H1b same with ImageNet ViT, reported separately because of class overlap; H1c smaller $\varepsilon(20) - \varepsilon(5)$ slope than energy's 0.287. Kill K1: no stream-legal router beats energy, so feature quality binds; K2: no slope reduction, so the hierarchy rationale weakens

ID	Question	Prerequisites	Cost (estimate)	Primary hypothesis and kill rule
R2	Does a replay-trained router beat OOD scores and buffer-free statistics?	R0, R1 best feature space	Minutes per (T, seed)	H2a buffer router (M_tot = 2,000, matching 8.16 baselines) beats energy; H2b it does not beat buffer-free RanPAC (replicating 9.5's kill); H2c a stacker on all experts' logits beats fixed readouts, which would show the information is present but badly read. Kill: no buffer router beats energy, so drop replay routers; stacker ties MSP, so self-report must be replaced, not fixed
R3	Does top-k soft mixing or scorer plurality beat argmax?	R1 best router; posterior temperature fit on 10% pool	Seconds per arm	H3a top-2 soft mixing weighted by router posterior beats argmax at T = 10 (negative control: weights from expert MSP); H3b five-scorer plurality does not beat the best single scorer, with pairwise error phi measured. Kill: no k beats argmax, so router top-1 is the whole story

ID	Question	Prerequisites	Cost (estimate)	Primary hypothesis and kill rule
R4	Does margin abstention beat MSP abstention, and which fallback pays?	R1, R3	Seconds per arm	H4a router-margin AURC $\leq 0.9 \times$ expert-MSP AURC; H4b routing the lowest-margin 10–30% to a fallback (top-k, 100-way head, plurality) raises full-coverage accuracy; report AUGRC and realised split-conformal coverage. Kill: MSP AURC $\leq 1.1 \times$ margin AURC, so the independence principle fails at scale

ID	Question	Prerequisites	Cost (estimate)	Primary hypothesis and kill rule
R5	Does hierarchy flatten ε(T); does a fixed router absorb new experts without harming old ones?	R0–R2; optional extra R0 runs for T = 50	Minutes per (T, seed); T = 50 adds 10 runs at unmeasured cost (probably ~1,000–1,500 s each)	H5a two-level router (k-means groups, G ∈ {4, 5}) beats flat at T = 20, with MSP/energy as identity checks; H5b incremental-statistics router keeps old-task ε within a pre-set margin as tasks arrive, while a retrained buffer router drifts more; H5c cold start at n ∈ {5, 20, 100} per class, measuring stealing in and out separately; exploratory `lib/reliability` admission gate. Kill: hierarchy no better than flat, so drop it; old-task ε exceeds margin, so "new experts are free" is false

ID	Question	Prerequisites	Cost (estimate)	Primary hypothesis and kill rule
R6	Does an embedding kNN router beat in-context LLM routing at $K = 64$?	Embedding model pulled (e.g. `nomic-embed-text`) or sentence-transformers in a separate venv; neither is on disk	A few Ollama hours; $\geq$ 1,000 fresh requests per arm sized by `planSampleSize`; only mis-routes are re-executed	H6a kNN over 20 exemplars per skill beats 8B's 0.030; H6b holds on a held-out-template arm (each skill has about two templates, so this guards against leakage); H6c embedding top-8 then 8B matches full in-context routing at a fraction of the prompt; library growth $32 \to 64$ measures old-skill ε after admission. Also run ToolRet's skipped dense arm. Kill: no gain on held-out templates, so in-context is adequate at $K = 64$ for $\geq$ 8B and the question moves to $K \gg 64$

The proposed order is R0 → R1 → R4 → R2 → R3 → R5, with R6 in parallel on Ollama. The total is about 10 h of MPS time for R0 plus unmeasured ViT extraction, under a day of CPU for R1–R5 (unmeasured), and a few Ollama hours. That sits well inside the 72-hour cap the Phase 1 study used. The decision logic is fixed in advance. If R1 (H1a or H1b), R4 (H4a) and R2 (H2b) hold, the record supports a buffer-free, confidence-free, shared-feature router with margin abstention as the default. R3 and R5 then decide whether top-k mixing and hierarchy are added. If K1 fires, the binding component is representation, not routing rule, and the next study is about features. Three known limits apply before any freeze. ImageNet pretraining overlaps CIFAR-100 classes, so H1b is not a clean test of the buffer-free claim. R6's synthetic skills cannot show transfer to natural requests, so ToolRet's real queries are the check for that. And no novelty gate has been run against RanPAC, FeCAM, TPL, HiDe-Prompt or the LLM-router literature. The graded-succession report assigns that check to `docs/NOVELTY-GATE.md`, a file not yet present in the repo.

Results: a shared space beats self-report, but with a strong shared space the experts are the bottleneck

R0 and R1–R5 ran as pre-registered (`preregistration.md`, v1.0, frozen before any test data; no deviations to R1–R5). One SlimResNet18 per task was trained on 90% of each task's data, for $T \in \{5, 10, 20\}$ and seeds 0–9: 30 runs, all exiting cleanly, about 16 hours on MPS. Every router was then fitted offline on 250 cached training images per class and scored once on the test split. Differences are paired by seed, with 95% bootstrap CIs over seeds (10,000 resamples) and Holm across H1–H6. The sanity gate passed: within-run max-softmax ε at $T = 10$ was 0.654, against the ledger's 0.644 (8.18), inside the pre-set 0.05.

H ($T = 10$ unless stated)	Prediction	Result	95% CI	Holm p	Verdict	Pilot predicted
H1	Stream-legal `ranpac/first` has lower ε than `energy`	0.614 vs 0.627, Δ +0.014	[+0.003, +0.027]	0.003	supported, small	fail (tie)
H2	ImageNet-ViT `ranpac/vit` has lower ε than `energy`	0.178 vs 0.627, Δ +0.450	[+0.440, +0.460]	< 0.001	supported	hold
H3	ε grows less from $T = 5$ to 20 for `ranpac/vit` than for `energy`	+0.030 vs +0.257, Δ +0.227	[+0.213, +0.239]	< 0.001	supported	—
H4	Router-margin abstention has lower AURC than chosen-expert max-softmax	0.158 vs 0.153, Δ −0.006	[−0.012, −0.000]	0.98	not supported (wrong direction)	fail
H5	Top-2 mixing beats argmax (`ranpac/vit`)	0.674 vs 0.671, Δ +0.003	[+0.002, +0.004]	< 0.001	supported, negligible size	hold, small

H (T = 10 unless stated)	Prediction	Result	95% CI	Holm p	Verdict	Pilot predicted
H6	A 100-way readout of the shared space beats routing to experts	0.803 vs 0.671, Δ +0.132	[+0.127, +0.138]	< 0.001	supported	hold

Selection error by router (ε, mean over seeds; lower is better).

Router	T = 5	T = 10	T = 20	Growth 5 → 20
`maxsoftmax` (self-report)	0.551	0.654	0.773	+0.222
`energy` (self-report)	0.514	0.627	0.771	+0.257
`ranpac/first` (stream-legal)	0.484	0.614	0.719	+0.235
`mahalanobis/first`	0.528	0.655	0.756	+0.228
`buffer/first` (replay logistic)	0.639	0.762	0.853	+0.214
`ranpac/vit` (pretrained)	0.159	0.178	0.189	+0.030
`mahalanobis/vit`	0.190	0.212	0.225	+0.035
`ncm/vit`	0.239	0.268	0.284	+0.045
`buffer/vit`	0.342	0.352	0.340	−0.002
`own_mahalanobis` (each expert's own space)	0.738	0.825	0.918	+0.180

What the results say, applying the interpretation rules fixed in the pre-registration (§7):

- The confidence-free rule wins, but only a little without good features. H1 is supported against the pilot's prediction, but the margin is 1.4 points of ε at T = 10 (5.2 at T = 20, unadjusted). Routing on the first task's backbone, the only stream-legal shared space here, still leaves ε above 0.6. The pilot's K1 reading stands in substance: at CIFAR-100 scale without pretraining, the binding component is the representation, not the rule.
- A strong shared space nearly removes selection error and flattens its growth. With ImageNet ViT-B/16 features, ε fell to 0.16–0.19 and grew by only 0.030 from 5 to 20 tasks, against 0.257 for energy (H2, H3). ImageNet's classes overlap CIFAR-100's, so this arm is not stream-legal. It measures what a router can do *given* a strong representation, not that one can be learned from the stream.
- With that space, the experts are the bottleneck (H6). A plain linear readout of the same ViT features, with no experts at all, scored 0.803 at every T. The best routed system scored 0.637–0.693. Under the strong router, experts were right on only 76–85% of the inputs they were handed (A_within), against 89–98% under max-softmax. Self-report routes the easy inputs it can answer

and misses the rest; a good router hands experts their hard cases too. Per rule §7, the frozen-expert design adds nothing over a linear readout of the shared features at this scale. Experts trained from scratch on 2,250–9,000 images per task are simply weaker than the representation that routes them.

- Margin abstention did not transfer from MNIST (H4). At T = 10 the router's top-1 minus top-2 margin ranked errors slightly *worse* than the chosen expert's own max-softmax (AURC 0.158 vs 0.153), and clearly worse for the stream-legal router (0.452 vs 0.429, unadjusted). Inside one expert, confidence still carries information about within-task correctness. The ledger's MNIST result (9.3) compared the margin with confidence used *across* experts, a different comparison. At T = 20 the ViT margin was better (0.140 vs 0.157, unadjusted, not tested), so the ordering may depend on task count.
- Top-2 mixing is real but negligible (H5). +0.3 points with the ViT router; +3.1 points with the stream-legal router (unadjusted), where the router is less sure. Mixing pays only when routing is weak.
- Replay-trained routers lost (secondary). A logistic router on a 2,000-image buffer had ε 0.15 higher than RanPAC in the first-task space and 0.17 higher in the ViT space, consistent with ledger 9.5. Each expert's own-space Mahalanobis density was the worst router of all.

What changes in the design table above: "Scoring signal" and "Feature space" are now lab-supported at CIFAR scale, with the caveat that stream-legal features barely beat self-report. "Abstention" moves from evidence-backed to not supported at CIFAR scale: abstain on a combination of the router's margin and the chosen expert's confidence, not the margin alone. "Combination" is settled as negligible under a strong router. R5's hierarchy test was not run: with max-type task scores, a two-level router returns the same choice as a flat one, so the test cannot separate the designs. The headline for NeuroGenesys is H6. Before investing in routers among weak experts, check whether a readout of the shared representation already wins; at Split-CIFAR-100 with ViT features, it does by 13 points.

Results files: `results/results.md`, `results/results.json` and `code/confirm.py` in this folder. Ledger section 15.

Conclusion

The question "what is the optimal router" breaks into two parts with very different evidential status. The part about what the router should *not* read is settled. Across the theory, three literatures and two lab scales, cross-expert confidence fails as a routing signal, and better calibration, more voters or more diverse experts do not rescue it. The part about what the router *should* read is a strong hypothesis that has been tested only where it was easy: a shared space, closed-form prototypes and margin abstention, so far measured on Split-MNIST and in borrowed CIFAR numbers. Two other components, combination and hierarchy, earn their place mainly by cutting cost, by shrinking the candidate set before expensive experts run, rather than by accuracy. That changes what "optimal" should mean for NeuroGenesys: the router that keeps ε flat as the library grows, not the one with the best top-1 at a fixed T. Scale is where every lab testbed shows selection failing.

There is a second, less obvious implication. A statistics-keyed router fixes both routing and succession. A router that admits experts by adding sufficient statistics, and is audited on probes it never trained on, gives graded succession its evaluator for free. A learned router retrained after each admission cannot provide that, because it is no longer independent of the experts it judges. R1–R5 have now answered the CIFAR-100 half of this. Without pretraining, the confidence-free rule beat self-report by only 1.4 points of ε: the shared representation was too weak to route on. With a strong pretrained representation,

selection error nearly vanished and stopped growing with the library, but a plain readout of that representation then beat the routed experts by 13 points. Both results point the same way: the programme's next investment belongs in features rather than in gating, and a router is worth building only where the experts know something the shared space does not.

Published by NeuroGenesys (AW3 Technology, Inc.). Canonical version: https://neurogenesys.si/research/optimal-router-architecture