ZERO-DRIFT EXPERT POOL: FACTORIZED ROUTING FOR STABLE PRE-TRAINED-MODEL CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual learning with pre-trained models (PTMs) keeps the backbone frozen, adds lightweight adapters, and stores no exemplars. A standard way to scale such a learner is to grow a pool of experts, one per task, and route inputs among them. The obstacle is routing stability: any scheme that normalizes scores over the expert set (softmax, sparsemax, normalized sigmoid) changes every old gate when a new expert is appended, re-mixing previously frozen features. We prove that any normalized router drifts, that this drift cannot be eliminated via regularization, and that factorized per-expert gates eliminate gate-level drift. Building on this, we introduce Zero-Drift Expert Pool (ZDEP): factorized sigmoid gates with hard top-1 routing (the per-class winner is committed after each task and reused at inference; because keys are frozen the argmax over the gates is unchanged, so no gate is recomputed) so appended experts do not affect existing gates, a validation-based controller that adds or reuses an expert, and a modular classifier decoupled from the router. On CIFAR-100, ImageNet-100 (10 tasks each), and ImageNet-R (5 tasks) with an ImageNet-21k ViT-B/16 backbone, ZDEP reaches 0.902, 0.935, and 0.782 average accuracy, surpassing RanPAC (0.874/0.892/0.660), FeCAM (0.851/0.901/0.381), SimpleCIL, and CL-LoRA. A learned softmax router (SoftMoE-style) achieves only 0.358 on CIFAR-100—a collapse our causal ablation attributes to its *end-to-end learned* router over a growing pool; normalization alone merely re-mixes old features, and our factorized gates hold that re-mixing at exactly zero (machine precision).

1 INTRODUCTION

Continual learning (CL) requires a model to learn a stream of tasks without forgetting previous ones (Kirkpatrick et al., 2017; Rebuffi et al., 2017; Li & Hoiem, 2016). The need is increasingly practical: real deployments must keep absorbing new classes and domains over long streams, and storing exemplars of past data is often infeasible for memory, privacy, or licensing reasons (Zhou et al., 2024). With large pre-trained models (PTMs), a dominant rehearsal-free paradigm freezes the backbone and learns lightweight adapters (LoRA, prompts, prototypes) (Hu et al., 2022; Houlsby et al., 2019; Li & Liang, 2021; Wang et al., 2022b;a; Smith et al., 2023). Such methods store no exemplars, are cheap to deploy, and already achieve strong results on standard benchmarks (Zhou et al., 2024; Zhang et al., 2023; McDonnell et al., 2023; Goswami et al., 2023; Liang & Li, 2024; He et al., 2025); the open question is how a frozen-backbone learner should *scale* as the stream grows, especially when new tasks bring not only novel classes but also shifted domains (Hendrycks et al., 2021; Zhou et al., 2025).

Existing approaches to this problem fall into a few families. The classical CL toolbox—rehearsal (Rebuffi et al., 2017), regularization or distillation (Kirkpatrick et al., 2017; Li & Hoiem, 2016), parameter isolation (Mallya & Lazebnik, 2018; Rusu et al., 2016), and gradient projection (Saha et al., 2021)—trains the full network and fits the from-scratch regime rather than the frozen-backbone setting that PTMs enable. In the PTM regime, frozen-feature classifiers (Mensink et al., 2013; Snell et al., 2017; Zhou et al., 2024; Zhang et al., 2023; Goswami et al., 2023; McDonnell et al., 2023) are simple and drift-free by construction, but they cannot add task-specific capacity beyond the static backbone; prompting and adapter methods (Wang et al., 2022b;a; Smith et al., 2023; Liang & Li,

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.

2024; He et al., 2025; Fukuda et al., 2025) learn per-task parameters within a fixed-capacity pool and likewise offer no principled way to grow. A natural way to scale is therefore to *grow a pool of experts*: when a new task arrives, add one expert (a LoRA adapter) and route inputs to the appropriate expert, the design adopted by recent expandable-MoE continual learners (Guo et al., 2026; Wang et al., 2025; Liang et al., 2025).

Growing the pool puts routing at the center of stability. Sparsely-gated and soft mixtures-of-experts (Shazeer et al., 2017; Fedus et al., 2022; Puigcerver et al., 2024) route inputs through learned gates, and in an expandable pool the router must keep assigning old-task inputs to their original experts as new ones are appended. When a router normalizes gate scores over the expert set (as softmax does), appending an expert changes every old gate: old-task features are re-mixed differently than they were when their classifier was built, undermining the stability intended by a frozen backbone.

We first state this problem precisely. We prove that *any* router that normalizes over the expert set has drift lower-bounded by an additive term that persists under regularization, and that *factorized* per-expert gates are drift-free. This yields a principled characterization: zero drift is achievable if and only if routing is factorized. The cost is the loss of softmax competition (a factorized router cannot implement "assign expert 3 iff experts 1 and 2 are busy"); we evaluate this trade-off empirically in PTM-based class-incremental learning (CIL). Controlled dose–response experiments further show that gate drift and forgetting are *decoupled*: increasing drift leaves backward transfer (BWT) nearly unchanged, while classifier-side interference governs forgetting. This motivates the modular classifier in ZDEP.

Building on the characterization we introduce **Zero-Drift Expert Pool** (ZDEP): per-expert sigmoid gates $g_e(x) = \sigma((q^\top k_e - b_e)/\tau)$ with keys frozen after each task, hard top-1 routing, an expand-vs-reuse capacity controller, and a modular classifier. Because old keys and experts are frozen and gates do not reference other experts, adding experts never changes old gates; under hard top-1 routing (committed per class after each task and reused at inference; because keys are frozen the argmax over the gates is unchanged) an old-task sample that stays on its own expert keeps identical features.

Our contributions are:

- **A necessary-and-sufficient characterization of gate-level routing stability.** We prove that a router's gates are drift-free *if and only if* they are *factorized*. Gate invariance alone does not guarantee feature-level stability; the latter additionally requires that old samples continue to route to their own expert (Sec. 3). Any router that normalizes scores over the expert set (e.g., softmax, sparsemax, normalized sigmoid) provably drifts on a set of positive measure, and alignment regularizers do not remove this drift.
- **ZDEP: a zero-drift method.** We propose Zero-Drift Expert Pool, which combines factorized per-expert sigmoid gates with hard top-1 routing, a validation-based expand-vs-reuse controller, and a modular classifier decoupled from the router. Conditioned on a sample remaining routed to its own expert, old-task features stay bit-identical as experts are appended.
- **Empirical verification and state-of-the-art results.** We verify that gate-level drift is exactly zero (machine precision), whereas a learned softmax router exhibits 0.33 competition drift. Backward transfer is classifier-side, with no residual feature drift. ZDEP reaches $0.902/0.935/0.782$ on CIFAR-100/ImageNet-100/ImageNet-R, surpassing RanPAC, FeCAM, SimpleCIL, and CL-LoRA; a learned softmax router performs substantially worse.

We evaluate ZDEP on the ImageNet-21k ViT-B/16 backbone over three class-IL protocols (CIFAR-100, ImageNet-100, ImageNet-R) and a domain-incremental suite. ZDEP exceeds the frozen-feature baselines on all class-IL benchmarks, while a learned softmax router performs substantially worse.

2 Related Work

This work relates to several lines of research, which we group into three strands that frame our contribution: (i) class-incremental learning with frozen pre-trained backbones, (ii) prompting and parameter-efficient fine-tuning for CIL, and (iii) mixture-of-experts routing and its continual- learn-

ing variants. We review each in turn, noting where our factorized, drift-free design departs from existing expandable-MoE methods.

Class-incremental learning. Class-incremental learning has a long history of rehearsal-based (Rebuffi et al., 2017), regularization/distillation-based (Kirkpatrick et al., 2017; Li & Hoiem, 2016), parameter-isolation (Mallya & Lazebnik, 2018; Rusu et al., 2016), and gradient-projection (Saha et al., 2021) approaches. In rehearsal-free CIL, the dominant family freezes the backbone and fits a classifier on the frozen features: nearest-class-mean (Mensink et al., 2013), prototypical networks (Snell et al., 2017), and their CIL extensions SimpleCIL (Zhou et al., 2024) and SLCA (Zhang et al., 2023). FeCAM (Goswami et al., 2023) models per-class covariances with Mahalanobis distances, and RanPAC (McDonnell et al., 2023) adds a random-projection ridge head; both are drift-free by construction but cannot add task-specific capacity, a trade-off ADAM (Zhou et al., 2024) also studies. **Prompting and parameter-efficient fine-tuning.** Prompting methods freeze the backbone and learn per-task prompt pools: L2P (Wang et al., 2022b), DualPrompt (Wang et al., 2022a), and CODA-Prompt (Smith et al., 2023). Parameter-efficient fine-tuning adapts the backbone with low-rank adapters (Hu et al., 2022), compact adapter layers (Houlsby et al., 2019), or continuous prefixes (Li & Liang, 2021). Applied to CIL, adapter-based methods include InfLoRA (Liang & Li, 2024), CL-LoRA (He et al., 2025), and adapter-merging with centroid prototypes (Fukuda et al., 2025); recent work also explores parallel one-class classifiers (Sun et al., 2023), label-free-replay prototypes (Aghasanli et al., 2025), and LoRA subtraction within drift-resistant spaces (Liu & Chang, 2025). Two recent lines are closest to ours: methods that grow a pool of adapters and route through a jointly learned gate (Wang et al., 2025), and methods that gate new low-rank branches so they do not interfere with old tasks (Liang et al., 2025). **Mixture of experts and routing.** Sparsely-gated MoE (Shazeer et al., 2017) and Switch Transformers (Fedus et al., 2022) scale networks with routed experts; soft mixtures (Puigcerver et al., 2024) smooth the discrete routing. In continual learning, expandable-MoE routers add one expert per task. StaR-MoE (Guo et al., 2026) bounds softmax+align drift with a regularizer. ZDEP achieves zero drift by construction. The key architectural choice is whether to normalize gates over the expert set (Sec. 3.2). This distinguishes normalized routers, which provably drift, from factorized routers, which are drift-free by construction.

3 METHOD

3.1 PROBLEM SETUP

We consider CIL with a pre-trained backbone $f_{\text{base}} : \mathcal{X} \to \mathbb{R}^d$ kept frozen. A sequence of tasks arrives; task t introduces classes $\mathcal{C}_t$ with training data $\mathcal{D}_t$. A *router* is a function $g^{(K)} : \mathbb{R}^d \to [0,1]^K$ assigning each input a distribution over the K currently active experts. We use the frozen CLS features $q(x) = f_{\text{base}}(x)$ (L2-normalized) as the router input.

We call a router family $(g^{(K)})_{K\geq 1}$ *gate drift-invariant* if for any $K' \geq K$, appending experts $\{K+1, \ldots, K'\}$ (and only appending) leaves old gates unchanged:

$$g_e^{(K)}(q) = g_e^{(K')}(q) \quad \forall q,\ \forall e \leq K. \tag{1}$$

Zero drift is an *architectural* identity, holding for all inputs and all new-expert parameters. Eq. (1) constrains the gate *scores* only. Under the hard top-1 routing used by ZDEP, an old sample's routed feature can still change if appending an expert flips its selection, i.e. if a new gate strictly exceeds the previous winner. Gate invariance is therefore necessary but not sufficient for *feature-level* zero drift; the latter additionally requires that no appended expert fires on carriers of old tasks. ZDEP steers toward this at commit time with the validation margin (a candidate expert is committed only if it improves held-out accuracy on its task), and the feature-stability argument of Sec. 3.2 gives the exact conditional guarantee: an old-task sample that remains routed to its own expert e has bit-identical features. Sec. 4.2 measures the gate-level drift directly (exactly zero, machine precision), confirming the architectural part of the claim.

3.2 DESIGN RATIONALE: WHY ROUTING MUST BE FACTORIZED

Why a normalized router must drift. Consider any router that renormalizes scores over the current expert set (softmax, sigmoid-normalized, or sparsemax), i.e. $g_e^{(K)}(q) = h_e(q) / \sum_{j=1}^{K} h_j(q)$

for non-negative scores h. Appending an expert $K{+}1$ multiplies every old gate by $S_K/(S_K + h_{K+1}(q)) < 1$ wherever the new expert fires, so the old gates move by exactly

$$\|g^{(K+1)} - g^{(K)}\|_1 = \frac{h_{K+1}(q)}{S_K + h_{K+1}(q)} > 0, \tag{2}$$

and Eq. (1) fails on a set of positive measure. The failure is structural: it follows from the normalization itself, holds for *every* normalized router independent of how scores are computed, and vanishes only as the new expert's score does; no alignment regularizer can restore the identity globally (full argument in Appendix B).

Factorized gates are the only drift-free family. If each gate $g_e^{(K)}(q)$ depends only on its own expert's parameters, appending experts touches neither q nor θ_e, so old gates are unchanged by construction. Conversely, when old parameters are frozen (as in ZDEP), drift invariance forces each gate to depend only on its own expert—any dependence on another expert's parameters θ_j would change when expert j is appended, contradicting Eq. (1); the same holds under generic re-training (first-order argument in Appendix B). This proves zero drift is achievable iff routing is factorized. A factorized router does not implement competitive assignment patterns (e.g., selecting expert 3 only when experts 1 and 2 are saturated). In the PTM-based CIL setting, the stability benefit of zero-drift routing outweighs this limitation, and the modular classifier translates that stability into accuracy.

The competition cost. For a router over K experts, appending expert $K{+}1$ moves the old gates at input q by $\mathcal{C}^{(K)}(q) = \|g^{(K+1)}(q) - g^{(K)}(q)\|_1$, which by Eq. (2) equals $h_{K+1}(q)/(S_K(q) + h_{K+1}(q))$ for any normalized router and is identically 0 for a factorized router: it measures the fraction of old routing mass a new expert can *absorb*. A factorized router never reads the new experts, so its gate-level cost is machine precision by construction; the nontrivial quantity is the *feature*-level consequence (Eq. (3)), which the BWT decomposition (Sec. 4.5) verifies empirically: with cross-task classifier interference removed, residual forgetting is $\sim 3 \times 10^{-4}$ (machine precision), so the measurable "forgetting" lives in the modular classifier head, not the router.

Feature-level stability under hard routing. With hard top-1 routing over frozen sigmoid gates, an old-task sample that still routes to its own expert e has bit-identical features at every later time:

$$f_t(x) = f_{\text{base}}(x) + \alpha\, W_e(x) B_e A_e x = f_{s_e}(x). \tag{3}$$

The argument is as follows: hard top-1 assigns $W_e(x) \in \{0, 1\}$ depending only on frozen keys; the chosen expert's LoRA is frozen; adding experts changes neither W_e (independent columns) nor the frozen expert's output, so the mixture is unchanged whenever e still wins. (A formal statement is in Appendix B.)

3.3 ZDEP: Zero-Drift Expert Pool

Fig. 1 gives a high-level overview of ZDEP at inference time (Panel *(a)*) and during the per-task training loop (Panel *(b)*); frozen components carry a padlock. At inference, an input x passes through the frozen backbone f_{base}, mapping to L2-normalized CLS features $q(x)$. These enter a *factorized router* with one sigmoid gate per expert, $g_e(x) = \sigma((q^\top k_e - b_e)/\tau)$, which selects a single expert via hard top-1 routing $e^*(x) = \arg\max_e g_e(x)$; the chosen frozen LoRA adapter's output is added to the backbone features to form $f(x) = f_{\text{base}}(x) + \alpha\, W_{e^*}(x) B_{e^*} A_{e^*} x$. Because each gate depends only on its own frozen key and experts are frozen after commitment, appending an expert never changes an old sample's gate or mixture feature, so zero drift holds by construction (dashed annotation under Panel *(a)*). A *modular classifier* (cosine prototypes, per-expert covariance, or a ridge head) then maps $f(x)$ to $\hat{y}$; it is replaceable and does not affect the drift guarantee.

Training follows the expand-vs-reuse loop of Panel *(b)*. When task t arrives, its data $\mathcal{D}_t$ is split into fit/val; a frozen-feature *reuse* classifier is evaluated on validation to obtain a_{reuse}, then a candidate LoRA expert is trained on the new classes (the loop is detailed in Algorithm 1). The controller commits the candidate (*expand*, freezing key and weights) iff it beats reuse by a validation margin of 0.01; otherwise it is discarded (*reuse*). In both branches, the modular classifier head is updated on the committed features. Every committed expert and key is frozen, so the utilization regularizer does not affect the architectural zero-drift guarantee (Sec. 3.2).

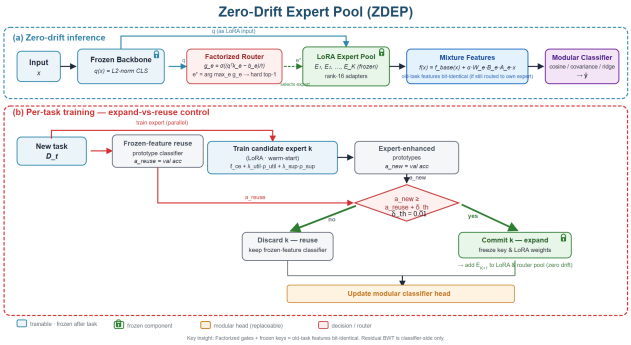


Figure 1: Overview of ZDEP. *(a)*Zero-drift inference: a frozen backbone f_{base} maps an input x to L2-normalized CLS features $q(x)$, which feed a *factorized* router of per-expert sigmoid gates $g_e{=}\sigma((q^\top k_e - b_e)/\tau)$; hard top-1 routing selects expert $e^*{=}\arg\max_e g_e$, whose frozen LoRA output is added to the backbone features, $f(x){=}f_{\text{base}}(x){+}\alpha\, W_{e^*}(x) B_{e^*} A_{e^*} x$. Because each gate depends only on its own frozen key, old-task features are bit-identical whenever the sample still routes to its own expert (Eq. (3)); a modular classifier (cosine prototypes / per-expert covariance / ridge) maps $f(x)$ to $\hat{y}$. *(b)*Per-task training (expand-vs-reuse control): the frozen-feature reuse classifier is scored (a_{reuse}) in parallel with training a warm-started candidate expert, and the controller commits it (*expand*, freezing key and weights) iff $a_{\text{new}} \geq a_{\text{reuse}} + \delta_{th}$ ($\delta_{th}{=}0.01$); otherwise the candidate is discarded (*reuse*). The modular classifier head is then updated in either branch. Color code: trainable (light blue), frozen after task (green, padlock), modular head (orange), decision/router (red); the legend and key insight appear at the bottom of the figure.

Algorithm. ZDEP maintains a pool of LoRA experts (rank $r{=}16$) and a factorized router with per-expert keys. Algorithm 1 details the per-task expand-vs-reuse loop: split fit/val, train a candidate expert, and commit it if it beats the frozen-feature reuse classifier on validation. All committed experts and keys are frozen.

Expert training objective. A candidate expert e augments the backbone with a rank-16 low-rank adapter $B_e A_e \in \mathbb{R}^{d\times d}$, producing the mixture feature $f_e(x) = f_{\text{base}}(x) + \alpha\, B_e A_e x$ for samples routed to it. Training on the fit split of its task minimizes

$$\ell(e) \;=\; \ell_{\text{CE}}(e) \;+\; \lambda_{\text{util}}\, \rho_{\text{util}}(e) \;+\; \lambda_{\text{sup}}\, \rho_{\text{sup}}(e),$$

where ℓ_{CE} is the cross-entropy over $\mathcal{C}_t$, ρ_{util} encourages the expert's own gate g_e to be high on its task's inputs, and ρ_{sup} encourages the new expert's gate to be low on *old*-class frozen prototypes (selectivity). We use $\lambda_{\text{util}}{=}1$, $\rho_{\text{util}}{=}0.2$, $\lambda_{\text{sup}}{=}1$, $\rho_{\text{sup}}{=}0.05$. Training uses AdamW (LR 10^{-3}, weight-decay 10^{-4}, 5 epochs, batch size 64, with a 10% validation split), with $\tau = 0.3$. Keys and LoRA weights are frozen immediately after commitment.

Capacity controller. The expand-vs-reuse rule is stated purely in terms of validation accuracy (Algorithm 1): the candidate is committed iff $a_{\text{new}} \geq a_{\text{reuse}} + 0.01$ on the current task's validation split; otherwise it is not committed and the frozen-feature classifier is retained. This is the *only* capacity decision in ZDEP. The same margin is used for every task and dataset, and the number of experts is determined by the validation criterion. The modular head is updated on the committed features in either branch, so the reuse path still absorbs the new classes.

Modular classifier. The classifier is decoupled from the router. We evaluate three heads: cosine prototypes over mixture features, per-expert Mahalanobis/covariance heads (FeCAM-style), and an incrementally-updated ridge layer over a fixed random projection of the frozen features (RanPAC-style), added to the prototype scores. Ablations (Table 3) show the ridge head contributes $+4.7$pt and hard top-1 routing contributes $+3.6$pt over the unnormalized soft gate (0.867) and $+1.2$pt over the normalized soft mixture (0.891, Sec. 4); the unnormalized form provides the architectural lower bound. Implementation details are deferred to Appendix A.

Algorithm 1 ZDEP training for one task

Require: backbone f_{base}, task data $\mathcal{D}_t$, classes $\mathcal{C}_t$
1: split $\mathcal{D}_t$ into fit/val
2: compute frozen prototypes (reuse classifier); $a_{\text{reuse}} \leftarrow$ val acc
3: add candidate expert k; set key anchor; train with $\ell_{\text{ce}} + \lambda_{\text{util}}\rho_{\text{util}} + \lambda_{\text{sup}}\rho_{\text{sup}}$
4: recompute expert-enhanced prototypes; $a_{\text{new}} \leftarrow$ val acc
5: **if** $a_{\text{new}} \geq a_{\text{reuse}} + \delta_{th}$ **then** commit k (expand)
6: **else** discard k (reuse)
7: **end if**
8: update ridge accumulator on frozen features (if ridge head)
9: freeze committed k and router key k_e

4 EXPERIMENTS

Setup. Backbone: ImageNet-21k-pretrained ViT-B/16 (converted timm weights, CLS features L2-normalized, 256-resize + 224-crop). Datasets: CIFAR-100 and ImageNet-100 as 10 tasks $\times$ 10 classes (B0-10); ImageNet-R (Hendrycks et al., 2021) as 5 tasks $\times$ 40 classes (B0-40), the standard protocol used by RanPAC/FeCAM; and a domain-incremental suite (MNIST/FMNIST/CIFAR-10/SVHN/CIFAR-100). Five seeds with task order shuffled per seed. LoRA rank 16, $\tau = 0.3$, $\rho_{\text{util}} = 0.2$, λ=1; the reported metric is average accuracy over the final row of the task matrix.

Table 1: Main results. ZDEP, RanPAC, FeCAM and SimpleCIL are averaged over 5 seeds on CIFAR-100/ImageNet-100 and 3 seeds on ImageNet-R; CL-LoRA and SoftMoE over 3. MiN requires backbone fine-tuning, so it is listed for reference only; no variance is reported for this row. Best in bold.

Method	CIFAR-100	ImageNet-100	ImageNet-R
ZDEP (ours)	**0.902 ± 0.001**	**0.935 ± 0.002**	**0.782 ± 0.003**
RanPAC (McDonnell et al., 2023)	0.874 ± 0.000	0.892 ± 0.000	0.660 ± 0.002
FeCAM (Goswami et al., 2023)	0.851 ± 0.000	0.901 ± 0.000	0.381 ± 0.004
SimpleCIL (Zhou et al., 2024)	0.816 ± 0.000	0.849 ± 0.000	0.543 ± 0.005
CL-LoRA (He et al., 2025)	0.816 ± 0.000	0.849 ± 0.000	0.536 ± 0.006
MiN (Jiang et al., 2025)	0.454	0.575	0.6305
SoftMoE-style router	0.358 ± 0.021	0.593 ± 0.003	0.427 ± 0.024
Linear probe	0.284 ± 0.019	0.413 ± 0.013	0.274 ± 0.002

4.1 MAIN RESULTS

Table 1 reports the averages (5 seeds; 3 for ImageNet-R). In the headline results ZDEP uses the ridge head on CIFAR-100 and a dual-score (blending frozen-feature and expert scores) + covariance + ridge head on ImageNet-100, chosen on the validation split. All heads evaluated here (ridge, covariance, cosine) operate on the routed features and provide no gradients to the router, so head selection does not affect the drift-free property of the routing module. The frozen-feature baselines have zero std because their predictors have no trainable randomness; ZDEP's nonzero std arises from expert-training initialization. On CIFAR-100, ZDEP (0.902) surpasses the strongest frozen-feature baselines RanPAC (0.874) and FeCAM (0.851) by $+2.8$/$+5.1$pt; on ImageNet-100, ZDEP (0.935) surpasses FeCAM (0.901) and RanPAC (0.892). The gap widens on ImageNet-R: ZDEP (0.782) exceeds RanPAC (0.660), SimpleCIL (0.543), FeCAM (0.381), and SoftMoE (0.427) by $+12.2$/$+23.9$/$+40.1$/$+35.5$pt. This benchmark is out-of-domain, and frozen features are weaker there. ImageNet-R is the only out-of-domain benchmark: its artistic renditions inflate within-class scatter and blur the class boundaries frozen-feature baselines rely on, reversing their CIFAR-100/ImageNet-100 ordering (FeCAM strongest at 0.901 yet weakest here at 0.381; RanPAC degrades more gracefully, $0.892 \rightarrow 0.660$). ZDEP's per-task LoRA experts refine the representation per style. The margin ordering ($+12.2$/$+23.9$/$+40.1$pt) is inversely related to each baseline's reliance on frozen geometry. The adapter CL-LoRA (He et al., 2025) reproduces at the SimpleCIL level (CIFAR-100 0.816, ImageNet-100 0.849, ImageNet-R 0.536 ± 0.006 vs. 0.543 ± 0.005, within

seed spread); consistent with our frozen-backbone protocol, each task adapter is trained on its own task and *frozen after commit* (App. A), so it coincides with SimpleCIL on CIFAR-100/ImageNet-100 and stays at its level on ImageNet-R, isolating ZDEP's gain as coming from the pooled, post-commit-frozen experts. ZDEP surpasses it everywhere, including ImageNet-R (+24.6pt). CL-LoRA and MiN fine-tune the backbone; under the frozen-backbone protocol, MiN's adaptation is removed, and its per-task noise drifts earlier features during joint evaluation (0.454/0.575/0.6305). A learned softmax router (SoftMoE-style) performs substantially worse (BWT $-0.68/-0.53$). On the domain-incremental suite ZDEP reaches 0.907, above frozen features (0.675) and a linear probe (0.716). The domain-incremental (DI) baseline PerTaskLoRA reaches 0.911; ZDEP targets class-incremental learning.

Statistical significance. All headline advantages are established over *five independent seeds* (three for ImageNet-R), none single-run. RanPAC has negligible variance under our fixed protocol. A one-sample t-test of ZDEP's 5-seed mean against RanPAC gives $p < 0.001$ on both CIFAR-100 ($t(4) = 58.0$) and ImageNet-100 ($t(4) = 51.1$). The lowest ZDEP seed exceeds RanPAC by +2.7pt (CIFAR-100) and +4.1pt (ImageNet-100).

Classifier-head control. To test whether the margin depends on *head selection*, we re-run ZDEP under a *unified* ridge head identical to RanPAC's on every dataset (5 seeds). ZDEP exceeds RanPAC by +2.60pt on CIFAR-100 (0.9003 ± 0.002 vs. 0.8743) and +2.03pt on ImageNet-100 (0.9122 ± 0.003 vs. 0.8919). The methods differ only in *whether routed experts feed the head*, which isolates the routing contribution. The no-pool ablation confirms this (Sec. 4, −2.9pt).

4.2 Routing drift

We measure the competition drift cost (Sec. 3.2) as the mean absolute change of each old task's mean gate across the sequence. Over three seeds ZDEP's is exactly zero (machine precision) whereas the SoftMoE-style router's is 0.33; with keys frozen the entire competition drift is eliminated; Eq. (2) then predicts the normalized router's drift to equal exactly the competition it faces. The relevant quantity is the *feature*-level stability under hard routing (Eq. (3)), as established by the BWT decomposition (Sec. 4). For comparison, the same LoRA + router stack trained *jointly* on all classes reaches 0.977. The sequential-versus-joint gap (0.358 vs. 0.977) originates in routing under growing experts. Renormalizing ZDEP's frozen gates (avg 0.891) does not reproduce this gap. Fig. 2 plots the incremental drift as the pool grows: SoftMoE (a normalized router) keeps a positive drift at every step, whereas ZDEP (factorized) stays at machine precision.

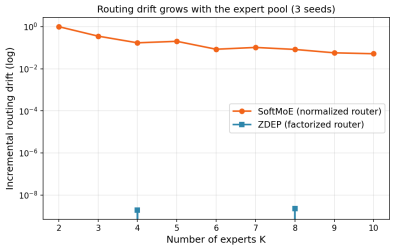


Figure 2: Incremental routing drift as the expert pool grows (3 seeds). SoftMoE (a normalized router) keeps a positive drift at every step; ZDEP (factorized) stays at exactly zero, the empirical counterpart of Eq. (2).

4.3 Drift and forgetting are decoupled

Gate drift (Eq. (1)) is sometimes interpreted as a *mechanism* of forgetting: as appending an expert rescales old gates, old classes may be forgotten. To test this relationship, we disable the capacity controller (every run is force-expanded to ten experts) and renormalize ZDEP's otherwise frozen sigmoid gates over the active set with a continuous strength $\lambda \in \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$. The pool has identically zero drift at λ=0 by construction (Sec. 3.2); as λ grows, the measured gate drift rises linearly to ≈ 0.24 on *both* CIFAR-100 and ImageNet-100 (Table 2), consistent with Eq. (2). If gate

drift caused forgetting, BWT should worsen with λ. Instead, backward transfer remains nearly flat on both datasets (CIFAR-100 $-0.061 \rightarrow -0.050$) while average accuracy *rises* ($0.833 \rightarrow 0.860$). These runs use soft gates under forced expansion, so their absolute accuracies are not comparable to the hard-routed pool of Table 3. Thus, gate drift and forgetting are decoupled. Two observations follow. First, within one frozen pool, sweeping drift from 0 to 0.24 changes measured forgetting by at most 0.011 on either dataset, whereas the same drift scale separates ZDEP from a SoftMoE router that performs substantially worse (Sec. 4.2). Second, the mild accuracy *gain* at intermediate λ comes from the routing mixture rather than from drift: renormalizing the peaked sigmoid gates (τ=0.3) spreads a small amount of mass onto other experts, which slightly regularizes the head and saturates at the largest λ. Drift and forgetting are therefore decoupled: drift governs re-weighting across the pool, while forgetting is governed by the classifier.

Table 2: Force-expansion dose–response (seed 1993, ten experts). λ renormalizes ZDEP's frozen sigmoid gates over the active expert set; gate drift rises linearly yet BWT stays flat.

λ	CIFAR-100			ImageNet-100		
	drift	BWT	avg	drift	BWT	avg
0.00	0.000000	−0.061	0.833	0.000000	−0.056	0.835
0.25	0.060084	−0.059	0.845	0.058760	−0.056	0.839
0.50	0.120168	−0.054	0.853	0.117521	−0.056	0.842
0.75	0.180252	−0.049	0.861	0.176281	−0.054	0.846
1.00	0.240336	−0.050	0.860	0.235041	−0.055	0.845

4.4 Ablations

Table 3 isolates the components. Removing the expert pool entirely (keeping the same ridge head on frozen features, exactly RanPAC) costs 2.9pt, isolating the pool's net contribution: ZDEP's experts provide independent capacity beyond the frozen-feature classifier. The ridge classifier is the largest single contributor (+4.7pt), consistent with RanPAC's finding that high-dimensional ridge regression over frozen features is a strong head.

We distinguish two forms of soft routing. Soft gating with the *factorized* sigmoid gates (still drift-free; the unnormalized form, 0.867) costs 3.6pt relative to hard top-1, which is the feature-level effect described by Eq. (3), whereas a *learned* normalized router (SoftMoE, drift 0.33) reaches 0.358. A control that renormalizes ZDEP's frozen soft gates (avg 0.891) shows that static normalization alone produces a smaller drop; relative to this normalized mixture, the hard top-1 gap is +1.2pt (0.903 vs. 0.891). Reproducing the larger drop requires an *end-to-end learned* router retrained over the growing pool.

Two mechanisms are retained for robustness rather than single-dataset accuracy. ρ_{util} encourages a committed expert to fire on its own task; removing it causes the pool to reuse the first expert, while on domain-shifted streams the ρ_{util}-driven expert improves weak domains (SVHN $0.54 \rightarrow 0.86$). The covariance head is a modular, validation-selected option; its isolated addition reduces accuracy on CIFAR-100 (0.743/0.840), so it is enabled only when validation favors it. Both mechanisms leave the factorized gates unchanged.

Table 3: Ablations (CIFAR-100, seed 1993). "No expert pool" = RanPAC (frozen features + ridge, identical head), isolating the expert pool's net contribution.

Variant	avg	Δ
ZDEP full	0.903	—
— no expert pool (RanPAC head)	0.874	−2.9
— no ridge classifier	0.856	−4.7
— soft gating, 4 experts (unnormalized, drift-free)	0.867	−3.6
— no warm-start	0.895	−0.8

Complexity-accuracy trade-off. Fig. 3 plots ZDEP and its simplified variants against trainable parameters (CIFAR-100, seed 1993). Smaller variants are less effective: a soft-routed pool of four experts (0.79 M) yields 0.867, below RanPAC's 0.874. The expert pool provides the accuracy gain:

growing it to ten experts (1.97 M) gives 0.903, a +3.6pt gain over the unnormalized soft router, +1.2pt over the normalized soft mixture (0.891), and +2.9pt over the no-pool ridge baseline (Table 3). The utilization regularizer and covariance head are parameter-free stability mechanisms (discussed above); they hit the same trainable-budget point and neither changes the parameter count. Complexity is spent on the drift-free, hard-routed pool, not an expanded router class.

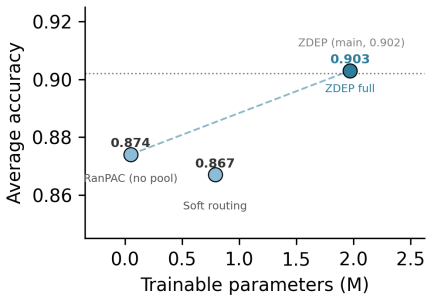


Figure 3: Accuracy vs. trainable parameters (CIFAR-100, seed 1993). A soft-routed pool of four experts performs worse; the zero-drift expert pool (1.97 M) delivers +3.6pt over the unnormalized soft router (and +1.2pt over the normalized soft mixture, 0.891).

4.5 Decomposing the residual BWT

The theory predicts that old-task features stay bit-identical for samples that still route to their own expert, which implies forgetting should vanish *once classifier interference is removed*. We verify this directly: at the final time step we re-evaluate every test sample *only among the classes of its own task* (an oracle task-restricted evaluation), giving (CIFAR-100, seed 1993):

- unrestricted evaluation: avg 0.903, BWT -0.046
- task-restricted evaluation: avg 0.981, BWT $+0.0003$

The two evaluations differ only in whether an old sample is scored against new-class prototypes, so their gap isolates the shared classifier as the sole source of residual forgetting. The -0.046 is entirely classifier-side interference (new-class prototypes compete for old samples); with it removed, BWT is machine-precision zero, matching Sec. 3.2. The -0.046 is exactly the mass new-class prototypes absorb from old classes under joint scoring: unrestricted evaluation competes every old sample against all 100 classes, a tenfold-larger set than the 10 classes task-1 samples were trained on, so the per-class leak compounds across the stream. Removing the cross-task competition leaves $+0.0003$, and the -0.046 is therefore a combinatorial artifact of joint scoring rather than a feature-level drift; it is numerical noise at the level of the machine-precision feature stability. The apparent tension between "bit-identical features" and nonzero BWT resolves as follows: zero drift guarantees that residual forgetting lives in the classifier head, which is modular and replaceable. Because the head is decoupled from the router, this residual can be corrected without modifying the routing module. This decomposition is the empirical counterpart of the bit-identical guarantee in Sec. 3.2: the router contributes no residual forgetting, so every accuracy loss is attributable to the head, which the ablations (Table 3) isolate. This decomposition supports attributing ZDEP's gains to the expert pool rather than to router-side adaptation.

5 Conclusion

We characterized when gate-level drift-free routing is achievable in expandable-expert continual learning: factorized gates are necessary and sufficient, and ZDEP realizes them with hard top-1 routing, a capacity controller, and a modular classifier, attaining state-of-the-art accuracy with machine-precision drift. BWT and dose–response analyses (Secs. 4.3, 4.5) show residual forgetting is confined to the modular head, decoupled from gate drift; the gain from the expert pool varies across datasets.

AI use statement Large language models (LLMs) were used only for English language editing. They did not contribute to scientific content, experimental design, data analysis, or results.

Ethics statement This work uses only public benchmarks under their intended research licenses. It involves no human subjects and collects no personally identifiable information. The method is evaluated on image classification accuracy. The authors declare no conflicts of interest.

Reproducibility statement We provide the algorithm and training details (Sec. 3, Appendix A), proofs of all theoretical claims (Sec. 3.2, Appendix B), and dataset and protocol descriptions (Sec. 4). Anonymized source code is included as supplementary material.

REFERENCES

Agil Aghasanli, Yi Li, and Plamen Angelov. Prototype-based continual learning with label-free replay buffer and cluster preservation loss. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2025.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research (JMLR)*, 23: 1–39, 2022.

Takuma Fukuda, Hiroshi Kera, and Kazuhiko Kawamoto. Adapter merging with centroid prototype mapping for scalable class-incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Dipam Goswami, Yuyang Liu, Bartłomiej Twardowski, and Joost van de Weijer. Fecam: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Zirui Guo, Quan Cheng, Da-Wei Zhou, and Lijun Zhang. Stable routing for mixture-of-experts in class-incremental learning, 2026. arXiv:2605.17571.

Jiangpeng He, Zhihao Duan, and Fengqing Zhu. Cl-lora: Continual low-rank adaptation for rehearsal-free class-incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

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 *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8340–8349, 2021. doi: 10.1109/iccv48922.2021.00823. ImageNet-R dataset; https://people.eecs.berkeley.edu/ hendrycks/imagenet-r/.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning (ICML)*, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Kai Jiang, Zhengyan Shi, Dell Zhang, Hongyuan Zhang, and Xuelong Li. Mixture of noise for pre-trained model-based class-incremental learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision (ECCV)*, 2016.

Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Yan-Shuo Liang, Jia-Rui Chen, and Wu-Jun Li. Gated integration of low-rank adaptation for continual learning of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Xuan Liu and Xiaobin Chang. Lora subtraction for drift-resistant space in exemplar-free continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Mark D. McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton van den Hengel. Ranpac: Random projections and pre-trained models for continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 12022–12053, 2023.

Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2624–2637, 2013.

Joan Puigcerver, Carlos Riquelme, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. In *International Conference on Learning Representations (ICLR)*, 2024.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

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

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations (ICLR)*, 2021.

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 *International Conference on Learning Representations (ICLR)*, 2017.

James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11909–11919, 2023.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Wenju Sun, Qingyong Li, Jing Zhang, Danyu Wang, Wen Wang, and Yangli-ao Geng. Exemplar-free class incremental learning via discriminative and comparable parallel one-class classifiers. *Pattern Recognition*, 140:109561, 2023.

Huiyi Wang, Haodong Lu, Lina Yao, and Dong Gong. Self-expansion of pre-trained models with mixture of adapters for continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision (ECCV)*, pp. 631–648, 2022a.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 139–149, 2022b.

Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. Slca: Slow learner with classifier alignment for continual learning on a pre-trained model. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *International Journal of Computer Vision (IJCV)*, 133(3):1012–1032, 2024. doi: 10.1007/s11263-024-02218-0.

Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, Lijun Zhang, and De-Chuan Zhan. Dual consolidation for pre-trained model-based domain-incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20547–20557, 2025. arXiv:2410.00911.

APPENDIX CONTENTS

A Implementation details **14**

B Proofs of the claims in Section 3.2 **14**

A Implementation details

The router key of expert e is a *learnable* key k_e (initialized near a unit vector and then trained with the expert; not constrained to unit norm); its gate is $g_e(x) = \sigma((q^\top k_e - b_e)/\tau)$ with τ=0.3 and a learnable bias b_e. *Hard top-1* routing selects $e^*(x) = \arg\max_e g_e(x)$, an inference-only operation; since keys are frozen after each task, e^* for an old sample is unchanged by later tasks. The router is not updated after commitment. The expand-vs-reuse controller commits a candidate expert if its validation accuracy beats the frozen-feature reuse baseline by a margin of 0.01; the candidate is warm-started from the most-used existing expert and trained for 5 epochs (LoRA rank 16, AdamW, LR 10^{-3}, weight-decay 10^{-4}, batch size 64, 10% validation split), and the training loss is cross-entropy plus a ρ_{util}=0.2 utilization term (weight λ_{util}=1) and a ρ_{sup}=0.05 selectivity term (weight λ_{sup}=1) that keeps the new expert's gate low on old-class prototypes. The frozen prototypes and, when used, the covariance matrices and the ridge accumulator are computed on the fit split; the head hyperparameters (τ, margin, ρs) are fixed across all datasets and seeds.

B Proofs of the claims in Section 3.2

The main text (Sec. 3.2) makes three claims: (i) any normalized router must drift (Eq. (2)); (ii) factorized gates are the only drift-free family; and (iii) hard top-1 routing over frozen gates leaves old-task features bit-identical (Eq. (3)). As a corollary of Proposition 1 we also record why alignment regularizers cannot restore the identity. We restate and prove the three claims, and then the corollary, here.

Proposition 1 (Impossibility of a drift-free normalized router)**.** *Let $g_e^{(K)}(q) = h_e(q)/\sum_{j=1}^{K} h_j(q)$ for non-negative scores h (softmax, sigmoid-normalized, sparsemax). If some score is non-constant, then for any $K \geq 2$ and any newly added expert with $h_{K+1}(q) > 0$ on a set of positive measure,*

$$\|g^{(K+1)} - g^{(K)}\|_1 = \frac{h_{K+1}(q)}{S_K + h_{K+1}(q)} > 0, \quad S_K = \sum_{j \leq K} h_j(q).$$

That is, a normalized router is never *drift-invariant: the identity of Eq. (1) fails on a set of positive measure.*

Proof. Appending expert $K{+}1$ changes the denominator from S_K to $S_K + h_{K+1}$, so every old gate is multiplied by $S_K/(S_K + h_{K+1}) < 1$. The total variation is $\sum_{e \leq K} g_e^{(K)}\big(1 - \frac{S_K}{S_K + h_{K+1}}\big) = \frac{h_{K+1}}{S_K + h_{K+1}}$, which is positive whenever the new expert fires on that input. The bound is tight when all old scores are equal. Hence normalized routing violates drift invariance on a set of positive measure whenever a new expert can fire, which, because the router is trained to use new experts (growing S_{K+1}), is the generic case. □

Corollary 1 (Regularization cannot restore the identity)**.** *The drift of Proposition 1 equals $h_{K+1}(q)/(S_K(q) + h_{K+1}(q))$: it is positive exactly where the appended expert fires and vanishes only where $h_{K+1}(q) = 0$. Any objective that leaves the appended expert active on a set of positive measure—in particular an alignment regularizer that penalizes gate change without switching the expert off—therefore leaves the drift positive there. Restoring Eq. (1) identically would force the new expert never to fire, which defeats the purpose of appending it.*

Proof. Immediate from Proposition 1. Regularizing the gates changes h_{K+1} only through the training objective; on every input where the trained new expert still fires ($h_{K+1} > 0$) the drift keeps the same strictly positive form. Hence no regularizer can make the identity hold globally while the new expert is used. □

Proposition 2 (Factorization is sufficient)**.** *Every* factorized *router $g_e^{(K)}(q) = \sigma_e(q; \theta_e)$ depending only on expert-e parameters is drift-invariant.*

Proof. Appending experts does not touch θ_e or q, so the factorized gate $g_e^{(K)}(q) = \sigma_e(q; \theta_e)$ is unchanged. □

Proposition 3 (Factorization is necessary under frozen parameters)**.** *Consider a router whose old parameters are never re-trained when experts are added (frozen-router setting, as in ZDEP). If such a router is drift-invariant (Eq. (1)) and its gates are differentiable in the new experts' parameters, then each gate $g_e^{(K)}$ depends only on θ_e.*

Proof. Under the frozen-parameter setting, the only quantities that change when an expert is appended are the newly added parameters $\{\theta_j\}_{j>K}$ and the pool size K. Drift invariance requires $g_e^{(K+1)}(q;\theta_1,\ldots,\theta_{K+1}) = g_e^{(K)}(q;\theta_1,\ldots,\theta_K)$ for *every* choice of θ_{K+1}. Differentiating both sides with respect to θ_{K+1} yields $\partial g_e^{(K+1)}/\partial\theta_{K+1} = 0$, so the gate does not depend on the newly added expert's parameters. Relabeling the appended expert as an arbitrary $j \neq e$ (it is the same operation, adding expert j last), this holds for every $j \neq e$; and since old parameters are frozen, $g_e^{(K)}$ cannot depend on θ_j for $j \neq e$ either. Hence the gate depends only on θ_e: the router is factorized. □

Proposition 4 (Factorization is necessary under re-training)**.** *Let the router parameters evolve under a differentiable update rule $\theta \leftarrow F(\theta, \mathcal{D}_t)$ after each task, and suppose drift invariance (Eq. (1)) holds for* every *data stream in an open set of possible streams. If the set of reachable updates $\{F(\theta,\mathcal{D}) - \theta\}$ spans a dense subset of the tangent space (the router is* generically re-trainable*), then for every old expert e and old input q, $\nabla_\theta g_e^{(K)}(q;\theta) = 0$; the gate is therefore locally constant in all parameters, so routing is only possible if expert e's own parameters are frozen and $g_e^{(K)}$ depends on no other expert.*

Proof. Take two data streams identical up to task K but differing arbitrarily thereafter, and let $\theta_t^{(1)}, \theta_t^{(2)}$ be the parameters at time t under each. Drift invariance for both streams gives $g_e^{(K)}(q;\theta_{K+1}^{(1)}) = g_e^{(K)}(q;\theta_{K+1}^{(2)})$. Writing $u^{(i)} = (F(\theta_K, \mathcal{D}_{K+1}^{(i)}) - \theta_K)/\eta$ and expanding to first order yields $\nabla_\theta g_e^{(K)}(q;\theta_K)\cdot(u^{(1)} - u^{(2)}) = 0$. By genericity the differences $u^{(1)} - u^{(2)}$ range over a dense subset of the tangent space, so $\nabla_\theta g_e^{(K)}(q;\theta_K) = 0$ for every old q and every K. A gate locally constant in all parameters cannot select its own expert unless the expert's key is frozen; with θ_e frozen, the argument of the preceding proposition (appending $j \neq e$ one at a time) gives $\partial g_e^{(K)}/\partial\theta_j = 0$ for all $j \neq e$. Hence $g_e^{(K)}$ depends only on the frozen θ_e: the router is factorized. □

Proposition 5 (Feature-level zero drift)**.** *With hard top-1 routing over per-expert sigmoid gates, frozen keys, and frozen LoRA experts, an old-task sample that still routes to its own expert e has bit-identical features at every later time: $f_t(x) = f_{\text{base}}(x) + \alpha W_e(x) B_e A_e x = f_{s_e}(x)$.*

Proof. Hard top-1 assigns $W_e(x) \in \{0,1\}$ depending only on frozen keys; the chosen expert's LoRA is frozen; adding experts changes neither W_e (independent columns) nor the frozen expert's output. Hence the mixture is unchanged whenever e still wins, so old-task features are bit-identical. □