NOUMENA RESEARCH

Making the Pretraining-to-Post-Training Boundary Observable in Large Mixture-of-Experts

Noumena

Technical Report

Abstract

Large Mixture-of-Experts (MoE) training is easy to misread because the usual monitoring stack reports symptoms, not the object being preserved or damaged. Loss, reward, entropy, and dead-expert counts can all move in sensible directions while leaving the operator unable to answer the lifecycle questions that matter most: when pretraining is done, whether post-training is still refining policy rather than rewriting capability, and when continued training is narrowing the routed knowledge base instead of improving it.

We argue that the missing object is geometric. RMS normalization factors residual states into angular position plus largely quotiented-out radial scale; on this angular manifold, expert down-projections define an overlapping immersion cover while the router induces a disjoint active-set stratification with swap boundaries. Together these form a layer-local MoE atlas. We then prove a local non-identifiability theorem: whenever an atlas-sensitive lifecycle margin varies off the span of a fixed family of output-only observables, nearby checkpoints can agree on those observables while differing in lifecycle state. Handoff readiness and refinement-versus-repurposing therefore cannot be identified from output-only surfaces alone near their decision boundaries.

We instantiate the theorem with concrete handoff and refinement margins, bridge them to exact empirical receipts, and make their falsifiers explicit. On the DeepSeek-V3 lineage, exact receipts calibrate a healthy atlas reference and identify a same-family stressed checkpoint with a modest same-canary scalar gap but materially worse overlap and occupancy structure. On live 640-expert distillation and SFT runs, scalar progress, overlap compatibility, occupancy, and resume-baseline semantics separate the claims that current monitoring conflates. The routed object has geometry, and lifecycle control requires receipts for that object.

Correspondence: research@noumena.com

1 Introduction

Large Mixture-of-Experts (MoE) models are attractive for exactly the reason that makes them difficult to monitor: they separate capacity from per-token computation. A 640-expert model with top-6 routing touches fewer than 1% of its expert parameters per token, yet the full expert ensemble still determines the model's knowledge capacity [1–5]. This separation creates a practical problem. When a large-MoE run improves, what exactly improved? Sometimes the answer is "the same routed object, only better trained." Often it is not.

The standard monitoring stack does not answer that question. Loss, router entropy, coefficient of variation of expert load, and dead-expert counts all report useful symptoms. They do not identify the preserved object itself. None of them tells the operator whether the model is still using the same neighborhoods of expert specialization, whether co-active experts remain mutually compatible, or whether a resumed job has earned the right to claim preservation. Those omissions matter in practice. They create two failure modes that look benign if one watches only the usual surfaces.

They also expose a broader gap in how we currently reason about frontier-model training. We still lack a precise way to say when pretraining has extracted what it can from a model family, whether RL or continual learning is still refining policy or has begun rewriting capability, and whether catastrophic forgetting has begun early enough that the right move is to return to pretraining. Those questions become more urgent in large MoE systems because the router is a load-bearing implementation detail. It is the interface that decides which knowledge-bearing experts are even in play. Keeping that interface stable is both technically difficult and operationally important.

The first is scalar improvement without structural recovery. In a later post-shift window of one of our 640-expert runs, expert weights and router weights are frozen, distillation loss falls from 5.25 to 4.48, and a conventional read would call that phase healthy. Across the same window the routed object changes materially: co-active overlap compatibility falls from a baseline of $+0.083$ to -0.063, dead experts rise from 42 to roughly 300, and routing entropy falls from 3.00 to 2.25. The run is still improving on a narrowed visited region while the broader expert structure degrades.

The second is the opposite mistake. After a cluster incident, a resumed 64-GPU SFT run comes back with zero dead experts, high entropy, and stable loss. Operationally, that is good news. Geometrically, it only establishes a new baseline. A resumed-checkpoint measurement establishes a new baseline and says nothing yet about preservation. The monitoring stack needs to know the difference.

This paper starts from that monitoring failure and asks what object is missing. Our answer is the *MoE atlas*: RMS normalization determines the angular geometry on which routing operates, the router determines which experts are co-active and which active sets compete, and expert down-projections determine the local coordinate structure on that geometry. Once that object is explicit, the lifecycle question becomes precise. If the lifecycle state depends on atlas motion and the chosen output-only observables do not span that direction, then two nearby checkpoints can look equally good by scalar criteria while justifying different stage transitions. The practical questions follow directly. Is post-training preserving the atlas or repurposing it? Is an apparent win actually broadening capability, or merely exploiting a narrower visited region? Is a run unstable because boundaries are noisy, because co-active overlaps no longer agree, or because occupancy has already collapsed? Those are the decisions this paper is trying to make legible. Concretely, the paper is aimed at four operational questions. First: when is a pretrained MoE stable enough, geometrically, for handoff to RL or other post-training? Second: during RL or continual learning, are we refining policy on top of the atlas or spending post-training compute to rewrite capability? Third: how far can we push post-training before atlas motion is large enough that catastrophic forgetting becomes a live risk? Fourth: can these questions be answered during the run, cheaply and repeatedly, without an expensive stop-the-world evaluation after the fact? These questions are related, but they do not carry equal weight in this paper. The handoff question is the center. The RL/CL question is the parallel extension of the same observability result. The forgetting and online-monitoring questions are consequences once those two boundaries are explicit.

We make four contributions:

1. **The angular manifold and tangent-dominated visible motion.** We show that RMS normalization [6] preserves angular position exactly and induces a canonical angular manifold $\mathbb{S}^{d-1}$ on which routed computation is most naturally described. The Fréchet derivative of the normalization map decomposes into a tangent component with gain $1/r_\epsilon$ and a radial component with gain ϵ/r_ϵ^3. The radial-to-tangent ratio is ϵ/r_ϵ^2, which is small whenever $\|x\|_2^2/d \gg \epsilon$. This condition holds empirically at all layers of trained models we have examined. Routed computation therefore depends primarily on angular position; radial magnitude contributes only through the attenuated correction term from Section 3.

2. **The MoE atlas: immersion cover, overlap relations, and active-set stratification.** Under a mild rank assumption ($\mathrm{rank}(W_1^{e,\ell}) = d$, standard when the expert intermediate dimension $h \geq d$) and a routing assumption (bias-free linear gate so that top-k membership descends to the angular manifold), we prove that expert down-projections define a canonical overlapping immersion cover on the regular routed region with C^∞ linear overlap relations, while the router induces a separate disjoint active-set stratification with discontinuous swap boundaries (Theorem 1). The constant-rank theorem then induces local intrinsic charts, though weights alone do not determine canonical intrinsic coordinates. This distinction between cover, cells, and swap boundaries is the geometric reason that load-balanced routing can still be structurally unhealthy. The geometric objects are determined entirely by the layer-ℓ weights as abstract subsets of $\mathbb{S}^{d-1}$; which states are actually visited depends on upstream transport and data (Section 4).

3. **A non-identifiability theorem for lifecycle control.** For any fixed finite family of output-only observables, we prove that if an atlas-sensitive lifecycle scalar varies along the corresponding output level set, then nearby checkpoints can agree exactly on the chosen output-only observables while differing in lifecycle state. As corollaries, concrete handoff and refinement margins built from smooth surrogates of overlap health, occupancy tail, and protected-layer drift cannot be identified from loss, reward, or benchmark surfaces alone near their respective decision boundaries whenever atlas motion lies outside the span of those surfaces (Section 6).

4. **Atlas health metrics and empirical validation of the theorem's regime.** We define four atlas-semantic drift functionals (coordinate, boundary, transition, plus content drift) that decompose structural health into transport, routing, and expert components. We introduce stratification occupancy as a mandatory adjunct for interpreting these metrics. On a 640-expert frozen-expert distillation run, we observe the predicted decoupling: scalar loss improves while overlap compatibility worsens across four consecutive geometry gates and routing occupancy remains materially degraded. On a separate SFT run, we show that operational recovery from a cluster incident is only operational recovery. It does not count as geometric evidence until a post-baseline preservation gate is passed (Section 7).

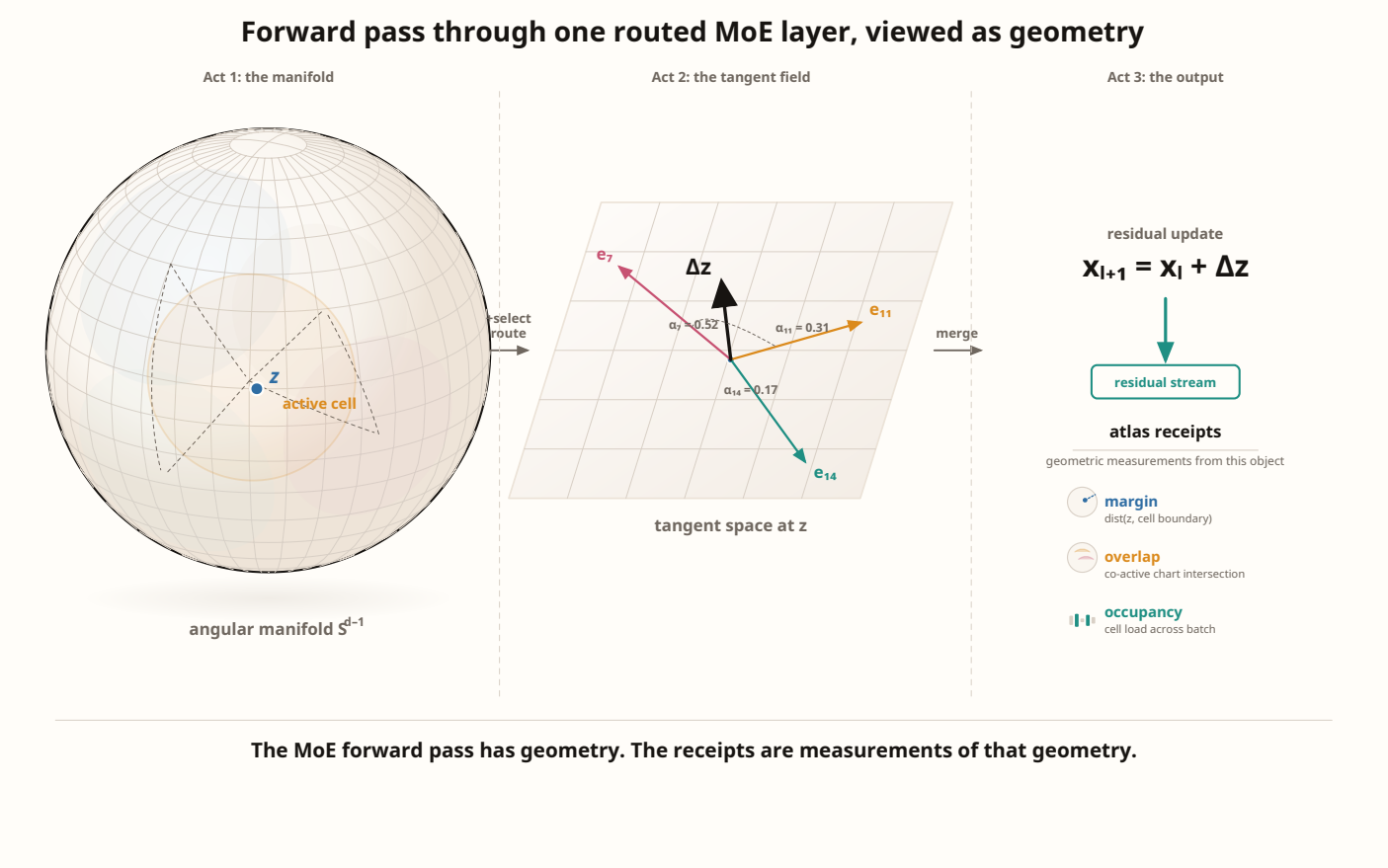

Figure 1 Forward pass of one top-k MoE layer. A normalized token state enters one active-set cell, only the selected expert charts contribute locally, and their merged field returns a routed update. The atlas receipts are read from this routed object, not from expert weights alone.

2 Background and Notation

We establish the notation and machinery needed for the atlas construction. All claims in this section concern a single pre-norm routed MoE layer ℓ within a transformer. This section isolates the few ingredients that matter operationally: the normalized state seen by the router, the top-k decision that creates the active set, and the expert maps that define the local computation. Once those pieces are written down cleanly, the geometry is hard to avoid. This section is doing a practical job as much as a formal one. If the reader cannot tell, from the equations alone, which surfaces can move the atlas and which ones can only move the readout around it, then the monitoring story later in the paper will sound more clever than useful.

2.1 RMS normalization

Given a residual state $x \in \mathbb{R}^d$ entering a pre-norm routed block, RMS normalization [6] computes:

$$r_\epsilon(x) := \sqrt{\frac{\|x\|_2^2}{d} + \epsilon}, \tag{1}$$

$$z_\epsilon(x) := \frac{x}{r_\epsilon(x)}, \tag{2}$$

where $\epsilon > 0$ is a small stability constant (typically 10^{-6} to 10^{-8}). The normalized state is then rescaled by a learned diagonal gain $\Gamma = \mathrm{diag}(\gamma)$ to produce the gate input:

$$y_\gamma(x) := \Gamma\, z_\epsilon(x). \tag{3}$$

These equations look innocuous, which is one reason the geometry is easy to miss. RMS normalization is usually treated as plumbing. For a routed model, it is the map that decides which part of state space the router and experts can even see. That is already enough to make the usual monitoring story look incomplete. If the routed system only ever sees the normalized state, then any account of MoE health written purely in terms of scalar loss or aggregate expert counts is leaving out the space in which the routing decision actually lives.

Three facts are immediate from these definitions. First, the angular representative $\hat{x} = x/\|x\|_2$ is exactly preserved: $\widehat{z_\epsilon(x)} = \hat{x}$ for all $\epsilon \geq 0$ and all nonzero x. This is because z_ϵ is a positive scalar rescaling along rays. Second, when $\epsilon = 0$, the normalized state lies on an exact sphere: $\|z_0(x)\|_2 = \sqrt{d}$ for all nonzero x. Third, when $\epsilon > 0$, the image $z_\epsilon(\mathbb{R}^d \setminus \{0\})$ forms a radial shell instead of a single sphere. The angular projection $q : z \mapsto z/\|z\|_2$ collapses this shell onto $\mathbb{S}^{d-1}$ by identifying points that differ only by positive radial scale. Accordingly, all smooth-manifold statements below are made on the angular quotient $\widehat{\mathcal{M}}_\ell = \mathbb{S}^{d-1}$, while shell-valued states are used only when exact magnitudes or gate coordinates matter.

The gain Γ introduces an anisotropic deformation. Define the gain anisotropy $\kappa_\Gamma := \sigma_{\max}(\Gamma)/\sigma_{\min}(\Gamma)$. Pre-gain and post-gain neighborhoods are then bilipschitz with constant κ_Γ: for any two normalized states z, z',

$$\sigma_{\min}(\Gamma)\,\|z - z'\|_2 \;\leq\; \|\Gamma z - \Gamma z'\|_2 \;\leq\; \sigma_{\max}(\Gamma)\,\|z - z'\|_2. \tag{4}$$

When $\kappa_\Gamma \approx 1$ (which we observe empirically on most routed layers), pre-gain coordinates are faithful proxies for post-gain gate geometry. When κ_Γ is large, exact gate analysis requires working in post-gain coordinates. In all experiments below, we report whether diagnostics use pre-gain or post-gain coordinates.

2.2 Top-k sparse routing

A router at layer ℓ computes logits $\ell_e = w_e^\top y$ for each of E experts, where $y = y_\gamma(x)$ is the gate input. The top-k active set is:

$$R_k(y) := \{e_1, \ldots, e_k\} \quad \text{where} \quad \ell_{e_1} \geq \cdots \geq \ell_{e_k} \geq \ell_{e_{k+1}} \geq \cdots \tag{5}$$

and the routed output is:

$$\mathrm{MoE}(y) = \sum_{e \in R_k(y)} \rho_e(y) \cdot E_e(y), \tag{6}$$

where ρ_e are softmax-normalized weights over the active set and E_e is the e-th expert feed-forward network. Each expert typically computes $E_e(y) = W_2^e\, \sigma(W_1^e\, y)$ where $W_1^e \in \mathbb{R}^{h \times d}$ is the down-projection, σ is an activation function (SiLU, GELU, or SwiGLU variant [7]), and $W_2^e \in \mathbb{R}^{d \times h}$ is the up-projection. In all architectures we study, $h > d$ (the expert intermediate dimension exceeds the model dimension), which is the standard regime.

2.3 Existing diagnostics and their limitations

Standard MoE monitoring [3, 8] tracks several per-step statistics:

- **Router entropy**: $H(\rho) = -\sum_e p_e \log p_e$ where p_e is the fraction of tokens routed to expert e. Low entropy indicates concentration.
- **Coefficient of variation (CV)**: $\text{CV} = \sigma(\text{load})/\mu(\text{load})$ where load is the token count per expert. High CV indicates imbalance.
- **Dead expert count**: the number of experts receiving zero tokens over a measurement window.
- **Scalar training loss**: cross-entropy, distillation loss, or task-specific objective.

These statistics are useful but have a fundamental limitation: they operate at the level of *aggregate* routing behavior and miss the *geometric structure* that routing creates. We can illustrate this concretely. Consider two checkpoints of a 640-expert model with top-6 routing. Checkpoint A has entropy 6.1, CV 48, dead experts 0, loss 6.50. Checkpoint B has entropy 6.0, CV 51, dead experts 0, loss 6.45. For the narrow question "did the current objective get cheaper while aggregate routing stayed calm?", standard monitoring would read B as slightly better. But suppose that between A and B, the backbone has shifted which angular regions of the sphere are visited, so that 30% of tokens now activate different expert subsets while maintaining the same aggregate load distribution. The expert specialization neighborhoods (which tokens see which experts as co-active partners) have been substantially rewritten, even though the aggregate statistics barely moved.

This concern is concrete. In the Distill-640 experiment we report below (Section 7.3), the router is *frozen* and the aggregate routing statistics are approximately constant across checkpoints, yet the co-active overlap compatibility deteriorates by 0.146 in absolute terms. The deterioration is driven entirely by changes in the trainable backbone (attention and embeddings) that alter which angular regions of the sphere are visited. No scalar or aggregate routing statistic catches this. This is the sort of result that forces a field to grow up. As long as the monitoring stack is asked only to summarize utilization, it will keep missing failures that live in the compatibility structure being utilized.

What is missing is a decomposition of structural health that separates *what changed* (which component of the atlas) from *how much changed* (the magnitude of drift in that component). The atlas framework we introduce below provides exactly this decomposition. It separates structural health into four components (coordinate, boundary, transition, and content), each corresponding to a different part of the geometric object that standard statistics average over. And it introduces stratification occupancy as a mandatory adjunct without which the other four metrics can be misleading. This is the same progression that happened in sparse-model systems work more broadly: first we got scalar and throughput counters, then we learned the hard way that the real bottlenecks lived in the structure those counters were averaging over.

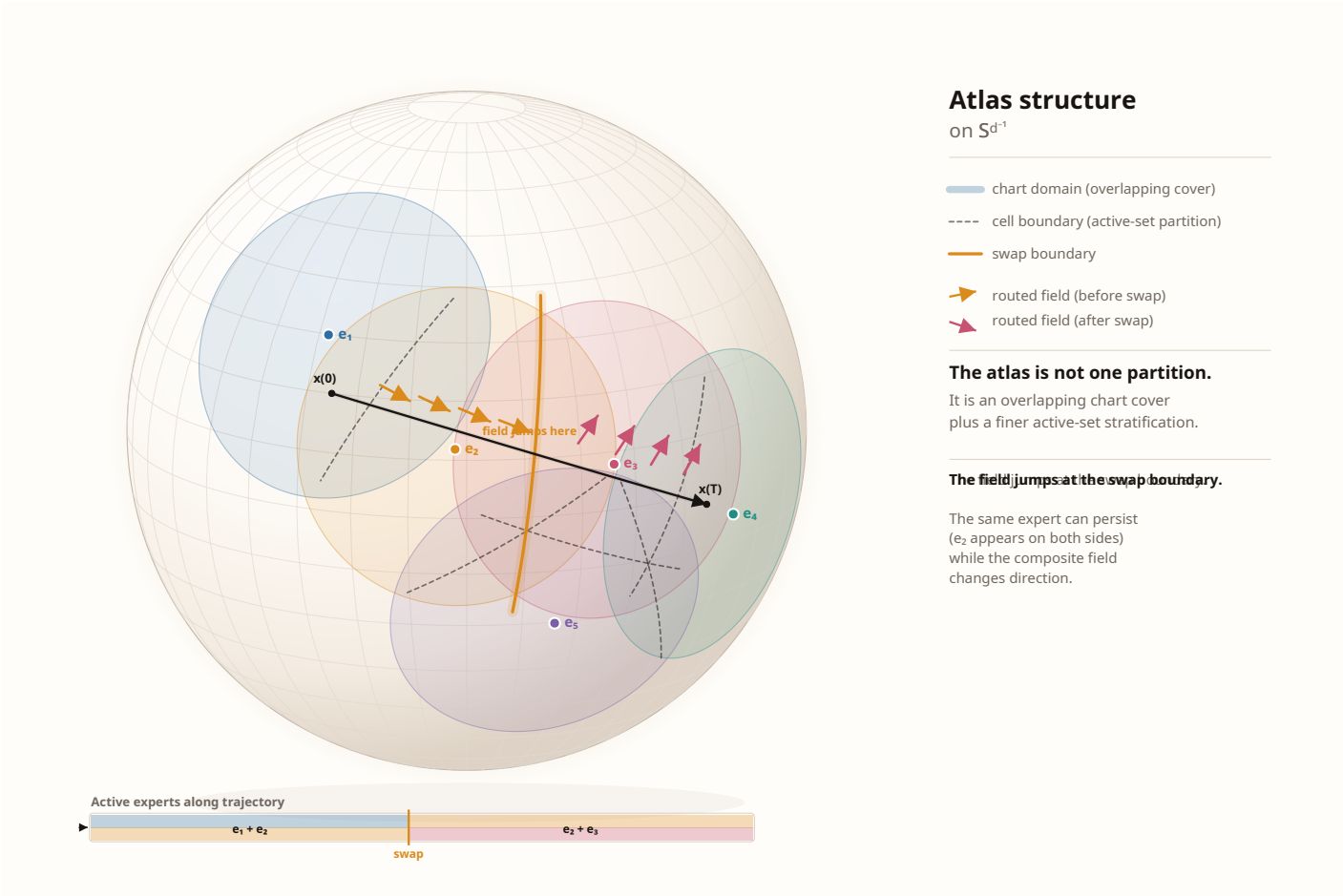

Figure 2 The MoE atlas on the angular manifold $\widehat{\mathcal{M}}_\ell = \mathbb{S}^{d-1}$. **(a)** Expert chart domains (colored patches) overlap where experts are co-active; active-set cells (dashed boundaries) partition the angular regular routed region by exact top-k membership. A swap boundary (solid curve) marks where one expert exits and another enters the active set. **(b)** Local view across a swap boundary. Expert 2 remains active in both adjacent cells; the e_2 contribution persists across the boundary. The composite field jumps because expert 1 is replaced by expert 3; the magnitude of that jump is the transition incompatibility $\|\Delta V\|$. The solid and dashed trajectories show the small-ΔV and large-ΔV regimes respectively.

3 The Angular Manifold

We now formalize the geometry induced by RMS normalization. The key result is that visible motion after renormalization is tangent-dominated: the routed block operates primarily on angular position on $\mathbb{S}^{d-1}$, with radial sensitivity attenuated by an explicit factor. This is the first place where the paper stops being a metaphor and becomes an engineering claim. If tangent motion dominates and radial motion is heavily attenuated, then the natural state variable for routing is the direction on the angular manifold together with the post-gain deformation seen by the gate.

3.1 Ray invariance and the angular projection

Proposition 1 (Ray invariance). *For any nonzero $x \in \mathbb{R}^d$ and $c > 0$, $z_\epsilon(cx)$ is a positive scalar multiple of $z_\epsilon(x)$. In particular, $\widehat{z_\epsilon(cx)} = \widehat{z_\epsilon(x)} = \hat{x}$. When $\epsilon = 0$, $z_0(cx) = z_0(x)$ exactly.*

The proof is immediate: $r_\epsilon(cx) = \sqrt{c^2\|x\|_2^2/d + \epsilon}$, so $z_\epsilon(cx) = cx/r_\epsilon(cx) = (c\, r_\epsilon(x)/r_\epsilon(cx))\, z_\epsilon(x)$, and the scalar $c\, r_\epsilon(x)/r_\epsilon(cx) > 0$. When $\epsilon = 0$, this scalar equals 1.

This means that RMS normalization quotients out positive scale. The angular representative $\hat{x}$ is the fundamental state variable for routing: two states on the same positive ray produce the same angular representative and therefore the same routing decision (when the router gate is bias-free linear, as we assume below). That point is easy to state and easy to underestimate. If routing is fundamentally angular, then Euclidean intuitions about "how far the state moved" are already the wrong language for much of the problem.

All smooth-manifold statements in the rest of this paper are made on $\widehat{\mathcal{M}}_\ell = \mathbb{S}^{d-1}$ via the angular projection $q : z_\epsilon(x) \mapsto z_\epsilon(x)/\|z_\epsilon(x)\|_2 = \hat{x}$.

3.2 The Fréchet differential and tangent dominance

The next question is dynamic: if the routed block applies a residual update u to produce $x' = x + u$, which part of u survives as first-order visible motion after the next renormalization step?

Proposition 2 (Tangent-dominated visible motion)**.** *The Fréchet derivative of z_ϵ at nonzero x is:*

$$\mathrm{D}z_\epsilon(x)[u] = \frac{1}{r_\epsilon(x)} \left(I - \frac{z_\epsilon(x)\, z_\epsilon(x)^\top}{d} \right) u. \tag{7}$$

Decomposing $u = u_\mathsf{T} + u_{\text{rad}}$ where $u_{\text{rad}} = (\hat{x}^\top u)\hat{x}$ and $u_\mathsf{T} = u - u_{\text{rad}}$:

$$\mathrm{D}z_\epsilon(x)[u] = \frac{1}{r_\epsilon(x)}\, u_\mathsf{T} + \frac{\epsilon}{r_\epsilon(x)^3}\, u_{\text{rad}}. \tag{8}$$

The tangent gain is $1/r_\epsilon$. The radial gain is ϵ/r_ϵ^3. Their ratio is:

$$\frac{\textit{radial gain}}{\textit{tangent gain}} = \frac{\epsilon}{r_\epsilon(x)^2} = \frac{\epsilon}{\|x\|_2^2/d + \epsilon}. \tag{9}$$

The proof is a direct computation from the quotient rule; see Appendix A for the full derivation. The important consequence is quantitative. For a typical trained model with $d = 2048$ and $\epsilon = 10^{-6}$, if the RMS norm of the residual state is $\|x\|_2/\sqrt{d} \approx 1$ (a reasonable order of magnitude), then $r_\epsilon(x) \approx 1$ and the radial-to-tangent ratio is approximately 10^{-6}. Even in pathological cases where the residual stream is unusually small (e.g., $\|x\|_2/\sqrt{d} \approx 0.01$), the ratio is still $\approx 10^{-2}$. In every model we have examined, radial sensitivity is attenuated by at least two orders of magnitude relative to tangential sensitivity.

This justifies analyzing expert updates as directional fields on $\mathbb{S}^{d-1}$; treating them as arbitrary vectors in $\mathbb{R}^d$ throws away the geometry that matters. For a raw expert output $u_e(z)$, we define the *directional visible field*:

$$v_e(z) := P_z\, u_e(z), \qquad P_z := I - \frac{z\, z^\top}{d}, \tag{10}$$

which strips the shared $1/r_\epsilon$ scalar. Since all experts at the same state z share this scalar, relative comparisons between expert fields (including transition incompatibility and the perturbation budget below) are unaffected by the convention. Absolute visible step sizes require restoring the $1/r_\epsilon$ factor. This convention is worth being explicit about because it buys a simpler language without changing

the comparisons that matter. When we later compare two experts at the same anchor state, the relevant question is "do these experts point in compatible directions on the routed manifold?"

At $\epsilon = 0$, P_z is the exact orthogonal projector onto $T_z\mathbb{S}^{d-1}_{\sqrt{d}}$. At finite ϵ, the bracketed map in Eq. (7) is not an exact projector, but the deviation is controlled by the ratio in Eq. (9) and is negligible in practice. The practical value of writing this out is that it tells us what kind of disagreement between experts should worry us. If two experts disagree mostly in a radial direction that the next normalization step will largely suppress, the disagreement is far less consequential than if they disagree tangentially on the manifold the router and the next block will actually act on.

3.3 Consequences for routed computation

The tangent-dominance result has three direct consequences for how MoE components should be analyzed:

Expert updates are directional fields. An expert computes a raw residual update $u_e(z)$ in $\mathbb{R}^d$, but the first-order *visible* effect after renormalization is the tangent-dominated field $v_e(z)$. Experts should therefore be analyzed as local update fields on the angular manifold. Treating them as arbitrary Euclidean functions on unnormalized residual space misses the routed geometry.

Routing boundaries are angular boundaries. The gate input is $y = \Gamma z_\epsilon(x)$, so routing decisions are made in post-gain normalized coordinates. Raw residual space is upstream of that decision surface. When later sections refer to chart boundaries, they mean boundaries in these operational coordinates.

Knowledge neighborhoods are directional. If pretraining builds an atlas on this geometry, then local knowledge should be associated with directional neighborhoods on $\mathbb{S}^{d-1}$ and with the transition structure between them. Catastrophic forgetting is more naturally framed as drift of chart-local fields and chart-transition semantics than as generic Euclidean norm drift in parameter space.

4 The MoE Atlas

We now characterize the geometric object that the combination of RMS normalization and top-k routing creates. The construction builds on the angular manifold from Section 3 and produces three coupled components: an overlapping expert cover, a disjoint active-set stratification, and transition structure at swap boundaries. Putting these three in one sentence is useful because most of the confusion in practice comes from collapsing them into one object. The overlapping cover is where compatibility lives. The disjoint cell stratification is where exact routed membership lives. The swap boundaries are where discontinuity enters. If the paper succeeds at anything, it should make those three objects hard to confuse again.

4.1 Regular routed region and active-set cells

For a routed layer ℓ with E experts and top-k routing, the router logits partition $\widehat{\mathcal{M}}_\ell = \mathbb{S}^{d-1}$ into regions by active set.

We assume the router logits are bias-free linear in the post-gain coordinate: $\ell_e(y) = w_e^\top \Gamma z$. Because Γ is invertible and the logits are linear, the top-k active set is determined by strict inequalities

among continuous functions of $\hat{z}$, so the top-k membership descends to the angular manifold and is constant along positive rays.

Define the *regular routed region* $\widehat{\mathcal{M}}_\ell^{\text{reg}}$ as the set of angular states where the top-k active set is locally constant, i.e. states away from routing tie surfaces where the k-th and $(k{+}1)$-th logits are equal. In non-degenerate models, these tie surfaces are algebraic hypersurfaces defined by equality of continuous logits and therefore have measure zero; hence $\widehat{\mathcal{M}}_\ell^{\text{reg}}$ is open and dense in $\mathbb{S}^{d-1}$.

For each k-element subset $S \subseteq \{1, \dots, E\}$, the *active-set cell* is:

$$\widehat{C}_{S,\ell} := \{\hat{z} \in \widehat{\mathcal{M}}_\ell^{\text{reg}} : R_k(\Gamma \hat{z}) = S\}. \tag{11}$$

The cells $\{\widehat{C}_{S,\ell}\}_{|S|=k}$ are disjoint and partition $\widehat{\mathcal{M}}_\ell^{\text{reg}}$ by exact active set.

For expert e, the *regular chart domain* is:

$$\widehat{U}_{e,\ell}^{\text{reg}} := \bigcup_{S \ni e} \widehat{C}_{S,\ell}, \tag{12}$$

the union of all cells in which expert e participates. At generic points away from higher-order tie loci, each angular state belongs to exactly k chart domains.

The distinction between chart domains and cells is fundamental and will recur throughout the experiments. Chart domains *overlap*: experts 1 and 2 are both active in cell $C_{\{1,2,3,4,5,6\}}$ (a top-6 example), and expert 1 is also active in cell $C_{\{1,2,3,4,5,7\}}$. Cells are *disjoint*: the region where exactly experts $\{1, 2, 3, 4, 5, 6\}$ are active is separate from the region where exactly $\{1, 2, 3, 4, 5, 7\}$ are active. Co-active overlap compatibility concerns the expert cover (do co-active experts produce compatible fields?). Swap-boundary behavior concerns the cell partition (what happens when crossing from one cell to a neighboring one?).

4.2 Immersion cover and linear overlap relations

We now state the main geometric result. One might reasonably ask whether "atlas" is just a fancy word for "the experts have different weights." It is not. The theorem below does two things. First, it shows there is a genuine smooth structure on the regular routed region: the expert down-projections are immersions with linear overlap relations, more than a collection of independent functions that happen to coexist. Second, it shows exactly where the smooth story stops: at swap boundaries where the active set changes. The existence of both the smooth interior and the discontinuous boundary is what makes the object an atlas rather than a partition or a soft mixture.

Theorem 1 (MoE Atlas)**.** *Assume* $\text{rank}(W_1^{e,\ell}) = d$ *for each expert* e *(standard when* $h \geq d$*), and assume the router logits are bias-free linear in the post-gain coordinate so that top-*k *membership descends to* $\mathbb{S}^{d-1}$*. Define the chart map* $\widehat{\phi}_{e,\ell} := W_1^{e,\ell}|_{\widehat{U}_{e,\ell}^{\text{reg}}} : \widehat{U}_{e,\ell}^{\text{reg}} \to \mathbb{R}^h$*. Then:*

1. *The regular chart domains* $\{\widehat{U}_{e,\ell}^{\text{reg}}\}$ *form an overlapping open cover of* $\widehat{\mathcal{M}}_\ell^{\text{reg}}$*, with each angular state belonging to exactly* k *domains at generic points.*
2. *The active-set cells* $\{\widehat{C}_{S,\ell}\}_{|S|=k}$ *form a disjoint partition of* $\widehat{\mathcal{M}}_\ell^{\text{reg}}$*.*
3. *Each* $\widehat{\phi}_{e,\ell}$ *is a smooth immersion with rank-*$(d{-}1)$ *Jacobian on* $T_{\hat{z}}\mathbb{S}^{d-1}$*.*

4. *On co-active overlaps $\widehat{U}^{\mathrm{reg}}_{e,\ell} \cap \widehat{U}^{\mathrm{reg}}_{e',\ell}$, the coordinate relation*

$$T_{e\to e',\ell} := W_1^{e',\ell}\,(W_1^{e,\ell})^{+} \tag{13}$$

is linear, hence C^∞. Here $(W_1^{e,\ell})^+$ denotes any left inverse; the result is independent of the choice because $y = W_1^{e,\ell}\hat{z}$ lies in the column space of $W_1^{e,\ell}$.

5. *The immersion cover induces local intrinsic charts on $\widehat{\mathcal{M}}^{\mathrm{reg}}_{\ell}$ via the constant-rank theorem (with non-canonical intrinsic coordinates).*

All objects are determined entirely by the layer-ℓ weights as abstract subsets of $\mathbb{S}^{d-1}$.

Proof. Because the router logits are continuous and positively homogeneous of degree one (bias-free linear gate composed with continuous gain), top-k membership is constant along positive rays and descends to the angular manifold. The strict inequalities defining the top-k set are open conditions on $\mathbb{S}^{d-1}$, so each angular cell is open in $\widehat{\mathcal{M}}^{\mathrm{reg}}_{\ell}$ and the cells partition it by exact active set. Every $\hat{z} \in \widehat{\mathcal{M}}^{\mathrm{reg}}_{\ell}$ belongs to the k chart domains indexed by its active set, proving the overlapping cover.

The map $\widehat{\phi}_{e,\ell}$ is the restriction of a rank-d linear map to $\mathbb{S}^{d-1}$, hence smooth. Since $\ker W_1^{e,\ell} = \{0\}$ (rank d with $h > d$), the Jacobian restricted to any $(d-1)$-dimensional subspace of $\mathbb{R}^d$ has rank $d-1$; in particular on $T_{\hat{z}}\mathbb{S}^{d-1}$.

For the overlap relation: if $y = W_1^e \hat{z}$ on a co-active overlap, then $\hat{z} = (W_1^e)^+ y$ on the column space and $W_1^{e'}\hat{z} = W_1^{e'}(W_1^e)^+ y$, which is linear in y.

By the constant-rank theorem, $\widehat{\phi}_{e,\ell}$ is a smooth embedding on a neighborhood of each point, so the embedded image is a smooth $(d-1)$-submanifold of $\mathbb{R}^h$. Local coordinates on the image compose with $\widehat{\phi}^{-1}$ to give intrinsic charts. On overlaps, intrinsic transition maps factor through a smooth local inverse, the linear map $T_{e\to e'}$, and a smooth local projection, so the composition is smooth. Weight-determinacy: all objects are defined from the layer-ℓ weights; which states are visited depends on upstream transport and data. □

It is worth pausing to note what is classical here and what is not. The induced intrinsic atlas on co-active overlaps is classical differential geometry: immersions, constant-rank theorem, smooth transition maps. What is *not* classical is the hard-routing behavior at swap boundaries: when the top-k set changes, the composite visible field is discontinuous. The atlas lives on the regular routed region where the active set is locally constant; the swap boundaries are the complement. This split is more than a matter of taste. It is the reason two very different pathologies have to be separated in practice. Co-active overlap failure is a failure of local compatibility. Swap-boundary failure is a failure of transition behavior. Treating them as one problem is convenient; it is also wrong. It is also one of the reasons large-MoE debugging feels slippery in practice. Teams often see a bad run, look at one routing number, and then argue about whether they are seeing "collapse," "noise," or "instability." Usually they are mixing two different geometric failures and asking one statistic to referee both.

Shared-expert routing. In architectures with shared experts that are always active [5, 9], the shared experts contribute a globally present baseline field and only the remaining experts participate in sparse competition. The theorem applies to the routed subset, with the shared-expert contribution treated as part of the transport baseline.

4.3 Transition incompatibility

At a swap boundary between neighboring active-set cells $\widehat{C}_S$ and $\widehat{C}_{S'}$ where $S' = (S \setminus \{e_k\}) \cup \{e_{k+1}\}$, the composite visible field jumps by:

$$\Delta V(z) = \rho_b(z)\big(v_{e_{k+1}}(z) - v_{e_k}(z)\big), \tag{14}$$

where ρ_b is the shared routing weight at the swap boundary. This identity holds under a one-swap idealization: exactly one expert exits and one enters, all other routing weights and expert fields are held fixed, and no higher-order renormalization terms are included. In practice, multi-expert swaps, renormalization corrections, and nonlocal weight changes introduce additional terms beyond this identity. Nevertheless, the norm $\|\Delta V(z)\|_2$ is the natural local measure of transition incompatibility suggested by the geometry.

4.4 Perturbation budget

Under adaptation from checkpoint θ_0 to θ_1, the change in the composite visible field at a fixed anchor state z_0 satisfies:

$$\|\Delta V_\ell(z_0)\|_2 \;\le\; \underbrace{\sum_e |\Delta\rho_e(z_0)|\,\|v_e^{(\theta_0)}(z_0)\|_2}_{\text{router drift}\times\text{expert magnitude}} \;+\; \underbrace{\sum_e \rho_e^{(\theta_1)}(z_0)\,\|\Delta v_e(z_0)\|_2}_{\text{routing weight}\times\text{content drift}}. \tag{15}$$

This is a triangle-inequality upper bound with no tightness claim. Router drift and content drift contribute additively: controlling one term can compensate for looseness in the other, but only at the level of the bound itself. Whether this bound is tight or practically useful as a control law is an empirical question.

4.5 Stratification occupancy

The atlas can be formally well-defined while the visited occupancy measure collapses. For a probe family $\mathcal{P}$ and routed layer ℓ, define:

$$\mu_{\mathcal{P},\ell}(S) := \Pr_{z\sim\mathcal{P}}[R_k(\Gamma z) = S], \tag{16}$$

$$\pi_{\mathcal{P},\ell}(e) := \Pr_{z\sim\mathcal{P}}[e \in R_k(\Gamma z)] = \sum_{S\ni e} \mu_{\mathcal{P},\ell}(S). \tag{17}$$

Standard telemetry (CV, minimum entropy, dead expert count, active expert mean) provides low-dimensional proxies for the degeneracy of π or μ. Occupancy is not part of the atlas itself; it is a property of how the atlas is *used* under a given data distribution. But it is necessary for interpreting atlas-health measurements, because two critical failure modes depend on it:

Occupancy degeneration. The atlas can remain formally well-defined while the visited occupancy measure collapses onto a small subset of active-set cells. When this happens, large portions of the chart cover carry negligible empirical support. Scalar objectives can continue to improve on the narrowed visited region while the effective atlas loses coverage. This is a distinct failure mode from chart-content failure (experts computing bad fields) or boundary failure (incompatible fields at swap boundaries): the charts themselves may be locally coherent and mutually compatible, but the model no longer visits enough of them.

Margin non-diagnosticity. Rising boundary margins (larger gaps between the k-th and $(k+1)$-th logits) might appear to indicate healthier routing. But under occupancy degeneration, larger margins can simply mean that the surviving routing competitions are becoming more decisive inside a degraded stratification while the aggregate geometry keeps getting worse. We observe this directly in the Distill-640 experiments below.

5 Atlas Health Metrics

The atlas structure from Section 4 implies a natural decomposition of structural health into four independently computable components. We define these as drift functionals on a fixed probe family $\mathcal{P}$ comparing a base checkpoint θ_0 and a comparison checkpoint θ_1. We use four functionals because there are four places the structure can change in operationally distinct ways. Adding more numbers without adding a new control question would only make the monitor harder to use. Using fewer would collapse together failure modes that need different interventions. That last point is worth stating directly. Monitoring papers often fail by offering a bag of clever surfaces and then asking the reader to admire the bag. We are trying to do the opposite. Each functional is here because removing it would erase a control question that came up in the runs.

All comparisons bootstrap over windows rather than tokens, because tokens within a window share sequence context and are not statistically independent.

For each probe token t and protected layer ℓ, let $z_{t,\ell}^{(\theta)}$ denote the canonical pre-gain operational state and $g_{t,\ell}^{(\theta)} := \Gamma_\ell^{(\theta)} z_{t,\ell}^{(\theta)}$ the exact post-gain gate coordinate. Define the base-anchored state $z_{t,\ell}^{(0)} := z_{t,\ell}^{(\theta_0)}$ and checkpoint-native gate coordinates: $g_{t,\ell}^{(0\to j)} := \Gamma_\ell^{(\theta_j)} z_{t,\ell}^{(0)}$ for $j \in \{0, 1\}$.

Coordinate drift (transport). $\Delta_{\text{coord}}(\ell) := \mathbb{E}_t[1 - \cos(g_{t,\ell}^{(\theta_0)}, g_{t,\ell}^{(\theta_1)})]$. This measures how much the operational state path has shifted between checkpoints. Large coordinate drift means the attention-and-backbone stack is sending tokens to different regions of the atlas.

Boundary drift (routing). $\Delta_{\text{boundary}}(\ell) := \mathbb{E}_t[1 - |R_k^{(\theta_0)}(g_{t,\ell}^{(0\to 0)}) \cap R_k^{(\theta_1)}(g_{t,\ell}^{(0\to 1)})|/k]$. This holds fixed the base canonical state, pushes it through each checkpoint's own gain, and measures whether the same experts are selected. It isolates routing changes from path changes.

Transition drift (routing). $\Delta_{\text{transition}}(\ell) := \mathbb{E}_t[D_{\text{JS}}(\rho_{\theta_0}(g_{t,\ell}^{(0\to 0)}), \rho_{\theta_1}(g_{t,\ell}^{(0\to 1)}))]$. This measures the JS divergence of the full router output, capturing routing *weight* changes even when the active set is preserved.

Content drift (expert). $\Delta_{\text{content}}(\ell) := \mathbb{E}_t \,\text{Quantile}_{q;\,e}[\|\widetilde{u}_e^{(\theta_1)} - \widetilde{u}_e^{(\theta_0)}\|_2 / (\|\widetilde{u}_e^{(\theta_1)}\|_2 + \|\widetilde{u}_e^{(\theta_0)}\|_2 + \epsilon)]$ where $q = 0.9$ to be sensitive to high-drift tails.

These decompose atlas health into transport (coordinate), routing (boundary + transition), and expert (content) components.

Occupancy as mandatory adjunct. These four functionals measure geometric change, but their interpretation requires healthy stratification occupancy. Two cautions apply. First, boundary-margin improvements are not health certificates under occupancy degeneration. Second, scalar

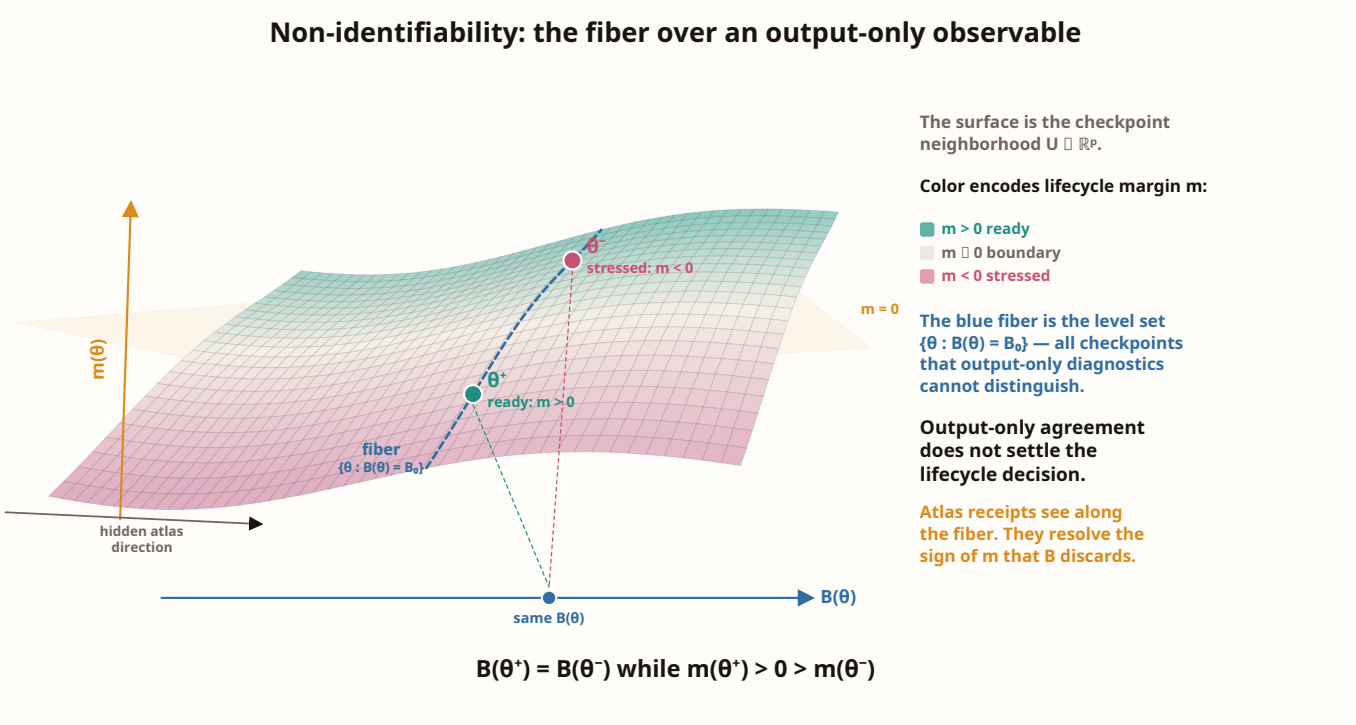

Figure 3 Intuition for the non-identifiability theorem. Two nearby checkpoints can agree on loss, reward, or benchmark score while landing on opposite sides of an atlas-sensitive lifecycle margin such as handoff readiness $h(\theta)$ or refinement-versus-repurposing $r(\theta)$. Output-only agreement therefore does not settle the lifecycle decision; atlas receipts reopen that distinction.

objectives can improve without geometric recovery. A convincing positive turn requires overlap compatibility *and* occupancy to recover jointly.

6 Why Output-Only Observables Cannot Identify Lifecycle State

The atlas exists independently of any particular monitoring stack. The next question is stronger and more dangerous: can the lifecycle questions that matter in practice be answered from output-only surfaces alone? By "output-only" we mean any finite family of observables that depends only on model outputs on a fixed evaluation family: loss, reward, benchmark scores, or similar scalar summaries. The theorem below says that the answer is no whenever the relevant lifecycle scalar varies along an output level set. The point is observability. If two checkpoints can differ in lifecycle state while agreeing on the chosen output-only surfaces, then no amount of confidence in those surfaces will make them answer the missing question. For the theorem itself we work with C^1 observables. When practitioners report discrete leaderboard metrics or exact pass/fail rates, those are coarser than the differentiable surrogates and averaged losses that the theorem analyzes, and therefore carry less information.

Let $U \subseteq \mathbb{R}^p$ be a checkpoint neighborhood around a reference checkpoint θ_0. Let

$$\mathcal{B} : U \to \mathbb{R}^m$$

be a C^1 map of output-only observables computed on a fixed evaluation family (for example, training loss, held-out loss, reward, or a finite benchmark vector). Let

$$\mathcal{A} : U \to \mathbb{R}^r$$

be a C^1 receipt map computed on a fixed probe family, whose components may include atlas-sensitive drift functionals together with any occupancy adjuncts required by the lifecycle decision. An *atlas-sensitive lifecycle scalar* is any C^1 scalar $\alpha = \psi \circ \mathcal{A}$ for some C^1 map $\psi : \mathbb{R}^r \to \mathbb{R}$. Examples include a handoff margin, a protected-layer preservation margin, or a refinement budget relative to a base checkpoint.

Theorem 2 (Local non-identifiability of atlas-sensitive lifecycle state from output-only observables)**.** *Fix $\theta_0 \in U$. Assume:*

1. *$\mathcal{B}$ has constant rank $q < p$ on a neighborhood of θ_0;*
2. *the atlas-sensitive scalar α satisfies*

$$d\alpha_{\theta_0} \notin \text{rowspan}(J_{\mathcal{B}}(\theta_0)) .$$

Then for every neighborhood V of θ_0, there exist checkpoints $\theta^+, \theta^- \in V$ such that

$$\mathcal{B}(\theta^+) = \mathcal{B}(\theta^-) = \mathcal{B}(\theta_0),$$

while

$$\alpha(\theta^+) > \alpha(\theta_0) > \alpha(\theta^-).$$

In particular, no decision rule that factors only through $\mathcal{B}$ can determine the local value of α near θ_0.

Proof. Because $\mathcal{B}$ has constant rank q near θ_0, the constant-rank theorem implies that the level set

$$M := \{\theta \in U : \mathcal{B}(\theta) = \mathcal{B}(\theta_0)\}$$

is a C^1 submanifold through θ_0 of dimension $p - q$. Its tangent space at θ_0 is $\ker J_{\mathcal{B}}(\theta_0)$. The condition

$$d\alpha_{\theta_0} \notin \text{rowspan}(J_{\mathcal{B}}(\theta_0))$$

is equivalent to the existence of some vector

$$v \in \ker J_{\mathcal{B}}(\theta_0)$$

with

$$d\alpha_{\theta_0}[v] \neq 0,$$

because the annihilator of $\ker J_{\mathcal{B}}(\theta_0)$ is exactly the row span of $J_{\mathcal{B}}(\theta_0)$. Choose such a v. Since M is a submanifold with tangent space $\ker J_{\mathcal{B}}(\theta_0)$, there exists a C^1 curve $c : (-\varepsilon, \varepsilon) \to M$ with $c(0) = \theta_0$ and $c'(0) = v$. Along this curve, $\mathcal{B}(c(t)) = \mathcal{B}(\theta_0)$ for all small t. Meanwhile,

$$\alpha(c(t)) = \alpha(\theta_0) + t\, d\alpha_{\theta_0}[v] + o(t).$$

Because $d\alpha_{\theta_0}[v] \neq 0$, the values of $\alpha(c(t))$ are strictly above and below $\alpha(\theta_0)$ for sufficiently small positive and negative t. Setting $\theta^+ = c(t)$ and $\theta^- = c(-t)$ for small enough t proves the claim. □

The theorem is deliberately exact rather than rhetorical. It does not say that loss or benchmarks are uninformative. It says something narrower and stronger: if the lifecycle scalar really does vary along an output level set, then output-only observables cannot identify it even in principle. The theorem fails precisely when the atlas-sensitive direction lies in the span of the chosen output-only surfaces. That condition is falsifiable.

Concrete lifecycle margins for the theorem. The theorem itself is generic. For the lifecycle claims in this paper we instantiate it with smooth surrogates of the exact receipts reported later. The empirical sections use the exact receipts because those are what operators actually see. The theorem uses smooth surrogates because the observability argument is differential.

For a fixed probe family $\mathcal{P}$, protected layer set $\mathcal{L}_{\text{protect}}$, and inverse-temperature $\beta > 0$, write

$$\text{smin}_\beta(a_1, \ldots, a_n) := -\beta^{-1} \log \sum_{i=1}^{n} e^{-\beta a_i}, \tag{18}$$

$$\text{smax}_\beta(a_1, \ldots, a_n) := \beta^{-1} \log \sum_{i=1}^{n} e^{\beta a_i}. \tag{19}$$

For probe token t, layer ℓ, and checkpoint θ, let $\rho_{\theta,\ell}(e \mid t)$ denote the full routed-expert gate probability and $s_{\theta,\ell}(e \mid t)$ the corresponding pre-top-k gate score at the checkpoint-native gate coordinate. Define the stabilized field compatibility

$$\kappa_{\ell,e,e'}(t;\theta) := \frac{\langle v^{(\theta)}_{e,\ell}(t),\, v^{(\theta)}_{e',\ell}(t)\rangle}{\|v^{(\theta)}_{e,\ell}(t)\|_2 \, \|v^{(\theta)}_{e',\ell}(t)\|_2 + \varepsilon_\kappa}, \tag{20}$$

and the smooth adjacency weights

$$\omega^{\text{adj}}_{\ell,e,e'}(t;\theta) := \frac{\rho_{\theta,\ell}(e \mid t)\, \rho_{\theta,\ell}(e' \mid t)\, \exp\!\big(-\beta_{\text{adj}} |s_{\theta,\ell}(e \mid t) - s_{\theta,\ell}(e' \mid t)|\big)}{\sum_{i \neq j} \rho_{\theta,\ell}(i \mid t)\, \rho_{\theta,\ell}(j \mid t)\, \exp\!\big(-\beta_{\text{adj}} |s_{\theta,\ell}(i \mid t) - s_{\theta,\ell}(j \mid t)|\big)}. \tag{21}$$

These emphasize expert pairs that are simultaneously likely and near-competitive. The corresponding smooth overlap-health surrogate is

$$\widetilde{C2}_\ell(\theta) := \mathbb{E}_{t\sim\mathcal{P}} \left[\sum_{e \neq e'} \omega^{\text{adj}}_{\ell,e,e'}(t;\theta)\, \kappa_{\ell,e,e'}(t;\theta) \right]. \tag{22}$$

For occupancy, define the smooth expert marginal

$$\widetilde{\pi}_{\mathcal{P},\ell}(e;\theta) := \mathbb{E}_{t\sim\mathcal{P}}\big[\rho_{\theta,\ell}(e \mid t)\big], \tag{23}$$

and the soft lower-tail occupancy

$$\widetilde{\pi}_{\text{tail},\ell}(\theta) := \text{smin}_{\beta_\pi,\, e}\, \widetilde{\pi}_{\mathcal{P},\ell}(e;\theta). \tag{24}$$

We calibrate handoff against a healthy pretrained reference family by fixing reference values $\widetilde{C2}^{\text{ref}}_\ell$ and $\widetilde{\pi}^{\text{ref}}_{\text{tail},\ell}$. The layerwise handoff score is

$$m^{\text{handoff}}_\ell(\theta) := \lambda_1\big(\widetilde{C2}_\ell(\theta) - \widetilde{C2}^{\text{ref}}_\ell\big) \;+\; \lambda_2 \left(\frac{\widetilde{\pi}_{\text{tail},\ell}(\theta)}{\widetilde{\pi}^{\text{ref}}_{\text{tail},\ell}} - 1 \right), \tag{25}$$

and the smooth handoff margin is

$$h(\theta) := \text{smin}_{\beta_h,\, \ell\in\mathcal{L}_{\text{protect}}}\, m^{\text{handoff}}_\ell(\theta). \tag{26}$$

Positive h means that every protected layer clears the calibrated overlap-and-occupancy floor up to the soft minimum; $h = 0$ is the handoff boundary.

For post-training relative to a base checkpoint $\bar{\theta}$, define the smooth boundary surrogate

$$\widetilde{\Delta}_{\text{boundary},\ell}(\theta, \bar{\theta}) := \mathbb{E}_{t\sim\mathcal{P}}\left[1 - \sum_{e} \rho_{\bar{\theta},\ell}(e \mid t)\, \rho_{\theta,\ell}(e \mid t)\right], \tag{27}$$

and the smooth content surrogate

$$\widetilde{\Delta}_{\text{content},\ell}(\theta, \bar{\theta}) := \mathbb{E}_{t\sim\mathcal{P}}\left[\sum_{e} \bar{\rho}_{\ell,e}(t)\, \frac{\|v_{e,\ell}^{(\theta)}(t) - v_{e,\ell}^{(\bar{\theta})}(t)\|_2}{\|v_{e,\ell}^{(\theta)}(t)\|_2 + \|v_{e,\ell}^{(\bar{\theta})}(t)\|_2 + \varepsilon_u}\right], \tag{28}$$

where $\bar{\rho}_{\ell,e}(t) := \frac{1}{2}(\rho_{\theta,\ell}(e \mid t) + \rho_{\bar{\theta},\ell}(e \mid t))$. The layerwise repurposing load is

$$m_{\ell}^{\text{refine}}(\theta; \bar{\theta}) := \mu_1\, \widetilde{\Delta}_{\text{boundary},\ell}(\theta, \bar{\theta}) \;+\; \mu_2\, \widetilde{\Delta}_{\text{content},\ell}(\theta, \bar{\theta}) \;+\; \mu_3 \left(1 - \frac{\widetilde{\pi}_{\text{tail},\ell}(\theta)}{\widetilde{\pi}_{\text{tail},\ell}(\bar{\theta})}\right), \tag{29}$$

and the smooth refinement margin is

$$r(\theta; \bar{\theta}) := \tau - \text{smax}_{\beta_r,\, \ell\in\mathcal{L}_{\text{protect}}}\, m_{\ell}^{\text{refine}}(\theta; \bar{\theta}). \tag{30}$$

Positive r means the tuned checkpoint remains within the protected-layer preservation budget; $r = 0$ is the refinement-versus-repurposing boundary.

These surrogates are C^1 whenever the gate-score maps, routed field maps, and probe-family expectations are C^1 in the checkpoint weights. The exact empirical receipts later in the paper use discrete top-k sets, exact occupancy minima, and exact drift summaries. The surrogates above are introduced only to make the theorem speak directly about the same lifecycle questions without pretending that exact routing indicators are differentiable. The corresponding exact empirical handoff and refinement margins are

$$h_{\text{ex}}(\theta) := \min_{\ell\in\mathcal{L}_{\text{protect}}}\left[\lambda_1\big(C2_{\ell}(\theta) - C2_{\ell}^{\text{ref}}\big) \;+\; \lambda_2\left(\frac{\pi_{\text{min},\ell}(\theta)}{\pi_{\text{min},\ell}^{\text{ref}}} - 1\right)\right], \tag{31}$$

$$r_{\text{ex}}(\theta; \bar{\theta}) := \tau - \max_{\ell\in\mathcal{L}_{\text{protect}}}\left[\mu_1\, \Delta_{\text{boundary},\ell}(\theta, \bar{\theta}) \;+\; \mu_2\, \Delta_{\text{content},\ell}(\theta, \bar{\theta}) \;+\; \mu_3\left(1 - \frac{\pi_{\text{min},\ell}(\theta)}{\pi_{\text{min},\ell}(\bar{\theta})}\right)\right], \tag{32}$$

where $C2_{\ell}$, $\pi_{\text{min},\ell}$, $\Delta_{\text{boundary},\ell}$, and $\Delta_{\text{content},\ell}$ are the exact empirical receipts reported later in the paper.

Proposition 3 (Surrogate-to-receipt sign consistency away from ties)**.** *Fix a compact checkpoint neighborhood K on which every protected layer avoids routing ties and occupancy ties, and assume the corresponding score and occupancy gaps are bounded below by positive constants on K. Then as $(\beta_{\text{adj}}, \beta_{\pi}, \beta_h, \beta_r) \to \infty$, the smooth margins h and r converge uniformly on K to h_{ex} and r_{ex} respectively. Consequently, if $|h_{\text{ex}}| \geq \delta_h$ and $|r_{\text{ex}}| \geq \delta_r$ on K for some $\delta_h, \delta_r > 0$, then for sufficiently large inverse temperatures the smooth and exact margins have the same signs on K.*

Proof. Away from routing ties and occupancy ties, each hard minimum, hard maximum, and hard competitor-selection rule entering h_{ex} and r_{ex} is isolated by a positive gap on K. The log-sum-exp approximations smin_{β} and smax_{β} therefore converge uniformly on K to their hard counterparts as

the corresponding inverse temperatures grow. The same gap condition gives uniform convergence of the smooth adjacency weights to the hard near-competitive selection rule and of the soft lower-tail occupancy to the exact minimum occupancy. Composing these convergences with the finite layerwise min/max constructions defining the lifecycle margins yields uniform convergence of (h, r) to $(h_{\text{ex}}, r_{\text{ex}})$ on K. Once the uniform approximation error is below $\min(\delta_h, \delta_r)$, the smooth and exact margins must agree in sign on K. □

Corollary 1 (Handoff ambiguity). *Let h be the smooth handoff margin defined above. If $h(\theta_0) = 0$ and $dh_{\theta_0} \notin \text{rowspan}(J_{\mathcal{B}}(\theta_0))$, then arbitrarily near θ_0 there exist checkpoints with identical output-only observables but opposite handoff status. Loss-equivalent checkpoints can therefore sit on opposite sides of post-training readiness near the handoff boundary.*

Corollary 2 (Refinement-versus-repurposing ambiguity). *Fix a base checkpoint $\bar{\theta}$ and let $r(\theta; \bar{\theta})$ be the smooth refinement margin defined above. If $r(\theta_0; \bar{\theta}) = 0$ and $d_\theta r(\theta_0; \bar{\theta}) \notin \text{rowspan}(J_{\mathcal{B}}(\theta_0))$, then arbitrarily near θ_0 there exist checkpoints with identical output-only observables but opposite refinement status. Reward or benchmark improvement alone therefore cannot certify capability preservation.*

Proposition 4 (Operational necessity of atlas receipts for lifecycle classification). *Assume the setting of Theorem 2 and suppose a local control policy must choose different actions on the regions $\{\alpha > 0\}$ and $\{\alpha < 0\}$ (for example, promote vs. hold, or continue vs. intervene). Then any local policy that factors only through $\mathcal{B}$ is unsound near θ_0.*

Proof. Any control policy that factors through $\mathcal{B}$ assigns the same action to all checkpoints on the level set M. But Theorem 2 gives nearby checkpoints on that same level set with different lifecycle scalar values, and Corollaries 1 and 2 show that the relevant stage label can change sign there. So the policy must misclassify at least one nearby checkpoint. □

This is the mathematical center of the paper. The handoff corollary is the paper's primary instantiation of the theorem. The refinement-versus-repurposing corollary is the parallel extension that matters just as much strategically, but does not yet carry equal empirical closure in this draft. The experiments below do not prove the lifecycle claim. They validate that real large-MoE training enters the ambiguous regime described by Theorem 2, and that the exact empirical receipts track the same sign structure as the smooth surrogate margins in the separated regime covered by Proposition 3.

What would falsify the two headline claims. The theorem is strong enough to be worth trying to kill. For the handoff claim, there are two ways to do it. Mathematically: if the exact handoff margin h_{ex} is locally measurable as a function of the chosen output-only observable map $\mathcal{B}$ on the relevant neighborhood, then Theorem 2 does not force ambiguity for that handoff criterion. Operationally: if matched-scalar checkpoints with opposite signs of h_{ex} promote indistinguishably under the same post-training recipe, then the handoff boundary is not operationally meaningful. The refinement claim has the same two falsifiers. Mathematically: if the exact refinement margin r_{ex} is locally measurable from the chosen output-only surfaces, then Corollary 2 does not force ambiguity for that criterion. Operationally: if checkpoints with opposite signs of r_{ex} show no meaningful difference in continued behavior or atlas preservation, then the refinement-versus-repurposing boundary is not operationally meaningful. These are the paper's actual falsifiers. They are stronger than the vague alternative that "geometry was not useful."

Table 1 Model architecture shared by both experimental runs.

Parameter	Value
Model dimension d	2048
Expert intermediate dimension h	5760
Routed layers	27
Routed experts E	640
Top-k	6
Shared experts	1 per layer
Attention	Multi-headed Latent Attention (MLA)
Tokenizer	`o200k_harmony` (vocab 201,088)

Table 2 DSV3-family V2 summary from earlier probe studies on the same model lineage. C1 is ΔBI (boundary vs interior) and C2 is $\Delta Brand$ (boundary vs random-adjacent). More negative indicates a stronger pass on that axis. The family pattern is stable: C1 is learned strongly in the base model, while C2 is weaker and improves only selectively in later descendants.

Checkpoint	Domain	Mean C1 (ΔBI)	Mean C2 ($\Delta Brand$)
Base	fineweb	−0.002437	−0.000180
Base	code	−0.001786	−0.000566
0324	fineweb	−0.002089	−0.000449
0324	code	−0.001797	−0.000863
R1	fineweb	−0.001976	−0.000697
R1	code	−0.001635	−0.000722

7 Experiments

We evaluate the atlas framework on two large-MoE training runs at Noumena. Both use the same DeepSeek-V3-shaped architecture (Table 1). We do this deliberately. Routing geometry is family-dependent, and mixing substantially different MoE families into one empirical section would make the monitoring question easier to state and harder to answer. The point of this section is to show how the framework distinguishes several sparse-MoE failure patterns. It is to show, on one concrete and demanding family, which questions geometric receipts answer that scalar loss and aggregate routing telemetry do not. The mathematical crux has already been established in Section 6. What remains is to validate that frontier runs on a real large-MoE family actually realize that ambiguity in practice, and that the atlas receipts separate the resulting claims the way the theorem predicts.

7.1 DSV3 family prior: C1 and C2 are different empirical objects

Before turning to the two live training runs, we need one family-level fact on the same architectural lineage. Throughout this section, "Base", "0324", and "R1" refer to three released checkpoints on the same DeepSeek-V3 family line: the original base checkpoint, a later general descendant, and a later reasoning-oriented descendant. On that lineage, the two V2 axes do not move together. Boundary-vs-interior redundancy (C1) and adjacency specificity (C2) separate across checkpoints, layers, and domains. That observation is central. It is the empirical reason this paper refuses to collapse co-active overlap structure into a slightly fancier version of boundary sharpness.

The aggregate pattern in Table 2 is already enough to make the point. Across the measured DSV3 slices, C1 passes all 18 layer-level tests, while C2 passes only 10 of 18. By checkpoint, the C2 pass count is 1/6 at base, 4/6 at 0324, and 5/6 at R1. If C1 and C2 were merely two noisy views

of one quantity, the family would not look like this. It does. The family learns one axis strongly and the other only partially. The layer-level receipts show the split more sharply still. On base fineweb, layers 24, 32, and 40 all pass C1 while all three fail C2; layer 40 is the clearest example, with $\Delta BI = -0.002997$ but $\Delta Brand = +0.000107$. On 0324 code, layer 24 passes both; on R1 code, layers 24, 32, and 40 all pass both. The family itself teaches the lesson: C1 and C2 measure different quantities. They are two different questions about the atlas. One asks whether boundary neighborhoods are more redundant than the interior. The other asks whether the *adjacent* co-active expert is actually more compatible than a random alternative.

7.2 Three empirical facts about the atlas

The experiments in this section report three facts. Each answers a different question about the atlas, and together they show that the framework distinguishes preserved, stressed, and recovered regimes rather than merely diagnosing failure.

The first fact is that an atlas can be preserved through a stage transition. A run that completed frozen-expert distillation with clean geometry and was then promoted to supervised fine-tuning maintained healthy overlap compatibility throughout. After roughly 20B tokens of SFT on top of the distillation baseline, the atlas read is still GREEN. This is the positive case: chart-preserving adaptation, as defined in Section 4, actually works when occupancy stays broad and co-active overlap remains compatible.

The second fact is that an otherwise successful run can later leave a healthy geometric regime while scalar loss continues to improve. A different frozen-expert distillation configuration enters a stressed post-shift window in which overlap compatibility slides and occupancy remains degraded. This is the pattern the framework is designed to catch: scalar improvement without geometric recovery.

The third fact is that geometric damage can be temporary. On an earlier configuration of the same model family, overlap compatibility degraded during early training, recovered around the midpoint of the run, and remained healthy for an extended period before eventually deteriorating again late in training. The recovery was lost because no checkpoint was saved at the healthy state. This is both encouraging (the atlas can repair itself given enough training) and cautionary (the repair window can close, and if you miss it, you may not get it back).

All three facts come from the same 640-expert DeepSeek-V3-shaped architecture (Table 1). All geometry evaluations use fixed deterministic canary windows against a pre-declared baseline. The paper is reporting the receipts the monitoring stack would have emitted during training rather than whatever slice of evidence happened to look interesting afterward.

7.3 Fact 2: scalar improvement without geometric recovery

Setup. The second run is a frozen-expert knowledge distillation configuration on the same architecture. The expert weights, shared experts, router gate weights, and FFN normalization parameters are frozen from a 640-expert DeepSeek-V3-shaped donor checkpoint. The trainable surface consists of: attention projections (Q/K/V/O), token embeddings, LM head, the MTP-4 module (multi-token prediction with 4 lookahead heads [5]), Canon A/B/C convolutions [10], and mHC mixing matrices [11], totaling approximately 12B trainable parameters out of approximately 691B total.

Training uses Adafactor [12] with peak learning rate 1.0×10^{-4} and cosine decay. Batch size is 8 sequences of length 8192 (≈65K tokens/step). The teacher signal is sparse logit artifacts (KD-next

Table 3 Distill-640 post-shift progression: scalar loss improves while atlas health deteriorates. State2 C2 measures co-active overlap compatibility (higher is better; negative means degradation below baseline). All geometry gates return YELLOW. Loss alone supports the narrow claim that the distillation objective improved on the currently visited region.

Step	Loss	State2 C2	State1 margin	Dead experts	Mean CV	Min entropy
Baseline	—	+0.083	(seed)	42	140	3.00
500	4.73	−0.027	0.009	285	190	2.32
1000	4.53	−0.020	0.011	285	190	2.32
1500	4.48	−0.042	0.013	300	194	2.25
2000	4.48	−0.063	0.014	300	205	2.25

+ MTP-4 cascade, $\alpha_{\text{mtp}} = 0.3$). Router bias update is disabled and aux loss weight is zero, since both router and expert weights are frozen. The run executes on 8 NVIDIA B200 GPUs (192 GB HBM each), single node. Geometry is evaluated every 500 steps on a fixed canary probe family against the seed baseline. The full run succeeded. The slice reported below comes from a later phase after a data-regime change, where the scalar objective continued to improve while the geometry and occupancy receipts left the earlier healthy regime.

Results. Table 3 shows the central result. Within this post-shift window, distillation loss falls from 5.25 to 4.48 over $\sim$130M tokens ($\sim$2000 optimization steps at 65K tokens/step). Standard monitoring would classify this phase as successful. That reading is incomplete. If a monitoring stack cannot distinguish scalar success from atlas preservation, it cannot do the job we actually need from it.

Co-active overlap compatibility (State2 C2) drops from $+0.083$ to -0.027 at step 500. A partial recovery occurs between steps 500 and 1000 (C2 rises to -0.020), but from step 1000 onward the trend worsens: $-0.020 \to -0.042 \to -0.063$. The total swing from baseline to step 2000 is 0.146 in overlap compatibility. The brief improvement at step 1000 matters because it rules out the easiest dismissal. This pattern persists across multiple gates. The post-shift phase had a chance to settle and then drifted further away from the earlier healthy regime anyway.

Routing occupancy remains materially degraded at every checkpoint. Dead experts hover between 285 and 300 (baseline: 42). CV ranges from 190 to 205 (baseline: 140). Minimum entropy is 2.25 at step 2000 (baseline: 3.00). Of 640 total experts, only approximately 55 receive meaningful traffic, less than 9% of expert capacity. this phase of the run is claiming scalar progress while behaving like a much smaller and much worse-routed model.

The State1 margin trap. State1 (mean boundary margin) shows a superficially positive trend: 0.009 at step 500 to 0.014 at step 2000. We initially found this reassuring. We were wrong. With 285+ dead experts, the surviving routing competitions involve a smaller, more concentrated set of experts. Wider margins here mean the model routes more decisively to fewer experts while losing breadth across the full atlas. Rising margins are non-diagnostic without stable occupancy. We mention this because it is the kind of mistake we expect others to make too.

Interpretation. This is the scalar-geometric decoupling the atlas framework predicts. The trainable surface optimizes effectively: loss falls, next-token predictions improve on the visited data. But the frozen expert atlas is accessed through a narrowing occupancy window. The model concentrates onto

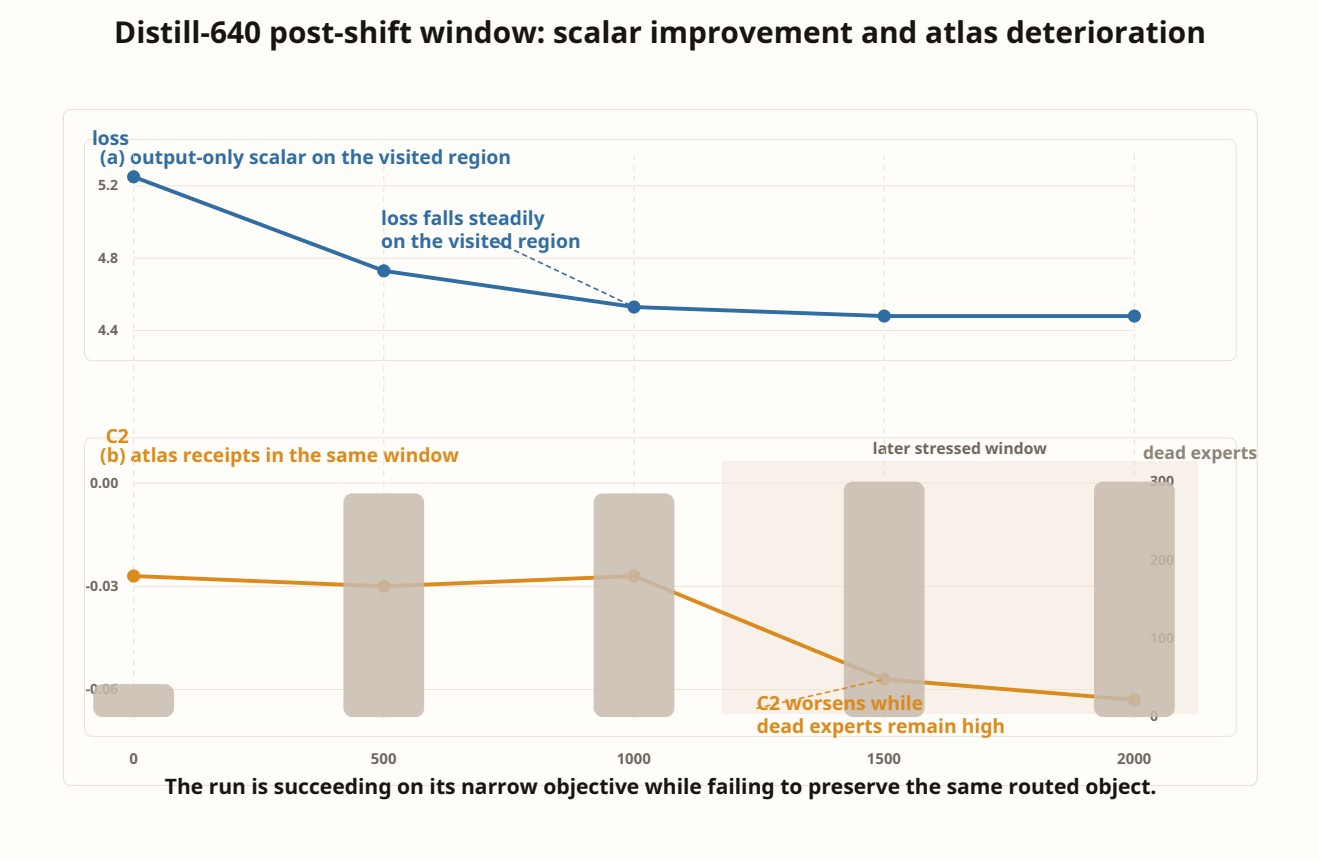

Figure 4 Scalar improvement without geometric recovery. **Top:** distillation loss falls steadily, which supports the narrow claim that the current objective is improving on the visited region. **Bottom:** co-active overlap compatibility (C2, amber) worsens from step 1000 onward while dead experts (gray bars) remain 7× above baseline. The decoupling between objective progress and structural health is the central empirical result of this paper.

a subset of experts and achieves scalar improvement on that subset, while the broader 640-expert knowledge geometry degrades in overlap compatibility.

Loss is answering a smaller question truthfully. Standard routing statistics detect the occupancy problem but cannot distinguish it from a structural overlap failure. The atlas metrics provide the missing decomposition: the primary failure is co-active overlap compatibility (State2) rather than boundary ambiguity (State1), and both are accompanied by severe occupancy degeneration. The uncomfortable lesson is simple. The run is still succeeding at its immediate distillation objective. It is also failing to preserve the same routed object in this phase.

Why freezing the experts makes the result stronger. One might think freezing the experts makes the atlas safe by construction. It does not. One can always produce geometric damage by moving the experts themselves. That case is less informative. The interesting case is the one everyone wants to believe is safe: freeze the experts, freeze the router, train only the light surrounding surfaces. The whole premise is that the atlas should be read-only. It isn't. Distill-640 shows that the frozen atlas can be functionally rewritten just by changing which angular regions the backbone visits. You can damage the atlas without touching a single expert weight. We do not have a good explanation for why this damage is as large as it is. We suspect the attention-and-backbone changes redirect

token paths enough that the effective co-active neighborhoods shift substantially, but we have not isolated the mechanism. We report the result we have rather than the one we wish we had.

7.4 Fact 1: preservation through a stage transition

Setup. The first run trained through frozen-expert distillation on the same 640-expert architecture, then was promoted to supervised fine-tuning (CE + MTP-4 CE, $\alpha_{\text{mtp}} = 0.3$) once the distillation geometry was clean. The SFT phase runs on 64 B200 GPUs across 16 nodes on a GB300 NVL72 cluster at a learning rate of 2.75×10^{-5} with Adafactor.

Results. After roughly 20B tokens of SFT on top of the distillation baseline, the atlas read is GREEN: State2 C2 = +0.0086, State1 margin = 0.0034, State3 NLL = 4.46. Routing occupancy is healthy throughout: zero dead experts, all 640 active, entropy above 6.0 across all layers. The run went through a cluster incident (stale GPU taints halted training), was restarted from checkpoint, and continued training. Even after the restart, the post-baseline geometry gates remain GREEN.

This is the positive case. The atlas survived both the objective change (distillation to SFT) and the operational interruption. C2 is positive and stable. Occupancy is broad. The stage transition did not rewrite the atlas.

Interpretation. Chart-preserving adaptation, as defined in Section 4, actually works here. The SFT objective refines policy on top of the frozen expert atlas without repurposing the mid-band interior. The geometry metrics confirm this: co-active overlap compatibility is preserved, boundary margins are stable, and occupancy remains broad.

Two things matter about this result. First, the distillation phase was clean before promotion. We did not promote a geometrically unhealthy checkpoint to SFT. The preservation story starts from a healthy baseline, which is the precondition the framework requires. Second, the cluster incident provides a natural test of stability. The atlas survived the objective change as well as the interruption and restart. That is stronger evidence than a single uninterrupted run would provide.

We would not claim this proves that all stage transitions preserve the atlas. We claim it shows that preservation is possible, that the metrics detect it, and that the framework distinguishes this outcome from the failure mode in the next subsection.

7.5 Fact 3: geometric recovery is possible but temporary

On an earlier configuration of the same model family, overlap compatibility degraded during early frozen-expert training, recovered around the midpoint, and remained healthy for an extended period before deteriorating again late in training. The recovery was real: C2 returned to baseline-compatible levels and the atlas functioned normally for billions of tokens. The subsequent deterioration was also real.

This means atlas damage during frozen-expert training can recover. The co-active overlap structure can repair itself as the trainable backbone learns to query the frozen experts more compatibly. But recovery windows close. A geometry-aware checkpoint policy should save when the atlas is healthy, because the next gate may not be. A geometry-aware checkpoint policy would have preserved the recovery point.

Table 4 Claim-licensing semantics of the receipts used in this paper. Each surface certifies some claims and is silent on others.

Receipt	Object measured	What it can license	What it cannot license
Loss / State3	Objective quality on the visited region	"The objective improved" or "canary NLL improved."	Atlas preservation, overlap compatibility, or whether progress came from a narrowed visited region.
Occupancy telemetry	Breadth of atlas usage	"The run is using a broad / narrow subset of experts."	Whether co-active experts are compatible or whether a healthy resume preserved the atlas.
State1	Boundary-local decisiveness	"Top-k competitions are sharper / noisier near the boundary."	Global routed health; under occupancy collapse it cannot certify healthier routing.
State2	Co-active overlap compatibility	"Adjacent co-active charts are becoming more / less compatible."	Whether the atlas is broadly used; C2 alone cannot separate local compatibility from global occupancy collapse.
Checkpoint-pair drift	Movement of the routed object between checkpoints	"Coordinates / boundaries / transitions / content moved by this much."	Whether that motion is acceptable for the current stage without a stage-specific claim.
Resume baseline	New post-interruption reference point	"This is the state of the atlas at resume time."	Any preservation claim about what happened *after* resume.

7.6 What claims the receipts can and cannot support

We should say explicitly what the paper is *not* claiming. The paper does not rank monitors by a single scalar. Loss, occupancy, State1, State2, and checkpoint-pair drift answer different questions about different objects. The operational mistake is to ask one receipt to certify a claim that belongs to another object.

Read this table against the two runs. On Distill-640, loss supports only the narrow claim that the distillation objective improved on the visited region. Occupancy licenses the claim that atlas usage collapsed. State2 licenses the claim that co-active overlap compatibility worsened. Together, those receipts license the stronger structural claim: scalar improvement occurred without geometric recovery. On SFT-640, loss and occupancy license the claim that operational recovery succeeded. The resumed baseline licenses only the claim that a new reference point exists. It does *not* license any preservation claim yet.

7.7 Which lifecycle question becomes answerable during the run?

The online monitoring problem is therefore not to decide which surface wins. It is to determine which question becomes answerable at which gate, and with which receipts. That is the framing that matters if the goal is to decide whether to continue, promote, or stop a run without a stop-the-world evaluation session.

This is the more faithful way to summarize what the experiments contribute. The receipts do not compete for a single crown. They compose into answers to different lifecycle questions. That is

Table 5 Which lifecycle question becomes answerable during the run. Different questions become answerable at different times because they depend on different receipts.

Question	Distill-640: first informative gate	SFT-640 after resume	Receipt combination required
Is the current objective improving on the visited region?	Immediately; loss improves from the first measured gate onward.	Immediately; live loss is stable and healthy.	Loss / State3.
Is atlas usage broadening or narrowing?	Step 500; occupancy collapses immediately and stays bad.	Immediately; occupancy is healthy after resume.	Occupancy telemetry.
Are co-active chart overlaps becoming more or less compatible?	Step 500; C2 turns negative immediately and worsens after the early blip.	Not answerable yet post-resume; only a new baseline exists.	State2 plus a valid baseline.
Can we claim preservation after the interruption?	Not the relevant question on this run.	Not answerable at resume time; it requires the first post-baseline comparison.	Resume-baseline semantics plus a later checkpoint-pair comparison.
Can we claim scalar improvement without geometric recovery?	Step 500 provisionally, and decisively by steps 1500–2000.	Not the relevant question on this run.	Loss / State3 + State2 + occupancy, read jointly.

exactly why the geometry lens matters. Without an explicit routed object, it is too easy to slide from "the objective improved" to "the model is healthier" or from "the resume looks operationally fine" to "the atlas was preserved."

7.8 What the operator should have decided at each gate

The next question is more operational. Suppose these receipts were arriving live during training rather than being read retrospectively for a paper. What should the operator have done at each Distill-640 gate? Table 6 answers that question directly. If the paper is useful, the answer should not be "wait for the appendix." It should be legible from the receipts themselves.

This table is worth being explicit about because it captures a common failure in large-scale training operations. Teams often treat "keep training" as the neutral option. On Distill-640 it becomes an active choice. By step 2000 it is an active decision to tolerate a known structural failure in exchange for a scalar win that the monitoring stack no longer knows how to interpret safely.

7.9 Atlas drift vs. parameter drift

The trainable parameters in Distill-640 represent ≈12B out of 691B total. The frozen expert weights do not change. Raw parameter distance between consecutive checkpoints is small relative to total model size.

Yet the atlas undergoes substantial restructuring: C2 moves from $+0.083$ to -0.063, a swing of 0.146 driven entirely by changes in how the frozen atlas is *queried* (attention patterns, embedding positions, MTP lookahead fields). The router weights are frozen, so this is backbone coordinate drift that changes which regions of the frozen atlas are visited and how co-active experts are combined.

This demonstrates a failure mode parameter distance cannot detect: the atlas can be restructured

Table 6 Gate-by-gate operator policy on Distill-640. The key point is that the first bad gate revokes the right to call the run healthy, but does not yet force a hard intervention. The subsequent gates decide whether the run is recovering or merely buying time.

Step	What is licensed already	What remains unlicensed or contradicted	Correct operator action
500	The objective improved on the visited region; occupancy has already narrowed sharply.	Any claim of structural health is contradicted immediately: C2 is negative at the first gate.	Revoke any success claim. Keep the next gate on schedule and stop treating the run as healthy by default.
1000	The objective improved further; the brief C2 rebound shows the run can move in the right direction.	Recovery is still unlicensed because occupancy does not improve at all.	Do not clear the run. Require joint recovery rather than a single encouraging blip.
1500	Scalar quality remains good.	The structural claim worsens again: overlap compatibility deteriorates and occupancy stays degraded.	Prepare intervention. Passive continuation is now on thin ice.
2000	A decent scalar outcome is still licensed.	The stronger structural claim is now clear: worst C2 so far, fourth YELLOW gate, occupancy still badly concentrated.	Intervene or stop claiming progress. At this point "wait and see" is no longer a neutral choice.

Table 7 Why parameter drift is the wrong primary monitoring surface for Distill-640. The trainable update is small relative to total parameter count and leaves experts frozen, yet the atlas read is decisively negative.

Monitoring surface	What it says on Distill-640
Raw parameter distance	Reassuring. Only $\sim$12B of 691B parameters are trainable and all expert weights are frozen.
Scalar loss	Reassuring. Distillation objective improves from 5.25 to 4.48.
Output proxy (State 3)	Reassuring. Canary-window NLL improves monotonically.
Occupancy telemetry	Worrying but ambiguous. Routing is badly concentrated, but this alone does not tell us whether the failure is structural or merely operational.
Atlas read (State 2 + occupancy)	Negative. Co-active overlap compatibility worsens from -0.027 to -0.063 after the early blip while occupancy remains severely degraded.

without moving expert weights at all, simply by changing the coordinate system through which it is accessed.

7.10 Per-layer geometry breakdown

The aggregate C2 metric in Table 3 averages across all 27 routed layers. A natural question is whether the degradation is uniform or concentrated in specific layers. Table 8 shows the per-layer C2 at step 2000 for a representative subset of layers, grouped by position in the model.

The mid-band layers (8–20) show the worst C2 degradation: mean -0.078 with a worst-layer value of -0.112. Early layers (1–7) are less degraded at -0.031, consistent with these layers performing interface-level tokenization and positional processing that is less dependent on expert specialization. Late layers (21–27) are intermediate at -0.044, which may reflect their role in terminal expression and response shaping.

Table 8 Per-layer State2 C2 at step 2000 (Distill-640). Early and late layers show different degradation patterns. Mid-band layers (8–20) show the most severe overlap-compatibility loss, consistent with the hypothesis that these layers carry the dominant knowledge-bearing atlas structure.

Layer group	Layers	Mean C2	Worst layer C2	Dead experts (mean)
Early (1–7)	7	−0.031	−0.048	312
Mid-band (8–20)	13	−0.078	−0.112	295
Late (21–27)	7	−0.044	−0.067	288
All layers	27	−0.063	−0.112	300

Figure 5 makes this pattern visual: the mid-band bars are substantially longer (worse C2) than early or late layers.

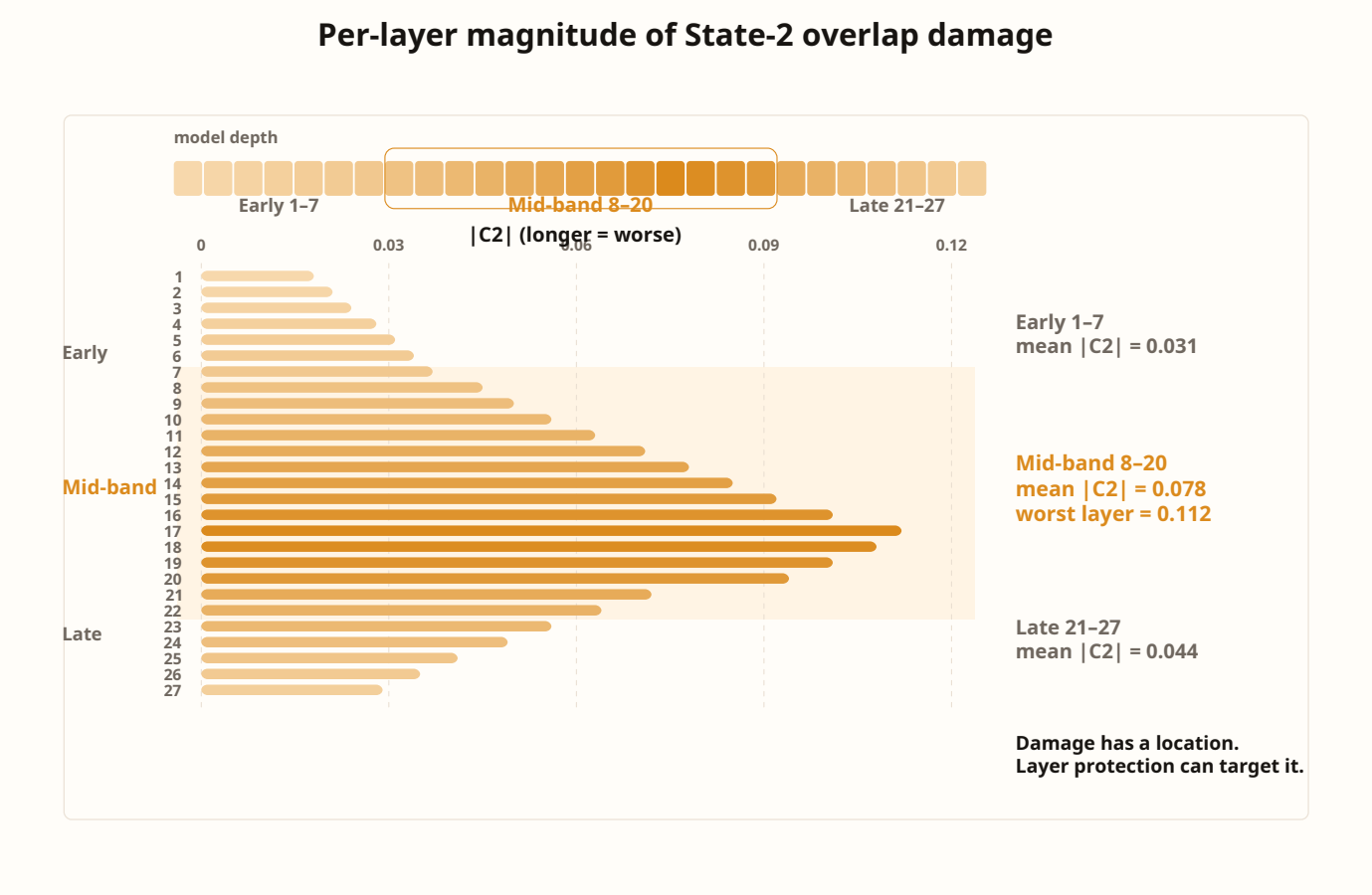

Figure 5 Per-layer overlap degradation in the post-shift window. Bar length is proportional to |C2| (longer = worse). Color intensity tracks severity from cream (baseline) to orange (worst). Mid-band layers 8–20 carry the dominant damage. Early and late layers are substantially less affected. This pattern is relevant for the design question of which layers to protect during adaptation.

The mid-band layers with the worst C2 degradation are the layers whose atlas structure is most knowledge-bearing, and therefore the layers that chart-preserving adaptation should prioritize protecting.

Table 9 SFT-640: operational state at step 36890 (post-recovery window 36780–36890). All routing occupancy metrics are healthy. Geometry baseline is GREEN and serves only as a resumed-checkpoint baseline; preservation remains untested.

Metric	Value	Baseline (step 36500)
Loss (mean)	6.50	—
Grad norm (mean)	117.7	—
Dead experts	0	0
Experts active	640	640
Min entropy	6.07	6.14
Mean CV	57.7	47.6
State1 margin	—	0.0035
State2 C2	—	+0.0057
State3 NLL	—	4.8241

8 Design Guide: Atlas Monitoring in Practice

The atlas framework has direct operational consequences. It implies a concrete monitoring protocol and a set of design decisions for large-MoE training. This section distills the framework into practical guidance, organized around the questions a training operator actually faces: when a run is ready to hand off from pretraining to post-training, whether post-training is refining or repurposing the atlas, how close the run is to forgetting, and whether those judgments can be made online rather than after the fact. This is a control problem. The goal is to make those four decisions legible with receipts that are cheap enough to collect during a live run and sharp enough to keep different claims from bleeding together.

8.1 When to compute atlas metrics

Atlas-semantic drift functionals (Section 5) should be computed at three points in every training run:

Step 0 baseline. Before any training step, evaluate all four drift functionals and all occupancy statistics on the fixed probe family. This establishes the reference atlas state. Without this baseline, no subsequent measurement is interpretable. The baseline should be stored alongside the checkpoint as a companion artifact.

Periodic gates. Evaluate every N steps, where N is chosen so that the evaluation cost is a small fraction of training compute. In our experiments, geometry gates run at the same cadence as checkpoint saves. The evaluation uses a fixed set of deterministic canary windows in a single forward pass. On both runs, the wall-clock overhead is below 3% of training time.

Post-resume baselines. Any time training is interrupted and resumed (whether by a cluster incident, a hyperparameter change, a data source switch, or a learning rate restart), the first post-resume evaluation is a *new baseline*; preservation remains untested until the next comparison gate. The SFT-640 experiment (Section 7.4) illustrates why this matters: a healthy-looking post-resume checkpoint tells you nothing about whether the atlas will be preserved going forward.

Table 10 The four lifecycle questions that motivate the atlas framework. Each question requires a different receipt set because each asks about a different object. Theorem 2 explains why output-only surfaces alone leave these questions underdetermined.

Lifecycle question	Minimum receipt set	What this paper contributes
Is pretraining done enough to hand off to post-training?	Stable baseline, healthy occupancy, co-active overlaps not fragile on protected layers.	This is the paper's primary claim. Corollary 1 shows that output-only observables cannot certify handoff readiness near the handoff boundary when the concrete handoff margin moves off their level set. The DSV3 family prior calibrates the healthy reference geometry, and SFT-640 shows what a preserved stage transition looks like on this family.
Is post-training refining policy or repurposing the atlas?	Checkpoint-pair drift plus State2 and occupancy, read against the intended training stage.	This is the theorem-backed parallel extension. Corollary 2 shows that reward, loss, or benchmark surfaces alone cannot certify refinement near the refinement boundary when the concrete refinement margin varies independently. SFT-640 is the preserved case; Distill-640 is the stressed case in which scalar success and atlas preservation separate.
How far can we push post-training before forgetting risk becomes real?	Repeated gates on protected layers, overlap receipts, and occupancy telemetry.	The theorem does not prove a forgetting boundary. It shows only that any forgetting-risk margin built from atlas motion is underdetermined by output-only surfaces whenever that margin varies off their span. The historical run shows that damage and recovery can alternate, which is why the monitoring window matters.
Can these judgments be made online?	Fixed canary windows, periodic gates, resume-baseline semantics.	Proposition 4 says a sound local control policy needs atlas receipts whenever the ambiguity conditions hold. Geometry gates run at checkpoint-save cadence with below 3% overhead, so the required receipts are operationally cheap enough to collect live.

8.2 The four lifecycle questions and the receipts they require

Table 10 is the shortest summary of what this paper is trying to make possible in practice. Each question asks about a different object. That is why the required receipts differ. If one wants to know whether pretraining is done, whether RL or continual learning is still refining policy rather than rewriting capability, whether forgetting risk is becoming live, or whether a run should be promoted or halted online, the answer cannot come from a single scalar. That is the whole point.

8.3 How to interpret the metrics

The four drift functionals and the occupancy statistics interact. The following decision table summarizes the joint interpretation. It is a claim-licensing table:

The fourth row of Table 11 is the Distill-640 post-shift pattern. It is the most dangerous because the scalar story stays positive while the structural story does not. This is the row to treat as an

Table 11 Joint interpretation of atlas metrics and occupancy. The rows should be read as claim-licensing states rather than as a scalar ranking of runs.

C2 trend	Occupancy	Loss	Interpretation
Stable/improving	Healthy	Improving	Objective improvement and structural usage are jointly consistent with safe continuation on the current horizon.
Stable/improving	Degrading	Improving	Objective improves, but handoff or preservation claims are unlicensed because usage is narrowing. Investigate.
Worsening	Healthy	Improving	Atlas motion is visible without occupancy collapse; the relevant question becomes which layers are moving and whether that motion is acceptable for the current stage.
Worsening	Degrading	Improving	**Scalar improvement without geometric recovery.** Objective progress is licensed; structural health is not. Escalate.
Worsening	Degrading	Stalling	Both the scalar and structural stories are bad. Intervene or reset.

escalation condition in practice. If loss is improving while C2 worsens and occupancy remains degraded, a careful operator should become less optimistic rather than more, because the receipts now answer different questions in incompatible ways.

8.4 Why this playbook is earned by the experiments

The recommendations in this section are evidence-backed. Each one is the shortest rule that survives the two runs in Section 7. Two runs do not settle all monitoring questions. They do force the recommendations below, because each one is anchored to an explicit failure of a simpler rule.

This is also why the design guide is deliberately asymmetric. We do not average all signals into one dashboard score. We are asking which claims each experiment rules in and rules out. SFT-640 rules out "healthy resume implies preserved model." Distill-640 rules out "falling loss implies atlas preservation" and "rising margins imply healthier routing." What remains after those exclusions is the smaller set of rules we are willing to recommend.

8.5 How many gates before you act?

Distill-640 is useful here because it contains both an early warning and a false dawn. If we intervened aggressively at the first bad gate, we could fairly be accused of overreacting to a transient fluctuation. If we waited until the final gate, we would be confusing patience with denial. The default policy suggested by the present evidence is therefore neither "panic immediately" nor "wait for certainty." It is a staged escalation rule.

Table 13 does not assume that all future runs will follow the same timeline. Its job is to make the operator's burden explicit. The monitor should tell you that something went wrong and when a bad pattern has persisted long enough that passive continuation stops counting as prudence.

Table 12 Why the design guide is evidence-backed rather than merely well argued. Each rule survives a concrete failure of a weaker policy in the experiments.

Rule	Evidence in this paper	Why the simpler rule fails
Re-baseline after any resume	SFT-640 resumes with healthy routing and a GREEN geometry baseline, yet no preservation verdict is available.	"The resumed job looks healthy" is an operational statement rather than a preservation statement. Without a post-baseline gate the monitor answers the wrong question.
Do not let State1 stand in for State2	Distill-640 shows rising margins ($0.009 \rightarrow 0.014$) while C2 worsens and occupancy stays degraded.	"Wider margins mean healthier routing" fails once occupancy narrows; the surviving competitions can widen inside a worse stratification.
Require State2 + occupancy together for structural-health claims	Distill-640 loss improves monotonically while C2 turns negative at step 500 and dead experts jump to 285.	Loss-only monitoring would answer only the objective question; occupancy-only monitoring would answer only the usage question. The paired read is what licenses the structural claim.
Use parameter distance only as a secondary surface	Distill-640 changes only $\sim$12B of 691B parameters and leaves experts frozen, yet the atlas read is strongly negative.	A small parameter move can still rewrite how the frozen atlas is queried. Parameter distance misses coordinate-driven structural damage.
Protect mid-band layers first	The worst per-layer C2 degradation is concentrated in layers 8–20.	Uniform protection wastes budget on layers that are carrying less overlap damage. The layer-local signal changes where the first intervention should land.

8.6 Which claims the receipts do not license

Not every receipt can certify every claim. The practical claim-licensing logic is asymmetric. The temptation is to average everything into one comfort score and hope that the good news and bad news cancel out. We think that is a mistake.

Loss does not license preservation. If loss improves while C2 worsens and occupancy remains degraded, the right conclusion is that the optimizer has found a cheaper region of the current training objective than the operator intended. That can still be useful progress, but it differs from preserving the routed object. The optimizer may be doing its job while the operator is asking the wrong question.

Occupancy does not license co-active compatibility or preservation. Healthy entropy, low CV, and zero dead experts are good news. They are useful operational news, but preservation evidence still requires a valid comparison. SFT-640 makes the point clearly: after the cluster recovery, the live line looked healthy, but the only geometry receipt was still the resumed-checkpoint baseline. There was nothing to compare it against yet. This is why resumed-checkpoint measurements need special handling. Otherwise a cluster recovery gets promoted into a scientific conclusion.

State 1 does not license global routed health. This is the easiest mistake to make because wider margins look reassuring. On Distill-640, the mean boundary margin rises while the overlap signal gets worse. The router becomes more decisive exactly while the atlas becomes less healthy. Once occupancy has narrowed, larger margins can simply mean that fewer experts are still meaningfully

Table 13 Default escalation rule suggested by the experiments. This is the most conservative policy that survives the Distill-640 progression and the SFT-640 control.

Observed gate pattern	Default action	Why this is the least-bad rule supported by the paper
First bad gate after a clean baseline	Revoke any preservation claim, but collect the next gate before escalating.	Distill-640 step 500 is already enough to say the post-shift phase has left the earlier healthy regime, but not yet enough to distinguish a persistent structural problem from an early fluctuation.
One encouraging rebound without occupancy repair	Stay on alert. Do not issue an all-clear.	Distill-640 step 1000 briefly improves on C2 while leaving occupancy unchanged, then degrades further. A single rebound is too cheap to trust.
Two subsequent worse gates with the same occupancy pathology	Intervene at matched scalar quality.	Distill-640 steps 1500 and 2000 show that repeated degradation after the rebound is no longer plausibly noise. Waiting longer is not caution; it is drift.
Resume checkpoint with only a new baseline measurement	Do nothing except wait for the first valid post-baseline comparison.	SFT-640 shows that a healthy resume can answer the operational question while leaving the preservation question untouched.

Table 14 Default intervention ordering implied by the atlas framework. The table is intentionally opinionated: it tells the operator what to try first before escalating to broader atlas motion.

Observed pattern	Likely failure object	First intervention to try
Loss improves, C2 worsens, occupancy degrades	Co-active overlap failure plus narrowed visited region	Endpoint/adaptor regularization or router-prior control at matched scalar success. Do not keep training passively.
Loss improves, occupancy healthy, State 2 worsens	Atlas drift without occupancy collapse	Inspect protected layers first; tighten chart-preserving controls before changing the experts.
Occupancy healthy after resume, no post-baseline geometry gate yet	Missing receipt rather than yet a model failure	Do nothing except collect the first valid post-baseline gate.
State 1 worsens while State 2 stable	Boundary-local routing issue	Check gain drift, thresholding, and boundary-local router behavior before escalating to broad intervention.
State 2 improves and occupancy broadens	Genuine repair plausible	Continue, but require repeated gates before declaring recovery.

in play.

8.7 First intervention by symptom

Operators do not need another ontology if it does not shorten the path to the next action. The useful question is "which first move follows from the failure object?" The useful question is "what should I try first once I know which object is failing?" Table 14 gives the default playbook implied by the current evidence. The table is intentionally not exhaustive. Its job is to encode the first move, because the first move is where teams usually waste the most time.

8.8 What to protect during adaptation

The per-layer breakdown in Table 8 suggests a natural layer-selection rule for chart-preserving adaptation:

1. Identify mid-band layers with the highest specialization persistence and the worst C2 sensitivity.
2. Designate these as the protected layer set $\mathcal{L}_{\text{protect}}$.
3. During adaptation, spend optimization budget first on surfaces that do not move the protected layers: readout, adapters, endpoint layers.
4. Escalate to router-prior control, chart insertion, or continued pretraining only if chart-preserving policies fail to reach the target objective.

This ordering (adapters first, then endpoints, then router control, then chart insertion, then broad interior motion) is the default surface ordering implied by the atlas framework. Deviations should be justified by evidence that the objective is knowledge-expansive (cannot be solved by any chart-preserving policy).

8.9 What counts as a real fix?

Distill-640 gives us a useful stress case. It is easy to produce a run that looks better by scalar criteria while the routed object gets worse. That means the standard for a successful intervention has to be stricter than "the loss went down some more." For the purposes of this framework, a fix counts as real only if it improves the geometry at matched scalar quality.

Concretely, a matched-success intervention should satisfy four checks:

1. State2 improves materially relative to the stressed run.
2. Occupancy broadens materially (dead experts down, entropy up, CV down).
3. The gain is not carried only by State1.
4. The improvement persists across repeated gates rather than appearing at a single convenient checkpoint.

This standard is intentionally harder than the usual training narrative. It is also closer to what the operator actually wants. If an intervention restores scalar quality by making the run more concentrated, more brittle, or more locally decisive inside the same degraded visited region, then it has not repaired the atlas. It has merely found a different shortcut through the same failure mode.

8.10 What we would do on the two runs

The framework is only useful if it changes intervention behavior. On Distill-640, another small loss improvement does not answer the question. The next meaningful experiment is a matched-success intervention that tries to raise C2 and reopen occupancy without giving back the scalar win. That could mean stronger endpoint regularization, explicit router-prior control, or a chart-preserving adapter policy. What it should not mean is another week of passive training with no geometric target. The easiest mistake here is to call the post-shift phase "mostly working" and leave it alone. That would be a monitoring failure.

On SFT-640, the framework says the opposite. Do not intervene yet. Do not celebrate yet, either. The operator should simply wait for the first post-baseline geometry gate and only then decide whether anything was preserved. The bookkeeping is simple, but it matters: resumed training and preserved geometry are different claims.

9 Discussion

What the atlas framework provides. Scalar monitoring misses three things: (1) a decomposition into four drift types, each independently computable from checkpoint pairs; (2) a formal distinction between the overlapping expert cover (where co-active compatibility lives) and the disjoint active-set cells (where swap boundaries live); (3) a prediction, confirmed by the Distill-640 progression, that scalar improvement can occur without geometric recovery. The paper is intentionally opinionated about this. We think the field has been too willing to accept scalar improvement as a sufficient summary of post-training quality in routed systems. That was tolerable when the models were smaller and the routing story was mostly about utilization. It becomes much less tolerable when a model can quietly turn into a smaller, worse-routed capability system while still making the objective look better. More materially, the framework turns four vague lifecycle questions into object-level ones and then proves why output-only observables cannot settle their atlas-sensitive versions near the relevant decision boundaries. Handoff readiness is the primary instantiation of that result in this paper. Refinement-versus-repurposing is the parallel extension. Both become atlas questions when the relevant lifecycle scalar moves independently of the chosen loss/reward/benchmark surface. In that sense the atlas is necessary.

Why this matters beyond one monitored run. The broader point is that large-MoE systems still lack a convincing language for several decisions that become unavoidable as models get larger and post-training gets more ambitious. How do we know pretraining has given us the capability-bearing atlas we want? How do we monitor RL or continual learning without confusing policy improvement with capability rewriting? When should an underperforming post-training run be fixed with better optimization, and when should it be sent back to pretraining because the atlas itself is missing structure? We do not claim that one paper closes all of those questions. We do claim something sharper: those questions are not observable from output-only surfaces alone near the corresponding lifecycle boundaries whenever the relevant lifecycle scalar moves along an output level set. That is a mathematical statement with operational consequences.

Chart-preserving adaptation. The atlas view classifies post-training by its geometric effect. *Chart-preserving* adaptation refines how the atlas is used (readout, calibration, policy, bounded local behavior) without repurposing the mid-band interior. *Chart-redefining* adaptation rewrites chart content, boundaries, or transition rules at scale. RL, SFT, and ordinary continual learning should be chart-preserving by default. If a task requires broad chart redefinition to succeed, it should be classified as a continued-pretraining or knowledge-expansion objective rather than forced through a standard post-training recipe. This is a stronger claim than the field usually makes, but it is also closer to how practitioners already reason when runs go wrong. If post-training only succeeds by broadly repurposing the mid-band atlas, the model is rewriting capability instead of refining policy. It is becoming a different capability system. That may be acceptable, but it should be named honestly and optimized as such. In practice, teams already feel the difference between "this run refined the model" and "this run made the model into something else." What they usually lack is a language strong enough to say why.

The router as semantic interface. The atlas structure makes the router's role explicit: it is the semantic interface to the expert atlas. It determines which charts co-activate, which swap boundaries are traversed, and which semantic neighborhoods count as adjacent. Router drift can change the operational meaning of the atlas even when expert weights are held fixed (as Distill-640 demonstrates through coordinate drift in the backbone). Router health monitoring should therefore be treated as a first-class semantic concern rather than as secondary telemetry. This is why router stability feels disproportionately painful in large-MoE training. The router is the piece that decides whether the model is still consulting the same knowledge neighborhoods, and it does so under a training signal that is perfectly happy to exploit shortcuts. In our view this is the operational center of gravity for large-MoE training. The router is both the hardest thing to keep stable and the thing that most quickly turns a nominally enormous model into a much smaller practical one. That is why we have been willing to build so much of the paper around it.

Limitations. We have two runs on one model family on one cluster. That alone is not enough to claim universality. It is enough to show that the object exists and that ignoring it has consequences, but a fair reader should want to see the same decomposition tested on unfrozen-expert training, on a second MoE family, and on downstream task evaluation before treating the framework as settled. We want to see those experiments too.

The transition incompatibility identity (Eq. 14) is a one-swap idealization. Real boundary crossings involve multiple expert swaps and higher-order normalization effects. We use it because it is the cleanest expression the geometry gives us rather than because we believe boundaries are always this simple.

The per-layer breakdown shows a clear mid-band concentration, but we do not yet have a principled selection rule that generalizes across architectures. The layer grouping in Table 8 is suggestive rather than prescriptive.

Finally, we do not know why the atlas damage in Distill-640 is as large as it is. The experts are frozen, the router is frozen, and only 12B of 691B parameters move. The size of the C2 swing (0.146) surprised us. We suspect the answer involves how attention-pattern changes redirect token paths through the angular manifold, but we have not isolated the causal mechanism. That question is open.

10 Related Work

Sparse MoE routing. The sparsely-gated MoE layer [1] established sparse routing as a practical scaling mechanism, using a trainable gating network to select a sparse combination of expert feed-forward networks. The original work already identified two problems that recur throughout the subsequent literature and are directly relevant to our atlas characterization: the tendency of gating to collapse onto a few favored experts (which we formalize as occupancy degeneration), and the need for auxiliary losses to maintain load balance (which operates on aggregate statistics but not on the geometric structure we characterize here). GShard [2] scaled the approach to 600B parameters across 2048 TPU cores and introduced group-level top-2 gating with an auxiliary loss, demonstrating that sparse MoE can achieve strong multilingual translation results. Switch Transformers [3] simplified the routing to top-1, reduced communication costs, and showed 7× pre-training speedups over T5, while also identifying training instabilities at scale that were later addressed by ST-MoE [8] through the router z-loss and systematic fine-tuning studies. GLaM [4] scaled to 1.2 trillion parameters with 64 experts and top-2 gating, demonstrating that sparse MoE is

a credible scaling alternative to dense transformers. Expert-choice routing [13] inverts the routing direction so experts select tokens rather than vice versa, which changes the active-set structure but not the underlying angular geometry we characterize. DeepSeekMoE [9] and DeepSeek-V3 [5] introduce shared experts that are always active alongside the routed experts; in our framework, shared experts contribute to the transport baseline rather than the chart-local atlas. Concurrent with our work, Liu [14] describes MoEs as soft clustering with overlapping expert-local charts and probes expert-local Jacobian spectra. We cite it as concurrent empirical support for the chart picture, while emphasizing the gap in scope: that work is empirical and spectral, and it does not formalize the RMS-induced angular manifold, immersion cover, active-set stratification, or transition maps that define the atlas here. GrMoE [15] is geometric in a different sense, replacing standard gating with routing on the Grassmannian manifold of subspaces; its geometric object is the router mechanism, not the routed state-space atlas we study.

Our work is complementary to all of the above. We do not propose a new routing algorithm, a new stability technique, or a new expert architecture. We characterize the geometric object that *any* top-k sparse MoE with RMS normalization creates, and show that this object's health can be measured independently of scalar loss. One way to say it is that prior work largely asked how to make sparse routing trainable and efficient. We are asking what sparse routing builds once it has trained, and how to tell when that built object is being preserved or damaged.

Continual learning and catastrophic forgetting. The continual-learning literature has long treated forgetting as a stability-plasticity tradeoff. Elastic weight consolidation [16] penalizes movement of parameters that are important for previous tasks, measured by the Fisher information matrix. Replay-based methods [17] maintain memory buffers of previous-task examples. Both approaches operate on parameter space and treat all parameters as interchangeable.

The atlas view provides a different decomposition. In a large MoE, forgetting has structure: chart content drifts, chart boundaries shift, or the transition structure between expert neighborhoods is rewritten. These are different failure modes with different signatures and different remedies. Parameter distance treats them all the same. This decomposition is finer-grained than parameter distance because it separates the distinct roles of the router, the experts, and the backbone. A small router change can produce large boundary drift (reassigning tokens to different expert subsets), while a large adapter update can produce noticeable parameter motion while leaving the protected atlas nearly unchanged. The atlas view thus reframes forgetting in MoE systems as a structured geometric problem with explicit failure objects, instead of a generic parameter-distance penalty. This does not make the continual-learning problem easy. It does make it less vague. Instead of asking whether the model moved "too much," we can ask whether the knowledge-bearing chart structure was preserved, narrowed, or rewritten.

Hyperspherical and angular geometry. There is substantial precedent for treating learned representations in angular terms. SphereFace [18] constrains face embeddings to a hypersphere for discriminative recognition. Hyperspherical prototype networks [19] place class prototypes on a sphere for few-shot learning. Information geometry [20] has long argued that parameter or state spaces carry non-Euclidean structure relevant for optimization. Geometry-Preserving Aggregation for MoE embedding models [21] independently reports that expert outputs lie on a shared hyperspherical manifold with specialization expressed through angular separation, which is consistent with the angular picture we develop here.

Our use of the sphere is different in kind. The angular manifold $\mathbb{S}^{d-1}$ arises from RMS normalization

inside the routed transformer, before the router even sees the state. The sphere is a consequence of the normalization architecture that shapes every subsequent computation rather than an imposed output-space objective. This distinction matters because it means the geometry is present in every RMS-normalized MoE regardless of the training objective, the data domain, or the downstream task.

Routing interpretability and expert specialization. Empirical analyses of trained MoE models [22] have shown that routing carries interpretable structure: experts specialize on syntactic categories, semantic domains, or language-specific patterns. The atlas framework provides a formal account of what this specialization means geometrically. A well-specialized atlas has coherent chart-local fields (each expert computes a locally meaningful update on its domain), compatible overlap relations (co-active experts produce consistent fields), and small transition incompatibility at swap boundaries (routing flips do not produce large field discontinuities). When any of these properties fails, the atlas is poorly formed regardless of whether the individual experts appear specialized in isolation.

11 Conclusion

We began with a practical complaint: large-MoE training is easy to misread. Loss can improve while the routed object deteriorates, and a run can look operationally healthy before any preservation question has been answered. That complaint points to a broader gap. We still lack a strong foundation for deciding when pretraining is finished, how to monitor post-training for genuine improvement, and when continual learning has crossed into catastrophic forgetting.

Our claim is that these questions are hard because the routed object has structure that the usual monitoring stack does not name. RMS normalization induces an angular manifold $\mathbb{S}^{d-1}$ on which expert down-projections form a canonical immersion cover with linear overlap relations. The router separately induces a disjoint active-set stratification. Together they create a layer-local atlas whose health decomposes into four drift types: coordinate, boundary, transition, and content. Theorem 2 then shows that if an atlas-sensitive lifecycle scalar varies along an output level set, loss, reward, and benchmark surfaces cannot identify that lifecycle state even in principle. This is the paper's main claim. The pretraining-to-post-training boundary is an atlas boundary before it is a loss threshold. That is why handoff readiness and refinement-versus-repurposing become atlas questions near their decision boundaries. The first is the center of this paper. The second is the theorem-backed extension we are strengthening in parallel. On a 640-expert MoE, we observe the predicted decoupling in a later post-shift phase of an otherwise successful distillation run: loss improves from 5.25 to 4.48 while overlap compatibility worsens from -0.027 to -0.063 and routing occupancy remains degraded.

The practical lesson is direct. In large MoE systems, keeping the router stable is both hard and semantically central. Scalar loss, occupancy, and parameter distance each miss part of the story. If the routed object has geometry, then training, RL, and continual learning all need geometric receipts. The default question therefore changes from "is the objective improving?" to "is the atlas being preserved, broadened, or repurposed?"

References

[1] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

[2] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021.

[3] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

[4] Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V. Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. Glam: Efficient scaling of language models with mixture-of-experts. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

[5] DeepSeek-AI. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2025. Version 2, last revised 18 Feb 2025.

[6] Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems 32*, 2019.

[7] Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

[8] Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. St-moe: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022.

[9] Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wenge Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.

[10] Zeyuan Allen-Zhu. Physics of language models: Part 4.1, architecture design and the magic of canon layers. *arXiv preprint arXiv:2512.17351*, 2025.

[11] Zhenda Xie, Defa Zhu, Hongzhi Huang, Zihao Huang, Yutao Zeng, Yunyao Mao, Banggu Wu, Qiyang Min, Xun Zhou, Damai Dai, Deli Chen, Chong Ruan, and Wenfeng Liang. mhc: Manifold-constrained hyper-connections. *arXiv preprint arXiv:2512.24880*, 2025.

[12] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, 2018.

[13] Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Y. Zhao, Andrew M. Dai, Zhifeng Chen, Quoc V. Le, and James Laudon. Mixture-of-experts with expert choice routing. In *Advances in Neural Information Processing Systems 35*, 2022.

[14] Feilong Liu. Mixture-of-experts as soft clustering: A dual jacobian-pca spectral geometry perspective. *arXiv preprint arXiv:2601.11616*, 2026.

[15] Ibne Farabi Shihab, Sanjeda Akter, and Anuj Sharma. Grassmannian mixture-of-experts: Concentration-controlled routing on subspace manifolds. *arXiv preprint arXiv:2602.17798*, 2026.

[16] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis,

Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

[17] German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

[18] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 212–220, 2017.

[19] Pascal Mettes, Elise van der Pol, and Cees G. M. Snoek. Hyperspherical prototype networks. In *Advances in Neural Information Processing Systems 32*, 2019.

[20] Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

[21] Sajjad Kachuee and Mohammad Sharifkhani. Geometry-preserving aggregation for mixture-of-experts embedding models. *arXiv preprint arXiv:2602.14039*, 2026.

[22] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lenber, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gerber, Thibaut Couairon, Timothée Vayer, Vignesh Mishra, Wasi Mohammad Syed, and William El Sayed. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

A RMSNorm Differential Derivation

Write $r = r_\epsilon(x)$ and $z = z_\epsilon(x)$. Since $r_\epsilon(x) = (\|x\|_2^2/d + \epsilon)^{1/2}$, its differential is $\mathrm{D}r_\epsilon(x)[u] = x^\top u/(dr)$. By the product rule:

$$\mathrm{D}z_\epsilon(x)[u] = \frac{u}{r} - \frac{x}{r^2}\frac{x^\top u}{dr} = \frac{1}{r}\left(I - \frac{xx^\top}{dr^2}\right)u = \frac{1}{r}\left(I - \frac{zz^\top}{d}\right)u. \tag{33}$$

For the tangent-radial decomposition: $zz^\top/d = (\|z\|_2^2/d)\hat{x}\hat{x}^\top$ and $\|z\|_2^2/d = 1 - \epsilon/r^2$, giving:

$$\mathrm{D}z_\epsilon(x)[u] = \frac{1}{r}(I - \hat{x}\hat{x}^\top)u + \frac{\epsilon}{r^3}\hat{x}\hat{x}^\top u = \frac{1}{r}\,u_{\mathsf{T}} + \frac{\epsilon}{r^3}\,u_{\mathrm{rad}}. \tag{34}$$

B Experimental Details

Geometry evaluation protocol. State1 (boundary margin): mean margin between the k-th and $(k{+}1)$-th router logits across the probe family and protected layers. State2 (C2): correlation-based co-active overlap compatibility; positive values indicate improvement over baseline, negative values degradation. State3 (NLL): mean negative log-likelihood on the probe family. All metrics use deterministic sampling on fixed canary windows; bootstrap over windows rather than tokens.

Hardware. Distill-640: 8× B200 (192 GB HBM), single node, NVLink. SFT-640: 64× B200 across 16 nodes, GB300 NVL72, CephFS. Both use BF16 for attention, FP8 for expert forward.

Training data. Sparse logit distillation artifacts (DSTL v1, $K_{\max} = 64$, $k_{\min} = 16$, mass target 0.98) from OLMo-1025 midtrain (≈98B tokens), Harmony-tokenized.