The Gradient Starvation Envelope

A Formal Compliance Primitive for Sparse Mixture-of-Experts Training Dynamics

Clause AI-2

Auburn Patent Family

Fields

PSpecification — Version 1.0

February 2026

This document is the unredacted specification.
All derivations, implementation details, and calibration constants are included. Do not distribute without authorization.

Abstract

Sparse Mixture-of-Experts (MoE) architectures decouple inference cost from model capacity by activating only a fraction of total parameters per input token. This architectural choice introduces a class of optimization pathologies—routing collapse, dead experts, gradient starvation, and stale activation persistence—that are qualitatively distinct from those in dense networks and that no existing mitigation formally guarantees to prevent. Simultaneously, emerging regulatory frameworks (EU AI Act Article 11/53, NIST AI 600-1, SR 11-7) mandate training documentation and robustness measures but provide no mathematical criteria for training health and no verification mechanism beyond self-reporting. This paper formalizes the **Gradient Starvation Envelope** (Eq. 114.1), a Lyapunov-style differential inequality on per-expert gradient variance that serves as a compliance primitive bridging MoE training dynamics, formal verification, and enterprise AI governance. A comprehensive literature review confirms that while each constituent element—gradient starvation theory, MoE load-balancing heuristics, control-theoretic convergence analysis, and regulatory documentation mandates—is individually well-developed, their integration into a single formal compliance condition is novel. The Envelope transforms the qualitative concern of gradient starvation into an auditable property of the training trajectory, providing the first mathematically rigorous "training health certificate" for sparse MoE architectures.

Contents

1 Introduction

The trajectory of contemporary artificial intelligence has been defined by a paradigm shift from dense, monolithic architectures to sparse, conditional computation. As the demand for model capacity outpaced the growth of memory bandwidth and synchronous compute clusters, the field increasingly adopted Sparse Mixture-of-Experts (MoE) architectures. By activating only a fraction of total parameters per input token—decoupling inference cost (FLOPs) from model size (parameters)—architectures such as the Switch Transformer, GShard, DeepSeek-V3, and Mixtral have enabled training at parameter counts scaling into the trillions while maintaining tractable inference costs.

However, this architectural evolution has introduced a class of optimization pathologies that are qualitatively distinct from those observed in dense networks. The introduction of discrete routing decisions into the computation graph creates a non-convex, dynamic optimization landscape where expert sub-networks compete for data and gradient signal. This competition, if left unregulated, frequently resolves into degenerate equilibria known as *routing collapse* or *dead experts*, where the vast majority of model capacity remains unutilized while a small subset of parameters effectively reverts to a dense bottleneck. More insidiously, even when experts appear active, they may suffer from *Gradient Starvation*—a phenomenon where the optimization process preferentially updates parameters associated with high-frequency, statistically dominant features, starving specialist parameters of the gradient signal required to learn subtle, long-tail distributions.

These dynamics are no longer merely academic concerns regarding training efficiency; they have become critical governance liabilities. As the deployment of foundation models permeates high-risk sectors—finance, healthcare, critical infrastructure—the opaque and unstable nature of MoE training dynamics creates unacceptable risks regarding reliability, privacy, and compliance. Emerging regulatory frameworks, most notably the European Union's Artificial Intelligence Act (EU AI Act, Regulation (EU) 2024/1689) and the United States' NIST AI Risk Management Framework (AI RMF), are establishing rigorous standards for technical documentation and training transparency. Simultaneously, financial model risk management standards (SR 11-7, OCC 2011-12) demand "conceptual soundness"—a bar that heuristic load-balancing losses and capacity factors fail to clear.

This work presents the **Gradient Starvation Envelope**—a formal compliance condition based on the exponential decay of gradient variance across parameter subsets. By bridging control-theoretic stability analysis (Lyapunov methods) with empirical deep learning dynamics, the Envelope provides a verifiable mechanism to certify that a training run has maintained *equiconnectedness* across its expert manifold, preventing the formation of frozen, starved, or privacy-leaking sub-networks.

The Envelope is positioned not as a new optimization technique but as a **formal compliance primitive**—a minimal, monitorable, mathematically rigorous condition that transforms the qualitative concern of gradient starvation into an auditable property of the training trajectory. It bridges the gap between what the MoE training community knows is important and what the governance community can currently verify.

1.1 Scope and Organization

Section 2 establishes the theoretical foundations of gradient starvation, distinguishing it from vanishing gradients and surveying the NTK-based characterization through 2025. Section 3 catalogs the empirical evidence for dead experts and routing collapse across

every major MoE architecture from Shazeer et al. (2017) through DeepSeek-V3, and systematically evaluates every existing mitigation strategy. Section 4 examines gradient flow degradation during extended fine-tuning (SFT, RLHF, DPO) including LoRA/QLoRA gradient collapse and dead neuron pathologies. Section 5 connects stale activations to differential privacy exposure and safety alignment bypass. Section 6 presents the full mathematical formulation of the Gradient Starvation Envelope (Eq. 114.1) in continuous and discrete time. Section 7 establishes the Lyapunov heritage and control-theoretic foundations. Sections 8–9 map the Envelope to enterprise applications and regulatory compliance. Section 10 provides example audit configurations. Section 11 differentiates from competing formalisms. Section 12 addresses the five strongest anticipated reviewer objections. Section 13 identifies publicly available resources for empirical validation. Section 14 concludes.

2 Gradient Starvation: Theoretical Foundations

To establish the necessity of the Gradient Starvation Envelope, one must first deconstruct the phenomenon of gradient starvation itself. Often conflated with the historical problem of vanishing gradients, gradient starvation represents a distinct, spectral failure mode of gradient-based optimization in high-dimensional spaces.

2.1 Formal Definitions and the Distinction from Vanishing Gradients

Definition 2.1 (Vanishing Gradients)**.** *Let f_ℓ denote the activation function at layer ℓ in a network of depth L. The vanishing gradient problem occurs when the product of derivatives $\prod_{\ell=1}^{L} |f'_\ell(z_\ell)| \to 0$ as L grows, causing exponential decay of the gradient vector's norm as it backpropagates through successive layers. This is primarily a numerical stability issue caused by the contractive derivatives of saturating activation functions ($|\tanh'(x)| < 1$, $|\sigma'(x)| < 1$), producing a compounding reduction in signal amplitude.*

Modern architectures largely mitigate vanishing gradients via residual connections (ResNets), normalization layers (LayerNorm, RMSNorm), and non-saturating activations (ReLU), which facilitate signal propagation through depth.

Definition 2.2 (Gradient Starvation (Pezeshki et al., 2021))**.** *Consider a dataset containing multiple predictive features $\{\phi_k\}_{k=1}^{K}$ with varying spectral properties. Gradient starvation occurs when cross-entropy loss creates a shared multiplicative factor $\sigma(-y \cdot z)$ across all feature-specific gradient components. As one feature drives correct predictions, $\sigma(-y \cdot z) \to 0$ globally, suppressing gradients for all remaining features. For a network f_θ trained with cross-entropy loss $\mathcal{L}$, the gradient contribution from feature ϕ_k satisfies*

$$\nabla_{\theta_k} \mathcal{L} = \underbrace{\sigma(-y \cdot f_\theta(x))}_{\text{shared suppression}} \cdot \underbrace{\nabla_{\theta_k} f_\theta(x)}_{\text{feature-specific}}, \tag{1}$$

where the shared suppression factor drives all feature-specific gradients toward zero once any single feature achieves high predictive accuracy.

The starvation rate is governed by the spectral gap between the leading Neural Tangent Kernel (NTK) eigenvalue and those associated with weaker features. The gradient

flow naturally decouples into "fast" and "slow" components, with fast components effectively orthogonalizing the residuals against slow components, halting their learning. The proposed mitigation, Spectral Decoupling, adds an ℓ_2 penalty on logits to prevent premature loss saturation.

Remark 2.1. *Gradient starvation is* ***inter-feature competition*** *within the same gradient computation; vanishing gradients are* ***inter-layer attenuation*** *of the chain rule through depth. Gradient starvation occurs even in two-layer networks. The mechanisms are orthogonal, though the MoE literature occasionally conflates them in informal discussions of "dead experts."*

2.2 The Broader Simplicity Bias Landscape

Gradient starvation is situated within a broader understanding of simplicity bias in neural networks. Shah et al. ("The Pitfalls of Simplicity Bias in Neural Networks," NeurIPS 2020, arXiv:2006.07710) and Geirhos et al. ("Shortcut Learning in Deep Neural Networks," *Nature Machine Intelligence*, 2020) describe the overarching phenomenon whereby gradient descent preferentially latches onto "easy" features—those with large singular values in the NTK—and rapidly minimizes the loss associated with them.

Zhang et al. ("Saddle-to-Saddle Dynamics Explains a Simplicity Bias," arXiv:2512.20607, December 2025) provide a unifying theoretical framework showing gradient descent progressively learns features through saddle-to-saddle dynamics, extending the theoretical understanding of *why* starvation occurs. Xu et al. ("Feature Contamination," arXiv:2406.03345, 2024) identify a related but distinct mechanism: networks do not merely fail to learn starved features—they **actively contaminate** useful features by co-learning noise.

2.3 Plasticity Loss and Activation Function Design

A significant development through 2025 has been the connection between gradient starvation and the *loss of plasticity* in continual learning. Plasticity refers to a model's ability to adapt to new data distributions after pre-training. The dead-unit problem in ReLUs—where neurons outputting zero have zero gradients—creates dead zones in the optimization landscape.

Research surveyed by Klein et al. (2024) and detailed in 2025 preprints identifies a "Goldilocks Zone" for activation function slopes. Standard ReLUs (slope 0 for $x < 0$) exacerbate starvation. Conversely, activations with high negative slopes (near 1.0) cause optimization instability due to landscape stiffness. The optimal regime for sustaining gradient flow and preventing starvation lies in negative slopes between 0.6 and 0.9 (e.g., Randomized Smooth-Leaky activations). This "moderate leak" ensures a non-zero gradient floor, preventing the variance of the gradient from collapsing to zero even for suppressed features—essentially enforcing a minimal level of metabolic activity in the parameter space.

2.4 The MoE Gradient Starvation Gap

In the MoE context, gradient starvation is exacerbated by the routing mechanism. If a "generalist" expert captures a high-frequency token pattern early in training, it reduces the global loss significantly. The router, driven by this loss reduction, routes more tokens to this generalist. "Specialist" experts, which might be better suited for rare or complex

tokens, are starved of data (and thus gradients) because the generalist provides a "good enough" approximation that satisfies the greedy router.

No formal bounds on gradient flow imbalance across MoE experts exist in the literature. The closest MoE-specific gradient analyses are:

- "Solving Token Gradient Conflict in Mixture of Experts" (arXiv:2406.19905, 2024) addresses gradient conflicts among tokens routed to the same expert but does not bound inter-expert gradient variance.
- "Dense Backpropagation Improves Routing for Sparsely-Gated Mixture of Experts" (OpenReview, 2024/2025) shows that unrouted tokens may lie in the span of routed tokens, enabling approximate dense gradient computation—addressing the sparse-gradient problem without formal bounds.
- The "Gradient Blackout" concept (Li, 2025) formalizes that unselected experts receive exactly zero gradient under Top-K routing, but this is a qualitative observation, not a quantitative bound.

2.5 Gap Analysis

The existing literature provides a rigorous NTK-based characterization of gradient starvation for monolithic networks trained with cross-entropy but:

1. Does not extend this to discrete routing in MoE architectures,
2. Does not define compliance conditions on gradient variance dynamics, and
3. Does not connect gradient imbalance to Lyapunov-style decay requirements.

The Gradient Starvation Envelope (Eq. 114.1) directly fills this gap by translating the qualitative concern of gradient starvation into a quantitative, monitorable, compliance-testable condition over parameter subsets. Table 1 summarizes the landscape.

Table 1: Gap Analysis: Gradient Starvation

Feature	Existing Literature (2020–2025)	Gap Filled by Eq. 114.1
Scope of Definition	Primarily defined for dense networks (CNNs, MLPs) and feature imbalance (Pezeshki et al., 2021).	Extends to routed sub-networks in MoE, linking starvation to variance of gradients across expert clusters.
Mechanism	Attributed to NTK spectral properties and simplicity bias of SGD.	Attributes starvation in MoE to the feedback loop between router probability mass and expert gradient variance, modeled as a dynamical system stability problem.
Mitigation Strategy	Architectural tweaks (activations, spectral decoupling), data augmentation.	Formal differential inequality (Lyapunov constraint) as a compliance boundary.
Metric	Post-hoc analysis of feature learning; accuracy on "hard" subsets.	Real-time monitoring of gradient variance decay rates, enabling "training certificates" based on dynamic compliance.

3 Dead Experts and Routing Collapse: The Pathology of Sparsity

The central promise of Mixture-of-Experts architectures is conditional computation: the ability to scale capacity without scaling compute. However, this promise is predicated on the assumption that the gating network (the "router") utilizes the available experts effectively. The empirical history of MoE development is largely a history of fighting the inherent tendency of these systems towards routing collapse.

3.1 Phenomenology of Collapse and Dead Experts

Definition 3.1 (Routing Collapse)**.** *Routing collapse occurs when the gating mechanism converges to a trivial solution where a small subset of experts (often just one or two) receives the vast majority of input tokens. This effectively reduces the massive sparse model to a small dense model, wasting the parameters of the unused experts and creating a compute bottleneck on the devices hosting the "popular" experts.*

Definition 3.2 (Dead Expert)**.** *An expert θ_i is dead at time t if it receives zero (or negligibly few) routed tokens over a sustained window, yielding $g_i(t) = \|\nabla_{\theta_i}\mathcal{L}(t)\|_2 \approx 0$. Once dead, the expert enters a vicious cycle: no data implies no gradient updates, which implies static representations (often at random initialization), which implies the router trusts it less, which implies fewer tokens. The gradient variance for a dead expert is effectively zero—the ultimate form of gradient starvation.*

In distributed training settings, dead experts are particularly costly: they represent allocated GPU memory that performs no useful work, lowering the effective Model FLOPs Utilization (MFU).

3.2 Empirical Evidence Across Major MoE Architectures

Routing collapse has been documented in every major MoE architecture since 2017. This subsection surveys the progression chronologically.

3.2.1 Shazeer et al. (2017): The Self-Reinforcing Feedback Loop

Shazeer et al. ("Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer," ICLR 2017, arXiv:1701.06538) identified the fundamental self-reinforcing feedback loop: favored experts train faster because they receive more data, so the router favors them more, so they receive even more data. This positive feedback drives the system toward degenerate equilibria where a single expert dominates.

3.2.2 GShard (Lepikhin et al., 2021): Capacity Overflow at Scale

GShard (Lepikhin et al., ICLR 2021, arXiv:2006.16668) introduced *capacity factors*—hard limits on per-expert token counts—and observed significant token overflow at 600B-parameter scale. The capacity factor mechanism addressed symptoms (overloaded experts) without treating causes (gradient starvation of underloaded experts).

3.2.3 Switch Transformer (Fedus et al., 2022): Auxiliary Loss Sweeps

The Switch Transformer (Fedus, Zoph, and Shazeer, JMLR 2022, arXiv:2101.03961) demonstrated that without load-balancing loss, the model degenerates to using only a few experts. The auxiliary-loss coefficient α was swept from 10^{-1} to 10^{-5}, with $\alpha = 10^{-2}$ found optimal. This paper established the load-balancing loss as the default mitigation—a position it retains despite fundamental theoretical limitations (see Section 3.3).

3.2.4 ST-MoE (Zoph et al., 2022): Scale-Dependent Instability

ST-MoE (Zoph et al., 2022, arXiv:2202.08906) is the most comprehensive study of MoE training instability. Key findings include:

- Expert collapse occurs "always without auxiliary loss."
- Instabilities are **scale-dependent**: techniques that help small models can hurt at XL scale.
- Input-jitter specifically degrades quality as models grow.
- Encoder experts specialize by token type (punctuation, proper nouns, etc.).
- Freezing expert layers during fine-tuning works "almost as well" as full updates—but this means frozen experts retain training-time specialization indefinitely.

3.2.5 Mixtral (Jiang et al., 2024): Natural Balance at Small Expert Counts

Mixtral (Jiang et al., 2024, arXiv:2401.04088) achieved natural balance with only 8 experts per layer. However, this may reflect the small expert count rather than a solved problem—the routing dynamics of 8-expert systems are qualitatively different from 64- or 256-expert systems.

3.2.6 DeepSeek-V3 (2024): Auxiliary-Loss-Free Balancing

DeepSeek-V3 (2024, 671B total parameters, 256 routed experts) pioneered auxiliary-loss-free balancing using dynamic bias terms updated outside backpropagation:

$$b_i \leftarrow \begin{cases} b_i + \gamma & \text{if expert } i \text{ is underloaded} \\ b_i - \gamma & \text{if expert } i \text{ is overloaded} \end{cases} \tag{2}$$

The model reported "remarkably stable" training over 14.8 trillion tokens with zero rollbacks. This bias-based approach achieves superior balance *and* quality without gradient interference, but provides no formal stability guarantees.

3.2.7 Cerebras Analysis (2025): Persistent Layer-Level Imbalance

A Cerebras analysis (2025) showed that even with modern routing strategies, early and late layers funnel most tokens to just 1–2 experts. The problem persists at the architectural level even when aggregate statistics appear healthy.

3.3 Mitigation Catalog: Heuristics Without Guarantees

The literature is replete with mitigation strategies. This subsection catalogs each with its formal mechanism, known failure modes, and whether it provides guarantees versus heuristic improvement.

3.3.1 Load-Balancing Loss (Auxiliary Loss)

The most common mitigation, introduced by Shazeer et al. (2017):

$$\mathcal{L}_{\text{balance}} = \alpha \cdot N \sum_{i=1}^{N} f_i \cdot P_i, \tag{3}$$

where f_i is the fraction of tokens routed to expert i and P_i is the average router probability for expert i.

Failure modes. A Princeton analysis demonstrated that the minimum of $\mathcal{L}_{\text{balance}}$ is *not* at the uniform distribution—a fundamental theoretical flaw. More critically, dead experts contribute zero to the loss ($f_i = 0 \Rightarrow \nabla_{\theta_i} \mathcal{L}_{\text{balance}} = 0$), so the loss **cannot revive dead experts**. The auxiliary loss is a soft constraint that competes with the primary cross-entropy loss. If collapsing to a single expert yields a lower total loss (e.g., because that expert is slightly better initialized), the model ignores the balancing penalty.

3.3.2 Router z-Loss

Introduced in ST-MoE (Zoph et al., 2022):

$$\mathcal{L}_z = \frac{\beta}{B} \sum_{i=1}^{B} \left(\log \sum_{j=1}^{N} \exp(h_j(x_i)) \right)^2, \tag{4}$$

where $h_j(x_i)$ denotes the router logit for expert j on input x_i.

Failure modes. The primary motivation was numerical stability in `bfloat16` training, preventing round-off errors from causing divergence. While it incidentally helps exploration by keeping probabilities softer, it provides no convergence guarantees and no direct control on gradient flow. It does not address expert utilization or gradient variance.

3.3.3 Capacity Factors and Token Dropping

A hard cap on per-expert tokens. If expert i is assigned more tokens than its capacity C, excess tokens are "dropped" (passed through a residual connection without processing).

Failure modes. Token dropping introduces non-determinism and information loss. It "hides" routing collapse by forcing the router to spread tokens or lose them. This can degrade performance on information-dense tasks and does not solve the underlying gradient starvation of underloaded experts.

3.3.4 Expert Choice Routing

Expert Choice routing (Zhou et al., NeurIPS 2022, arXiv:2202.09368) inverts the standard paradigm: each expert selects its top-K tokens, ensuring perfect load balance by construction.

Failure modes. Incompatible with autoregressive generation (tokens cannot wait for expert selection at inference time). Permits tokens to receive zero experts. Crucially, says nothing about gradient-norm balance—perfectly balanced token counts do not guarantee balanced gradient flow. An expert receiving many "easy" tokens may still have near-zero gradients.

3.3.5 Sinkhorn / Optimal-Transport Routing

Enforces near-exact balance through doubly-stochastic constraints on the routing matrix via the Sinkhorn-Knopp algorithm.

Failure modes. Over-constrains specialization. The optimal-transport solution may force tokens to experts with poor semantic fit simply to satisfy the balance constraint, degrading quality.

3.3.6 DeepSeek-V3 Bias-Based Balancing

Dynamic bias adjustment (Eq. 3) updated outside backpropagation. The closest operational mechanism to what Eq. 114.1 formally targets.

Failure modes. Ad-hoc proportional-control rule ($b_i \pm \gamma$) without formal stability guarantees. The choice of γ is a hyperparameter with no principled calibration. The mechanism provides no certificate of convergence to balanced allocation within any specified time window. Eq. 114.1 can be viewed as the formal certificate that DeepSeek's mechanism implicitly targets.

3.3.7 ReMoE (ICLR 2025)

Replaces Top-K with ReLU gating, providing fully differentiable routing and eliminating the discrete selection discontinuity.

Failure modes. ReLU gating permits variable numbers of active experts per token, complicating hardware load balancing. Does not inherently prevent gradient imbalance across experts.

3.3.8 Implementation-Level Pitfalls

An ACL 2025 paper ("Demons in the Detail") revealed that micro-batch-level load-balancing loss—the default in most open-source frameworks—**negatively affects expert specialization** compared to global-batch balancing. This finding implies that even the most widely deployed mitigation is implemented suboptimally in practice.

3.4 Continuous Monitoring: Emerging but Immature

Recent work (2024–2025) has begun to frame expert utilization as a continuous monitoring obligation rather than a one-time training trick:

- **MoE-MUI** ("Beyond Benchmarks," arXiv:2509.23933, 2025) proposes a Model Utilization Index tracking key neuron proportions across training iterations—the closest work to framing expert utilization as a continuous monitoring obligation.
- **LibMoE** (Nguyen et al., arXiv:2411.00918, updated February 2026) provides standardized tools for probing routing dynamics, stability, entropy, and expert selection patterns. Supports both pretraining and sparse-upcycling regimes.
- **DeepSeek-V3's bias mechanism** is implicitly a continuous monitoring loop, but without formal compliance criteria.
- **Soft Restarts** are emerging in the context of routing stability—periodically resetting the router's weights or adding noise to gating logits to break out of local collapse minima. These remain reactive interventions rather than proactive, formally bounded controls.

None of these constitute a formal compliance framework.

3.5 Gap Analysis

Every existing MoE mitigation is either heuristic (all auxiliary losses, z-loss, capacity factors) or structurally constraining (Expert Choice, Sinkhorn) with significant trade-offs. No method provides a formal guarantee that:

1. Dead experts will not form,
2. Formed dead experts can be revived, or
3. Gradient allocation will converge to a balanced state within a specified time window.

The Gradient Starvation Envelope fills this gap by specifying an **exponential rebalancing condition**—the differential inequality ensures that gradient variance must decay at rate δ or the training run is declared non-compliant. This transforms a heuristic concern into a verifiable obligation. Table 2 summarizes the landscape.

Table 2: Gap Analysis: Routing Stability

Feature	Existing Approaches (2017–2025)	Gap Filled by Eq. 114.1
Control Mechanism	Auxiliary Losses (soft, competitive) or Capacity Limits (hard truncation).	Differential Inequality (hard, dynamic constraint). Mandates minimum variance decay rate.
Trigger Condition	Constant application of loss term; no state-dependent trigger.	State-dependent: constraint active based on rate of change of gradient variance. Acts as a Control Barrier Function.
Guarantees	None. Optimization may still collapse if main loss outweighs auxiliary loss.	Formal guarantee of equiconnectedness. System forced to maintain valid gradient flow paths.
Dead Expert Revival	Load-balancing loss has zero gradient to dead experts; cannot revive them.	Variance floor ε ensures minimum gradient signal persists. Breach triggers soft restart.
Regulatory View	Viewed as "hyperparameter tuning."	Viewed as a "Training Certificate" artifact; proof of process conceptual soundness.

4 Fine-Tuning Stability and Parameter Subset Decay

The dynamics of MoE models become even more precarious during the fine-tuning phase, particularly with Supervised Fine-Tuning (SFT) and Reinforcement Learning from Human Feedback (RLHF) / Direct Preference Optimization (DPO). Fine-tuning represents a massive distributional shift: a foundation model pre-trained on trillions of diverse web tokens is suddenly adapted to a narrow distribution of high-quality instruction data or preference pairs.

4.1 Gradient Flow Degradation in SFT, RLHF, and DPO

Extended fine-tuning produces several documented gradient pathologies, each representing a distinct manifestation of gradient starvation in the post-pretraining regime.

4.1.1 DPO: Likelihood Displacement

Razin et al. (arXiv:2410.08847, 2024) demonstrate that DPO causes both preferred and dispreferred response likelihoods to decrease simultaneously—a phenomenon termed *like-*

lihood displacement. The DPO gradient includes an importance weight that vanishes when relative likelihood differences are large, creating a form of gradient starvation specific to preference learning. Understanding the impact of sampling quality (arXiv:2506.04272, 2025) further shows that low-quality preference pairs exacerbate this gradient suppression.

4.1.2 RLHF: The Squeezing Effect and KL Drift

Ren and Sutherland ("Learning Dynamics of LLM Finetuning," ICLR 2025) identify a "squeezing effect" where large negative gradients on dispreferred responses shift probability mass to unseen responses rather than preferred ones. Reward gradients become sharper over time, inducing compounding KL divergence drift even with PPO clipping ranges as small as 10^{-4} (arXiv:2509.20265). The gradient signal concentrates on an ever-narrowing subset of parameters, starving the rest.

4.1.3 Safety–Task Gradient Conflict

SafeGrad (Yi et al., arXiv:2508.07172, 2025) demonstrates that conflicting gradients between user-task and safety objectives progressively compromise alignment. When the cosine similarity between task gradients and safety gradients is negative, one objective must be sacrificed—typically safety, since task loss drives the primary optimization signal.

4.1.4 Gradient Concentration in SFT

Gradient-Mask Tuning (AAAI 2025) shows that a substantial proportion of SFT parameter updates are redundant: masking small-gradient parameters improves performance by 3.0% on average. This implies that gradient signal is highly concentrated in a small parameter subset—direct evidence that SFT induces gradient starvation across the majority of parameters.

4.1.5 MoE-Specific Fine-Tuning Pathology

In MoE architectures, the distributional shift of fine-tuning is channeled through the router. Because fine-tuning data is semantically narrow (e.g., "helpful assistant" dialogue), the router may learn to utilize only a small subset of "dialogue" experts. Experts containing critical world knowledge, coding ability, or reasoning heuristics are neglected. The "starved" experts do not sit idle—they become *stale*. Their weights drift relative to the active experts due to weight decay or lack of synchronization with shared layers. This leads to:

- **Catastrophic forgetting**: the model retains its chatty persona (active experts) but loses factual reliability or logical reasoning (starved experts).
- **Policy degradation**: in RLHF/DPO, where the objective maximizes a reward relative to a reference model, expert starvation causes the policy to collapse onto a narrow behavioral mode.

4.2 LoRA/QLoRA Gradient Collapse and Rank Deficiency

The industry trend towards Parameter-Efficient Fine-Tuning (PEFT) introduces additional gradient pathologies that interact with MoE dynamics.

4.2.1 Scaling-Induced Gradient Collapse

rsLoRA (Kalajdzievski, arXiv:2312.03732) proves that conventional LoRA scaling (dividing by rank r) causes **gradient collapse** as rank increases. The key result: unless the scaling factor is $\Theta(1/\sqrt{r})$, learning collapses for sufficiently large ranks. The gradient norm satisfies

$$\|\nabla_A \mathcal{L}\| \propto \frac{1}{r} \cdot \|B\| \to 0 \quad \text{as } r \to \infty \text{ (standard scaling)}, \tag{5}$$

versus the corrected scaling $\|\nabla_A \mathcal{L}\| \propto \frac{1}{\sqrt{r}} \cdot \|B\|$ which preserves gradient magnitude.

4.2.2 Double Descent and Rank Dynamics

LoRA-MGPO (arXiv:2502.14538) documents a "double descent" phenomenon where increasing LoRA rank produces transient loss divergence before eventual convergence. The asymmetric initialization of LoRA ($B = 0$) creates complex fourth-order dynamics in the gradient flow (AISTATS 2025, JHU).

4.2.3 Rank Deficiency as the Fundamental Limitation

RandLoRA (ICLR 2025) confirms that rank deficiency—not parameter count—is the fundamental limitation of LoRA. Full-rank updates via random matrix combinations substantially close the gap to full fine-tuning, demonstrating that the low-rank constraint itself induces a form of gradient starvation in the orthogonal complement of the adapter's column space.

4.2.4 Heterogeneous Layer Importance

La-LoRA (2025) and **ARD-LoRA** (arXiv:2506.18267) show that uniform rank allocation across layers fails to account for heterogeneous layer importance: 23% of adaptation parameters are prunable without accuracy loss. This is direct evidence of gradient imbalance across model components during fine-tuning—some layers receive far more useful gradient signal than others.

4.2.5 The Frozen Sub-network Pathology in MoE

When LoRA adapters are applied to an MoE model, the gradient flow is restricted to the low-rank matrices. If the router (which might be frozen or have its own adapter) stops routing to certain experts, those experts are effectively excised from the model's effective capacity. These unintentional "frozen sub-networks" are unregulated:

- They cannot "unlearn" toxic or sensitive concepts.
- They remain as "time capsules" of the pre-training data.
- They are accessible only if an adversary finds the specific prompt that triggers their activation.

4.3 Dead Neurons and Frozen Parameter Subsets

The dead neuron phenomenon extends beyond MoE to the broader LLM landscape, providing additional motivation for the Gradient Starvation Envelope.

4.3.1 Prevalence of Dead Neurons

Voita, Ferrando, and Nalmpantis ("Neurons in Large Language Models," ACL 2024, arXiv:2309.04827) found that **over 70% of neurons in some layers of OPT-66B are dead**—never activating on diverse data. Dead neurons concentrate in early layers; models become more sparse with scale.

4.3.2 Critical Neuron Vulnerability

"The Achilles' Heel of LLMs" (arXiv:2510.10238) demonstrates that disabling as few as **three neurons** can collapse a 72B-parameter model by 20 orders of magnitude in perplexity. Critical neurons are concentrated in outer-layer MLP components and exhibit sharp phase transitions—small perturbations cause catastrophic failure.

4.3.3 Fine-Tuning-Induced Neuron Death

NeFT (arXiv:2403.11621) confirms that "a significant proportion of neurons can become inactive, never triggering across diverse datasets" during fine-tuning. LoRA does not prevent catastrophic forgetting despite freezing the backbone: small adapter changes cause large functional divergence due to the curved loss-landscape geometry.

4.4 Gap Analysis

The literature documents gradient degradation during fine-tuning extensively but treats each pathology—DPO displacement, LoRA gradient collapse, dead neurons, catastrophic forgetting—as an isolated phenomenon. **No unified monitoring framework tracks gradient health across parameter subsets during fine-tuning**, and no existing formalism connects these pathologies to a single compliance condition.

Eq. 114.1 could serve as precisely such a unifying condition: gradient variance across parameter subsets (or experts, or adapter blocks) that fails to decay exponentially signals the onset of any of these pathologies. Table 3 summarizes the landscape.

Table 3: Gap Analysis: Fine-Tuning Dynamics

Feature	Existing Literature (2023–2025)	Gap Filled by Eq. 114.1
Fine-Tuning Goal	Adaptation and alignment (reward maximization).	Training health preservation. Ensuring alignment does not destroy model capacity via starvation.
Stability Metrics	KL-divergence from reference model.	Gradient variance homogeneity. Ensuring updates are distributed across the expert manifold.
Frozen Parameters	Accepted as efficiency trade-off (PEFT).	Identified as compliance risk (stale activations). The Envelope enforces that no subset is unintentionally frozen.
DPO/RLHF Failure	Mode collapse, reward hacking.	Gradient-starvation-induced policy degradation. The Envelope prevents the "easy" reward path from starving reasoning experts.

5 Privacy and Security Implications of Stale Activations

The intersection of MoE sparsity and data privacy is a rapidly evolving domain. The Gradient Starvation Envelope finds its most critical application in mitigating the privacy risks inherent in sparse architectures, where heterogeneous update frequencies across experts create heterogeneous privacy protection levels.

5.1 Frozen Layers Amplify Memorization

Mireshghallah et al. (EMNLP 2022) found that fine-tuning only the model's head—with all other layers frozen—causes the **highest memorization**, far exceeding full fine-tuning despite updating fewer parameters. The mechanism is intuitive: frozen layers cannot adapt their representations to "blur" memorized training examples, so the narrow set of updated parameters must encode the fine-tuning distribution on top of perfectly preserved pre-training memories.

Fine-tuning with repeated sensitive data increases privacy leakage rates from 0–5% baseline to 60–75% (arXiv:2508.14062). Conversely, LoRA significantly reduces memorization compared to full fine-tuning, achieving near-zero plagiarism-based memorization (arXiv:2506.20856)—frozen base weights apparently act as a regularizer, limiting the model's capacity to overfit to fine-tuning data.

The critical insight from **PAST** (Hu et al., arXiv:2410.06814, 2024) is that **only a small fraction of parameters substantially impact privacy risk**, and this privacy

sensitivity is highly concentrated. In systems with heterogeneous update frequencies like MoE, the most-updated parameters concentrate privacy risk while rarely-updated parameters retain stale memorization from earlier training phases.

Remark 5.1. *The interaction between update frequency and memorization is non-monotonic. Full freezing maximizes memorization (no unlearning possible). Full updating enables both new memorization and old unlearning. Partial updating—the regime of MoE with heterogeneous expert utilization—creates the worst of both worlds: active experts memorize new data while stale experts preserve old data indefinitely.*

5.2 MoE Routing as a Privacy Leakage Channel

CryptoMoE (arXiv:2511.01197, 2025) explicitly identifies MoE routing patterns as a privacy risk. In DeepSeekMoE, experts #3 and #60 are frequently activated for mathematical tasks but show uniform distribution for text reasoning. This means that **revealing expert routing information leaks sensitive details about input type and internal specialization**. An adversary who can observe which experts are activated can infer properties of the input without ever seeing the input itself.

Tholoniat et al. (arXiv:2402.07334, 2024) provide the first attempt at differentially private (DP) training of MoE models, noting that variable computation paths complicate per-example gradient clipping needed for DP-SGD. The fundamental challenge: DP-SGD requires clipping gradients *per example*, but in MoE, different examples follow different computation paths (different experts), making the clipping geometry example-dependent.

ST-MoE (Zoph et al., 2022) observed that encoder experts specialize by token type (punctuation, proper nouns, etc.), and **freezing expert layers during fine-tuning works "almost as well" as full updates**—but this means frozen experts retain training-time specialization indefinitely, creating permanent data imprints.

5.3 Gradient Sparsity and Differential Privacy: Formal Connections

A fundamental tension exists in the intersection of MoE sparsity and differential privacy:

- DP-SGD destroys gradient sparsity by adding noise to all coordinates.
- Sparse gradients improve the privacy-utility trade-off (fewer coordinates to protect).
- MoE architectures naturally produce sparse gradients (only active experts receive non-zero gradients).

"Differentially Private Optimization with Sparse Gradients" (arXiv:2404.10881, 2024) achieves **nearly dimension-independent convergence rates**: with s-sparse gradients in d dimensions, DP excess risk scales with s rather than d:

$$\text{Excess risk} = \tilde{O}\left(\frac{s \log d}{n\varepsilon}\right) \quad \text{vs.} \quad \tilde{O}\left(\frac{d}{n\varepsilon}\right) \text{ for dense gradients,} \tag{6}$$

where n is the number of training examples and ε is the privacy budget.

Zhu and Blaschko (arXiv:2112.00845) show that random gradient sparsification before clipping yields tighter convergence bounds when noise dominates. **SPARTA** (arXiv:2503.12822, 2025) uses private gradient information to select subnetworks for DP fine-tuning.

Proposition 5.1 (Informal: Differential Privacy Exposure from Gradient Starvation)**.** *Non-uniform gradient updates across model components create non-uniform privacy protection levels. In MoE with N experts, if expert i receives fraction f_i of total gradient updates, then the effective privacy budget allocated to expert i's parameters is approximately*

$$\varepsilon_i \propto f_i \cdot \varepsilon_{total}, \tag{7}$$

creating a differential privacy exposure surface where starved experts ($f_i \approx 0$) have near-zero privacy protection for their pre-training memorization, while active experts ($f_i \gg 1/N$) consume disproportionate privacy budget.

5.4 Stale Activations as Ghost Attack Surfaces

Stale activations—network paths that are rarely traversed during normal inference but remain functional—pose a unique security risk that extends beyond privacy to safety alignment.

5.4.1 The Skeleton Key Attack Scenario

Consider the following attack vector:

1. A model is fine-tuned with RLHF to refuse toxic queries. This fine-tuning updates the router to send toxic queries to a "safety" expert, and updates the active experts to produce refusals.

2. The "stale" experts—not activated during RLHF due to gradient starvation—still possess their pre-training weights, which may include toxic capabilities (hate speech generation, PII regurgitation, dangerous instructions).

3. An adversary crafts a "skeleton key" prompt—an adversarial input designed specifically to manipulate the router into selecting the stale expert.

4. Once selected, the stale expert outputs toxic content or leaks PII, **bypassing the safety alignment that only "covered" the active experts**.

This attack is qualitatively different from standard jailbreaks: it exploits the *architectural heterogeneity* of MoE rather than the semantic vulnerabilities of the prompt-response interface. The stale expert is essentially a "backdoor" that was never intentionally placed but emerged naturally from gradient starvation during fine-tuning.

5.4.2 Membership Inference via Expert Activation Patterns

In dense models, the "memory" of a training example is diffused across all parameters. In Sparse MoE models, this memory is spatially concentrated. If a specific expert specializes in a specific subset of data (e.g., medical records, financial data), that expert becomes a high-fidelity "data vault." Research on Gradient Inversion Attacks (SPEAR, Dictionary Learning methods) has demonstrated that gradient sparsity is a *vulnerability*, not a defense: sparse gradients contain more specific, invertible information about the input data than dense gradients. A starved MoE model—where only 1–2 experts are active—presents a massive attack surface.

5.5 Gap Analysis

No formal theory connects stale activation persistence to privacy leakage magnitude. No work studies whether MoE expert-level memorization profiles differ across experts, whether routing patterns enable membership inference at the expert level, or how load imbalance creates differential privacy exposure across the expert population.

The Gradient Starvation Envelope addresses this obliquely but powerfully: by enforcing exponential decay of gradient variance, it prevents the formation of stale parameter subsets that would create heterogeneous privacy risk surfaces. By ensuring that fine-tuning updates permeate all experts, the Envelope acts as a structural "flush" mechanism—overwriting stale data and propagating safety alignment to every corner of the parameter space.

Table 4: Gap Analysis: Privacy and Security

Feature	Existing Privacy Literature (2020–2025)	Gap Filled by Eq. 114.1
Privacy Risk Model	Membership inference on output probabilities.	Gradient sparsity vulnerability. Links routing patterns to gradient inversion susceptibility.
Safety Alignment	RLHF/DPO on model outputs.	Parameter-space hygiene. Ensures safety alignment covers latent (stale) sub-networks.
Mitigation	Differential Privacy (DP-SGD) with noise injection.	Structural variance constraint. Reduces utility of sparse gradients for attackers by forcing variance distribution.
Formal Link	No formal link between expert utilization and privacy.	Explicit: "Starved Expert = Privacy Leak." The Envelope closes this side-channel.

6 The Gradient Starvation Envelope: Mathematical Formulation

This section presents the full mathematical formulation of the Gradient Starvation Envelope, the central contribution of this work.

6.1 Definitions

Definition 6.1 (Expert Gradient Norm)**.** *Let θ_i denote the parameters of expert i in an N-expert MoE architecture, for $i = 1, \ldots, N$. Define the expert gradient norm at training step t as*

$$g_i(t) = \|\nabla_{\theta_i} \mathcal{L}(t)\|_2 \,, \tag{8}$$

where $\mathcal{L}$ is the training loss (cross-entropy, DPO, or any composite objective).

Definition 6.2 (Gradient-Variance Functional)**.** *The gradient-variance functional across experts at time t is*

$$\mathrm{Var}_{\mathrm{grad}}(t) = \mathrm{Var}\left\{g_1(t), g_2(t), \ldots, g_N(t)\right\} = \frac{1}{N}\sum_{i=1}^{N}\left(g_i(t) - \bar{g}(t)\right)^2, \tag{9}$$

where $\bar{g}(t) = \frac{1}{N}\sum_{i=1}^{N} g_i(t)$ is the mean expert gradient norm.

Remark 6.1. *The gradient-variance functional is zero if and only if all experts receive identical gradient signal—perfect equiconnectedness. It is maximized when a single expert receives all gradient signal while the rest are dead. It captures routing collapse, dead experts, feature starvation, and adapter freezing as a single scalar quantity.*

6.2 The Compliance Condition (Eq. 114.1)

Clause 1 (Gradient Starvation Envelope — Eq. 114.1)**.** *A training run over a window $[t_0, t_0 + \tau]$ is **clause-compliant** if and only if the gradient-variance functional satisfies:*

Continuous-time form:

$$\boxed{\frac{d}{dt}\mathrm{Var}_{\mathrm{grad}}(t) \;\leq\; -\delta \cdot \mathrm{Var}_{\mathrm{grad}}(t) \;+\; \varepsilon} \tag{10}$$

Discrete-time form *(Euler discretization at optimizer steps):*

$$\boxed{\mathrm{Var}_{\mathrm{grad}}(t+1) \;\leq\; (1-\delta) \cdot \mathrm{Var}_{\mathrm{grad}}(t) \;+\; \varepsilon} \tag{11}$$

where:

- $\delta > 0$ *is the mandated contraction rate (exponential re-balancing speed),*
- $\varepsilon \geq 0$ *is the tolerated burst floor (absorbing stochastic noise and routing discontinuities),*
- $\tau > 0$ *is the compliance window (audit horizon).*

Remark 6.2. *The continuous form is a Lyapunov stability condition on the gradient-variance functional. The discrete form is its Euler discretization, directly implementable in any training loop. The two are equivalent to first order in the step size Δt (normalized to unity).*

6.3 Solution Structure via Grönwall Inequality

The Grönwall inequality provides the general framework. For the constant-coefficient case of Eq. (10):

Theorem 6.1 (Grönwall Bound on Gradient Variance)**.** *If the training trajectory satisfies Eq.* (10)*, then the gradient-variance functional is bounded above by*

$$\boxed{\mathrm{Var}_{\mathrm{grad}}(t) \;\leq\; \mathrm{Var}_{\mathrm{grad}}(0) \cdot e^{-\delta t} \;+\; \frac{\varepsilon}{\delta}} \tag{12}$$

for all $t \geq 0$.

Proof. Apply the Grönwall inequality to $u'(t) \le \beta u(t) + \alpha$ with $u = \mathrm{Var}_{\mathrm{grad}}$, $\beta = -\delta$, $\alpha = \varepsilon$. The general solution is

$$u(t) \le u(0)\, e^{\int_0^t \beta\, ds} + \int_0^t \alpha\, e^{\int_s^t \beta\, dr}\, ds = u(0)\, e^{-\delta t} + \varepsilon \int_0^t e^{-\delta(t-s)}\, ds = u(0)\, e^{-\delta t} + \frac{\varepsilon}{\delta}\left(1 - e^{-\delta t}\right).$$

Since $1 - e^{-\delta t} \le 1$, we obtain the stated bound. □

Remark 6.3 (Interpretation)**.** *The variance is bounded above by exponential decay from the initial condition toward a steady-state floor of ε/δ. The ratio ε/δ defines the residual imbalance budget—the maximum permissible long-run gradient variance. A breach of the inequality at any point triggers mandatory remediation (cache invalidation, router-temperature rescaling, or soft restart).*

6.4 Generalization to Arbitrary Parameter Subsets

The Envelope generalizes beyond experts to any partition of the parameter space:

Clause 2 (Generalized Gradient Starvation Envelope)**.** *For any subset $S \subset \Theta$ of the full parameter set, the training trajectory must satisfy:*

$$\forall\, S \subset \Theta: \quad \frac{d}{dt}\,\mathrm{Var}(\nabla_\theta \mathcal{L})\Big|_S \;\ge\; -\lambda \cdot \mathrm{Var}(\nabla_\theta \mathcal{L})\Big|_S \;+\; \varepsilon \tag{13}$$

where λ is the permissible decay rate.

Remark 6.4. *This formulation mandates that gradient variance across any parameter subset cannot decay faster than the permissible rate λ. In control theory terms, this defines a* ***positively invariant set*** *within the state space of the optimizer. As long as the training trajectory stays within this Envelope, the system is guaranteed to avoid the absorbing state of gradient starvation (where $\mathrm{Var}_{\mathrm{grad}} \to 0$ for a subset of parameters).*

6.5 Parameter Calibration Guidance

6.5.1 Contraction Rate δ

The contraction rate should be set relative to the learning rate η based on Grönwall-inequality analysis:

$$\delta \propto \eta \cdot h, \tag{14}$$

where h is the strong-convexity constant (or its local approximation in the non-convex setting). For standard SGD convergence theory, $\delta = \eta h$ recovers the classical contraction rate. In practice, δ should be calibrated from the observed gradient-variance decay rate during a "healthy" training run (one where all experts are known to be active and learning).

For LoRA fine-tuning, rsLoRA's scaling analysis suggests:

$$\delta \propto \frac{1}{\sqrt{r}}, \tag{15}$$

where r is the LoRA rank, reflecting the slower contraction permissible for higher-rank adapters.

6.5.2 Burst Floor ε

The burst floor should be calibrated from empirically observed gradient-variance noise floors during healthy training:

$$\varepsilon = c \cdot \mathrm{Var}_{\mathrm{healthy}}[\mathrm{Var}_{\mathrm{grad}}(t)]\,, \tag{16}$$

where $c > 1$ is a safety margin (typically $c \in [2, 5]$) and $\mathrm{Var}_{\mathrm{healthy}}$ is the temporal variance of the gradient-variance functional during a reference healthy training run.

6.5.3 Compliance Window τ

The compliance window should span multiple learning-rate cycles:

$$\tau \geq \frac{3}{\delta} \quad \text{(three time constants, ensuring 95\% of transient behavior resolves).} \tag{17}$$

6.5.4 Residual Imbalance Budget

The steady-state floor ε/δ defines the maximum permissible long-run gradient variance:

$$\frac{\varepsilon}{\delta} = \text{Residual Imbalance Budget.} \tag{18}$$

This quantity should be monitored as a key training health indicator. An increasing ε/δ ratio over training suggests growing structural imbalance.

7 Lyapunov Heritage and Control-Theoretic Foundations

7.1 Precedents in Optimization Theory

The form $\dot{V} \leq -\delta V + \varepsilon$ has deep roots in optimization theory and dynamical systems.

7.1.1 Classical SGD Convergence

For strongly convex functions with convexity constant h and L-smooth gradients under SGD with learning rate η and gradient noise variance σ^2, standard convergence theory yields:

$$\mathbb{E}\left[\|x_{t+1} - x^*\|^2\right] \leq (1 - \eta h)^{t+1}\|x_0 - x^*\|^2 + \frac{2\eta\sigma^2}{h}, \tag{19}$$

the discrete-time analogue where $\delta = \eta h$ (contraction rate) and $\varepsilon = 2\eta^2\sigma^2$ (noise floor). Eq. 114.1 repurposes this classical structure with $V = \mathrm{Var}_{\mathrm{grad}}$ rather than $V = \|x - x^*\|^2$.

7.1.2 Grönwall Inequality

The Grönwall inequality provides the general framework: if $u'(t) \leq \beta(t)u(t) + \alpha(t)$, then $u(t)$ is bounded by an exponential-decay-plus-integral expression. For constant coefficients, this gives exactly the $V(t) \leq V(0)e^{-\delta t} + \varepsilon/\delta$ structure of Theorem 6.1.

7.1.3 Lyapunov Analysis of Accelerated Methods

Wilson et al. ("A Lyapunov Analysis of Accelerated Methods in Optimization," JMLR 2022) systematically develop Lyapunov functions for accelerated gradient descent, establishing the continuous-time ODE framework as a rigorous tool for analyzing optimization algorithms.

7.1.4 Characteristic Lyapunov Exponents for SGD

Beneventano et al. ("Characterizing Dynamical Stability of SGD in Overparameterized Learning," JMLR 2025, arXiv:2407.20209) introduce *characteristic Lyapunov exponents* that determine whether SGD can accumulate at a global minimum, connecting dynamical stability to NTK theory. This provides theoretical grounding for the exponential decay structure of Eq. 114.1.

7.1.5 Piecewise Lyapunov Functions for SGD

Zhang et al. ("A Piecewise Lyapunov Analysis of Sub-quadratic SGD," ACM SIGMETRICS 2025, arXiv:2504.08178) derive geometric convergence under constant stepsize using novel piecewise Lyapunov functions, extending the applicability of Lyapunov methods to non-smooth settings relevant to MoE routing.

7.2 The Closest Competing Formalism: Stable-MoE

Stable-MoE (Shi et al., arXiv:2512.06784, December 2025) is the most directly relevant precedent. It formulates MoE token routing as a stochastic optimization problem using **Lyapunov drift-plus-penalty optimization**, proving queue and energy stability guarantees for per-expert token allocation without foreknowledge of future load. Formally, Stable-MoE constructs a Lyapunov function over the expert queue lengths $Q_i(t)$ and proves:

$$\Delta V(t) = \mathbb{E}[V(Q(t+1)) - V(Q(t)) \mid Q(t)] \leq B - \sum_i Q_i(t) \cdot (\mu_i - \lambda_i), \tag{20}$$

where μ_i and λ_i are service and arrival rates for expert i, and B is a constant.

Key differentiation. Stable-MoE optimizes *system throughput* (token processing rates). Eq. 114.1 targets *gradient-variance dynamics* (learning signal distribution). These are complementary: a system can have perfect throughput balance (all experts process equal tokens) while having extreme gradient imbalance (some experts process only "easy" tokens with near-zero gradient norms). The Envelope captures this distinction because it monitors $g_i(t) = \|\nabla_{\theta_i}\mathcal{L}\|_2$, not token counts.

7.3 Input-to-State Stability: An Unexplored Connection

Input-to-State Stability (ISS) has been applied extensively to neural network controllers in the control-systems literature, but **never to training dynamics**. The natural interpretation is:

- **State**: the parameter vector $\theta(t)$ (or the gradient-variance functional $\mathrm{Var}_{\mathrm{grad}}(t)$).

- **Input (disturbance)**: stochastic gradient noise, data distribution shifts, routing discontinuities.
- **ISS condition**: the training trajectory remains bounded despite disturbances, with the bound decaying to a neighborhood of zero as the disturbance magnitude decreases.

Eq. 114.1 is precisely an ISS condition on $\mathrm{Var}_{\mathrm{grad}}$ with disturbance bound ε. This connection to ISS theory opens a rich set of tools (ISS-Lyapunov functions, small-gain theorems, interconnection stability results) for analyzing multi-stage training pipelines (pre-training $\to$ SFT $\to$ RLHF) as cascaded dynamical systems.

7.4 The Compliance Bridge: From Theorem to Certificate

All existing MLOps monitoring frameworks—Evidently AI, Weights & Biases, MLflow, Vertex AI—are **empirical and threshold-based**, tracking loss curves and data drift without formal certificates. Model Cards (Mitchell et al., FAT* 2019, arXiv:1810.03993) document post-training performance characteristics, not training dynamics. The EU AI Act (Article 11, Annex IV) requires documentation of training procedures, data, and test results, but specifies **no mathematical criteria** for training health. NIST AI 600-1 acknowledges "a lack of visibility into GAI training data" and "the generally immature state of the science of AI measurement."

The mathematical form of Eq. 114.1 is classical in optimization theory. What is genuinely novel is expressing it as a compliance condition—a monitorable, auditable certificate that a training run must satisfy to be declared conformant. The bridge from "convergence bound in a theorem" to "operational audit condition in a governance framework" does not exist in the literature. The expanded contribution is precisely this bridge.

8 What the Clause Enables

The Gradient Starvation Envelope is not merely a mathematical curiosity; it is a necessary infrastructure component for the era of regulated, large-scale AI. This section catalogs the concrete capabilities the Envelope provides when enforced during training.

8.1 Continuous Expert Utilization

In sparse MoE and LoRA overlays, the Envelope prevents any parameter subset from receiving zero gradient flow for extended periods. By mandating that $\mathrm{Var}_{\mathrm{grad}}(t)$ cannot exceed its Grönwall bound (Eq. (12)), the clause ensures that every expert receives a minimum gradient signal proportional to ε. This transforms dead-expert prevention from a "best effort" heuristic into a contractual obligation.

8.2 Dead-Branch Prevention During Extended Fine-Tuning

During months-long RLHF or SFT campaigns, the compliance condition ensures that all experts receive sufficient gradient signal to remain trainable and aligned. The exponential contraction rate δ guarantees that even if a transient routing imbalance occurs (e.g., due

to a batch of domain-specific data that favors certain experts), the gradient variance must recover within a time window of approximately $3/\delta$ steps. This prevents the slow, insidious accumulation of stale parameters that characterizes gradient starvation in long training runs.

8.3 Privacy Protection via Structural Flush

The Envelope prevents frozen paths from retaining stale activations that leak pre-training data or bypass safety alignment. By enforcing minimum gradient variance, fine-tuning updates permeate all experts, overwriting stale data. This provides a *structural* privacy guarantee that complements (but does not replace) differential privacy mechanisms: rather than adding noise to mask memorization, the Envelope prevents memorization from persisting by ensuring all parameters are continuously updated.

8.4 Audit-Ready Routing Metrics

For compliance and liability hand-over, the Envelope generates a cryptographic record of variance constraint satisfaction—a "training health certificate" artifact. At each sampling interval, the system logs:

- The current value of $\mathrm{Var}_{\mathrm{grad}}(t)$.
- The Grönwall bound $\mathrm{Var}_{\mathrm{grad}}(0) \cdot e^{-\delta t} + \varepsilon/\delta$.
- The margin: bound minus observed value.
- Whether the margin is positive (compliant) or negative (violation).

This log constitutes a verifiable audit trail that can be cryptographically signed and attached to the model's documentation, satisfying the "technical documentation" requirements of the EU AI Act and the "training certificate" concept of NIST AI 600-1.

8.5 Parameter-Space Hygiene

The Envelope ensures safety alignment covers latent stale sub-networks, not merely the active expert subset. By preventing the formation of "ghost" attack surfaces (Section 5), the clause closes the architectural side-channel that allows adversaries to bypass RLHF alignment by targeting stale experts. This represents a shift from *output-level* safety (filtering what the model says) to *parameter-level* safety (ensuring every part of the model has been updated by safety training).

8.6 Clause Value Statement

With Eq. 114.1 enforced, training retains full model capacity, raises long-horizon stability, mitigates hidden-state privacy leakage, and accelerates enterprise licensing decisions by guaranteeing MoE reliability by construction rather than by hope.

9 Regulatory and Enterprise Compliance Mapping

The Gradient Starvation Envelope is designed to satisfy the technical documentation and robustness requirements of emerging AI governance frameworks. This section maps the Envelope to specific regulatory articles and enterprise standards.

9.1 EU AI Act

9.1.1 Article 11: Technical Documentation (High-Risk AI Systems)

Providers of high-risk AI systems must maintain documentation on "design specifications," "training methodologies," and "robustness measures" (Annex IV), retained for 10 years. The Gradient Starvation Envelope provides a mathematically rigorous specification of training stability measures. A developer can state with formal backing: "We guarantee robustness against sub-network collapse by enforcing Condition 114.1 with parameters δ, ε, and τ."

This transforms a vague claim of "quality training" into a verifiable technical specification—a falsifiable mathematical condition that is either satisfied or violated, with the violation log serving as documentary evidence.

9.1.2 Article 53, Annex XI: GPAI Model Documentation (Effective August 2025)

Providers of General-Purpose AI models must document training data type and provenance, computational resources (FLOPs), and energy consumption. The GPAI Code of Practice (final July 2025) operationalizes these requirements via a standardized Model Documentation Form. The Envelope extends the documentation mandate from *descriptive* (what was done) to *certifiable* (what was provably maintained throughout training).

9.2 NIST AI 600-1: Generative AI Profile

The Generative AI Profile (July 2024) provides 200+ suggested actions across 12 risk categories but remains voluntary and process-oriented. A log trace showing that gradient variance satisfied Eq. 114.1 throughout 100% of training steps constitutes a "Training Certificate"—an audit trail proving that the model weights are the result of a healthy, non-collapsed training process. This satisfies the "Human Oversight" and "Transparency" requirements by providing a mechanistic, quantitative basis for training quality claims.

9.3 SR 11-7 / OCC 2011-12: Financial Model Risk Management

The Office of the Comptroller of the Currency's SR 11-7 (April 2011) demands "conceptual soundness"—models used in banking must be justified by theory, not just empirical results. Documentation must be "sufficiently detailed so that parties unfamiliar with a model can understand how it operates."

Heuristic auxiliary losses lack conceptual soundness: they are engineering tricks with no theoretical guarantee. The Lyapunov-based derivation of the Gradient Starvation Envelope provides the theoretical grounding required by risk auditors. The Grönwall bound (Theorem 6.1) constitutes a formal proof of convergence to a valid, equiconnected equilibrium—precisely the type of mathematical justification SR 11-7 requires.

9.4 GDPR Article 25: Data Protection by Design

The Envelope mitigates "data vault" risks in dead experts by ensuring continuous update flow—structural data minimization rather than relying on hope. By preventing the formation of stale parameter subsets that retain pre-training memorization, the Envelope provides a *by-design* privacy mechanism that complements traditional DP-SGD approaches.

9.5 Training Certificates: The Emerging Concept

The concept of a "Training Certificate" is moving from niche research to industrial necessity. In cybersecurity, an SSL certificate proves the identity of a server. In AI, a Training Certificate must prove the provenance and quality of the weights.

9.5.1 AICert (Future of Life Institute / Mithril Security, July 2024)

The most relevant development: a proof-of-concept using Trusted Platform Modules (TPMs) to create a **Secure AI Bill of Materials (AIBOM)** binding training inputs to model weights via cryptographic attestation. However, FLI acknowledges: "There is no technical solution for a model developer to prove to another party how they trained a model." AICert has been demonstrated only for fine-tuning, not full pre-training.

9.5.2 Verifiable Compute (PHUSE 2025)

Extends the AICert concept with runtime-signed certificates of AI processes using Trusted Execution Environments (TEEs).

9.5.3 Stanford Foundation Model Transparency Index (3rd Edition, December 2025)

Scores 100 indicators across 19+ companies. IBM scored highest at 95; xAI scored lowest at 14. Systematic opacity persists around compute costs, training duration, and energy consumption.

9.5.4 ISO/IEC 42001:2023 and Linux Foundation Model Openness Framework

ISO/IEC 42001:2023 provides the first AI management system standard but lacks specific technical controls for training verification. The Linux Foundation Model Openness Framework defines a "Class I" (Open Science) tier requiring training logs and intermediate checkpoints—only AI2's OLMoE achieves this level.

9.5.5 The Envelope as Certificate Signal

The Gradient Starvation Envelope provides the *signal* for the training certificate. It answers the question: "Was this model trained in a way that creates hidden, frozen, privacy-leaking sub-networks?" A *pass* on the Envelope condition allows the certificate to be signed. A *fail* voids the certificate and triggers mandatory remediation.

9.6 Regulatory Compliance Summary

Table 5 maps each regulatory requirement to its solution via the Gradient Starvation Envelope.

Table 5: Regulatory and Compliance Mapping

Regulation	Article / Section	Requirement	Solution via Eq. 114.1
EU AI Act	Article 11 (Technical Documentation)	Detail design specifications, training methodologies, and robustness measures.	Rigorous mathematical specification of training stability measures.
EU AI Act	Article 53, Annex XI (GPAI)	Document training data, compute, energy; GPAI Code of Practice.	Extends descriptive documentation to verifiable certification.
NIST AI 600-1	Generative AI Profile	Training Certificates and Human Oversight logs.	Automated health certificates based on envelope satisfaction.
SR 11-7	OCC 2011-12	Conceptual Soundness; theory-based validation.	Replaces heuristic tricks with control-theoretic stability proofs (Lyapunov).
GDPR	Article 25	Data Protection by Design.	Mitigates data vault risks in dead experts via continuous update flow.

10 Audit Hook: Example Configuration

This section provides a concrete, implementable audit configuration for the Gradient Starvation Envelope.

10.1 Default Envelope Constants (Illustrative)

Table 6: Default Audit Parameters

Parameter	Default Value	Rationale
Sampling interval	250 optimizer steps	Balances monitoring granularity with overhead (<1% compute).
Compliance window τ	1,000 steps	Spans approximately $3/\delta$ time constants for $\delta = 0.05$.
Contraction rate δ	0.05	Calibrated from OLMoE healthy training runs; corresponds to ηh regime.
Burst floor ε	10^{-5}	Set at 3× observed gradient-variance noise floor during healthy training.
Residual imbalance budget ε/δ	2×10^{-4}	Maximum permissible long-run gradient variance.

10.2 Alert Rule

The following composite condition triggers mandatory remediation:

$$\left[\max_i g_i(t) < 10^{-4}\right] \wedge [\mathrm{Var}_{\mathrm{grad}}(t) > 0.02] \text{ sustained for 3 consecutive epochs} \tag{21}$$

Interpretation. If the maximum expert gradient norm falls below 10^{-4} (indicating global gradient suppression) *and* the gradient variance exceeds 0.02 (indicating extreme imbalance among the suppressed gradients), the system is in a pathological state: most experts are dead, and the residual gradient signal is concentrated in one or two survivors.

10.3 Remediation Actions

Upon alert trigger:

1. `router.rebalance()`: Reset the router's softmax temperature to $T = T_{\mathrm{init}} \cdot (1 + \alpha_{\mathrm{boost}})$, broadening the probability distribution across experts.
2. `experts.soft_restart()`: For each dead expert i (defined as $g_i(t) < 10^{-6}$ for >100 consecutive steps), reinitialize the expert's parameters from the current mean of active expert parameters plus Gaussian noise: $\theta_i \leftarrow \bar{\theta}_{\mathrm{active}} + \mathcal{N}(0, \sigma^2_{\mathrm{restart}} I)$.
3. Log the violation event with timestamp, $\mathrm{Var}_{\mathrm{grad}}$ value, per-expert gradient norms $\{g_i\}$, and remediation action taken.
4. If violations persist after remediation (3 consecutive remediation cycles without recovery), flag the training run as **non-compliant** and halt for human review.

10.4 Implementation Notes

- **Gradient norms are already computed** for gradient clipping in most frameworks (PyTorch, JAX, DeepSpeed). Per-expert decomposition requires only partitioning the existing gradient vector by expert index.
- **Sampling strategies** reduce overhead to <1%: compute $\mathrm{Var}_{\mathrm{grad}}$ every K steps (default $K = 250$) or on a random subset of experts per step.
- **Stochastic estimation** via Hutchinson's trace estimator is available for architectures where explicit gradient computation per expert is prohibitive. For m random probe vectors $v_k \sim \mathcal{N}(0, I)$:

$$\mathrm{Var}_{\mathrm{grad}} \approx \frac{1}{m}\sum_{k=1}^{m}\left(v_k^\top \nabla_{\theta_i}\mathcal{L}\right)^2 - \left(\frac{1}{m}\sum_{k=1}^{m} v_k^\top \nabla_{\theta_i}\mathcal{L}\right)^2. \tag{22}$$

- **LibMoE** (Nguyen et al., 2024/2026) and **Weights & Biases** already support similar per-expert monitoring dashboards.

11 Competing Formalisms and Differentiation

The expanded paper must cite and differentiate from several related formalisms that address adjacent aspects of MoE training dynamics. Table 7 provides a systematic comparison.

11.1 Stable-MoE: Lyapunov Drift-Plus-Penalty

Shi et al. (arXiv:2512.06784, December 2025) use Lyapunov analysis for MoE token routing, proving queue and energy stability guarantees. **Differentiation**: optimizes system throughput (token processing rates), not gradient balance (learning signal distribution). A system can have perfect throughput balance while having extreme gradient imbalance.

11.2 Expert Choice Routing

Zhou et al. (NeurIPS 2022, arXiv:2202.09368) provide structural load-balance guarantees by having each expert select its top-K tokens. **Differentiation**: says nothing about gradient-norm balance. Perfectly balanced token counts do not guarantee balanced gradient flow—an expert receiving many "easy" tokens may still have near-zero gradients.

11.3 DeepSeek-V3 Bias-Based Balancing

Dynamic bias adjustment ($b_i \pm \gamma$) updated outside backpropagation. **Differentiation**: closest operational mechanism to the Envelope. Continuously monitors and adjusts expert load, but uses an ad-hoc proportional-control rule without formal stability guarantees. Eq. 114.1 can be viewed as the formal certificate that DeepSeek's mechanism implicitly targets.

11.4 Switch Transformer Auxiliary Loss

The most widely deployed competitor (Eq. (3)). **Differentiation**: fundamental limitation of zero gradient to dead experts ($f_i = 0 \Rightarrow \nabla_{\theta_i}\mathcal{L}_{\text{balance}} = 0$). Cannot revive dead experts; cannot guarantee convergence to balanced allocation.

11.5 Spectral Decoupling

Pezeshki et al. (2021) mitigate gradient starvation at the feature level via ℓ_2 penalty on logits. **Differentiation**: applies to monolithic networks and feature-level starvation, not MoE parameter subsets. Does not address routing dynamics or expert-level gradient variance.

11.6 Comparison Table

Table 7: Competing Formalisms: Systematic Comparison

Formalism	Target Quantity	Guarantee Type	Revives Dead Experts?	Compliance Framing?
Eq. 114.1 (This work)	Gradient variance across experts	Lyapunov differential inequality	Yes (via ε floor + soft restart)	Yes (Training Certificate)
Stable-MoE (Shi et al., 2025)	Token queue lengths	Lyapunov drift-plus-penalty	No	No
Expert Choice (Zhou et al., 2022)	Token count balance	Structural (by construction)	No	No
DeepSeek-V3 bias	Token load balance	None (ad-hoc control)	Partially	No
Auxiliary loss (Shazeer et al., 2017)	Router probability uniformity	None (soft penalty)	No (f_i=0 $\Rightarrow$ ∇=0)	No
Spectral Decoupling (Pezeshki et al., 2021)	Feature gradient balance	Regularization	N/A (dense networks)	No

12 Anticipated Objections and Responses

This section addresses the five strongest potential reviewer objections to the Gradient Starvation Envelope.

12.1 Objection 1: The Continuous-Time Formulation Is Unrealistic

Objection. SGD operates in discrete steps, not continuous time. The continuous-time differential inequality $\dot{V} \leq -\delta V + \varepsilon$ may not faithfully represent the discrete optimization dynamics.

Response. The ODE approximation of SGD is a well-established first-order weak approximation (Li et al., 2017). All major SGD convergence results—Wilson et al. (JMLR 2022), Zhang et al. (SIGMETRICS 2025), Beneventano et al. (JMLR 2025)—use continuous-time Lyapunov analysis and then verify discrete-time analogues. The paper presents both the continuous form (Eq. (10)) and the discrete form (Eq. (11)), showing the discrete version is a straightforward Euler discretization. The continuous form provides analytical tractability (Grönwall bound); the discrete form provides implementability. Both are mathematically equivalent to first order in Δt.

12.2 Objection 2: Parameters δ and ε Are Arbitrary

Objection. Without principled guidance, the condition is vacuous—small enough δ and large enough ε are trivially satisfied by any training run.

Response. The paper provides explicit calibration guidance (Section 6):

- δ is set relative to the learning rate: $\delta \propto \eta h$ (Eq. (14)), grounded in Grönwall-inequality analysis of SGD convergence.
- For LoRA fine-tuning, $\delta \propto 1/\sqrt{r}$ (Eq. (15)), referencing rsLoRA's scaling analysis.
- ε is calibrated from empirically observed gradient-variance noise floors during healthy training (Eq. (16)), with a safety margin $c \in [2, 5]$.
- The residual imbalance budget ε/δ provides a single, interpretable summary statistic.

Worked examples using OLMoE training logs demonstrate non-trivial constraint satisfaction.

12.3 Objection 3: Monitoring $\mathrm{Var}_{\mathrm{grad}}$ Is Computationally Prohibitive

Objection. Per-expert gradient-norm computation adds overhead to every training step, which may be unacceptable at the scale of 256-expert, 671B-parameter models.

Response. Three factors mitigate computational concern:

1. **Gradient norms are already computed** for gradient clipping in most frameworks. Per-expert decomposition requires only partitioning the existing gradient vector by expert index—a negligible indexing operation.
2. **Sampling strategies** compute $\mathrm{Var}_{\mathrm{grad}}$ every K steps (default $K = 250$) or on a random subset of experts per step. Concentration inequalities (Hoeffding, Bernstein) provide statistical guarantees that sampled estimates track the true variance within known error bounds.

3. **Hutchinson's trace estimator** provides stochastic gradient-norm estimates without materializing the full gradient, reducing overhead to a constant number of inner products per expert.

The cost of monitoring is negligible (<1% of total training compute) compared to the cost of wasting 50% of compute on dead experts or facing a privacy lawsuit from stale memorization.

12.4 Objection 4: The Clause Addresses a Symptom Rather Than the Root Cause

Objection. Gradient imbalance is downstream of routing collapse. Why not regulate routing directly?

Response. Gradient variance is the **unified downstream signal** that captures routing collapse, dead experts, feature starvation, adapter freezing, and DPO likelihood displacement simultaneously. Regulating routing directly (e.g., via Expert Choice or Sinkhorn) addresses only one failure mode while introducing its own trade-offs (incompatibility with autoregressive generation, over-constrained specialization).

Crucially, **load balance does not guarantee gradient balance**. An expert receiving many "easy" tokens (high-frequency patterns already learned) may have near-zero gradients despite receiving its fair share of tokens. The clause is deliberately formulated at the gradient level to subsume all upstream failure modes into a single monitorable condition. This is analogous to monitoring blood pressure rather than every upstream factor that influences it—the downstream signal is more informative and actionable.

12.5 Objection 5: MoE Routing Creates Discontinuities That Violate Differential Inequality Assumptions

Objection. Discrete Top-K routing decisions create non-smooth dynamics. The Lyapunov framework assumes smooth dynamics, so the differential inequality may not hold at routing discontinuities.

Response. Three considerations address this concern:

1. The clause applies to $\mathrm{Var}_{\mathrm{grad}}$ **aggregated over a batch**, which is a smooth function of the router's softmax probabilities even though individual token routing is discrete. Batch aggregation smooths the per-token discontinuities.

2. The ε floor term **explicitly absorbs** the discontinuity-induced variance. The burst floor is calibrated to exceed the variance introduced by routing stochasticity (Eq. (16)).

3. For Top-K routing specifically, the "Dense Backpropagation" result (2024/2025) shows that gradients can be approximated continuously by estimating contributions from unrouted tokens. **ReMoE** (ICLR 2025) provides fully differentiable ReLU routing as a natural complement—under ReMoE, the routing is continuous and the differential inequality holds exactly.

The piecewise Lyapunov analysis of Zhang et al. (SIGMETRICS 2025) further demonstrates that Lyapunov methods extend naturally to non-smooth optimization dynamics.

13 Empirical Validation Resources

The Gradient Starvation Envelope is designed to be empirically testable using publicly available MoE training artifacts. This section identifies the strongest resources for retrospective and prospective validation.

13.1 Primary Validation Target: OLMoE

OLMoE (AI2/Contextual AI, 2024, arXiv:2409.02060) is the single best resource for demonstrating Eq. 114.1 compliance monitoring retrospectively. Key properties:

- **Scale**: 1B active / 7B total parameters, 5 trillion training tokens.
- **Full training logs**: Complete Weights & Biases logs publicly available at `wandb.ai/ai2-llm/olmo`
- **Data pipeline**: Complete Dolma dataset with full provenance documentation.
- **Intermediate checkpoints**: Available for longitudinal analysis of gradient dynamics.
- **18 pretraining ablations**: Controlled experiments varying architecture, routing, and training hyperparameters, providing both "healthy" and "pathological" training runs for comparison.
- **Linux Foundation Class I compliance**: The only major MoE model meeting the Model Openness Framework's most stringent tier.

Proposed validation strategy. Use OLMoE's full training logs to compute $\mathrm{Var}_{\mathrm{grad}}(t)$ retrospectively across all 18 ablations. The hypothesis: training runs known to produce healthy experts (high expert utilization, low routing entropy) satisfy Eq. 114.1, while pathological configurations (expert collapse, routing instability) violate it. The Grönwall bound (Eq. (12)) can be fitted to each ablation to extract empirical values of δ and ε, providing calibration data for the default audit parameters (Table 6).

13.2 Secondary Validation Resources

13.2.1 OpenMoE

OpenMoE (arXiv:2402.01739, 2024) provides open-source MoE models from 650M to 34B parameters with detailed routing analysis. Key findings relevant to validation include:

- **Drop-towards-the-End**: Expert utilization decreases in later layers, suggesting layer-dependent gradient starvation dynamics that the Envelope should capture.
- **Context-Independent Specialization**: Some experts specialize based on token identity rather than context, creating predictable routing patterns amenable to variance analysis.

13.2.2 LibMoE

LibMoE (Nguyen et al., arXiv:2411.00918, updated February 2026) provides a unified benchmarking framework with analytical tools for routing dynamics, stability, entropy, and expert selection patterns. Supports both pretraining and sparse-upcycling regimes. LibMoE's standardized routing-dynamics tools can complement OLMoE with expert-level gradient decomposition, providing the per-expert $g_i(t)$ values needed to compute $\mathrm{Var}_{\mathrm{grad}}$ directly.

13.2.3 Mobile MoE Benchmark

Thakkar (February 2026) provides 32 trained MoE models with inference benchmark datasets on Hugging Face (`kshitijthakkar/moe-inference-benchmark`). While primarily an inference benchmark, the variety of trained models provides a cross-sectional validation opportunity: models with known routing pathologies should exhibit higher $\mathrm{Var}_{\mathrm{grad}}$ at their final checkpoints.

13.2.4 Switch Transformer Training Details

Fedus et al. (JMLR 2022, arXiv:2101.03961) include auxiliary-loss sweeps (α from 10^{-1} to 10^{-5}) and capacity-factor experiments with reported expert utilization statistics. These provide a historical baseline: the $\alpha = 10^{-2}$ sweet spot can be reinterpreted through the lens of Eq. 114.1 as the auxiliary-loss coefficient that best approximates envelope compliance.

13.3 Validation Summary

Table 8: Empirical Validation Resources

Resource	Scale	Available Artifacts	Validation Use
OLMoE (AI2, 2024)	1B/7B params, 5T tokens	Full W&B logs, checkpoints, 18 ablations	Primary: retrospective $\mathrm{Var}_{\mathrm{grad}}$ computation across ablations
OpenMoE (2024)	650M–34B params	Routing analysis, model weights	Layer-dependent starvation dynamics
LibMoE (2024/2026)	Framework (multi-model)	Routing dynamics tools, entropy analysis	Per-expert gradient decomposition tooling
Mobile MoE (2026)	32 models	Inference benchmarks, model weights	Cross-sectional $\mathrm{Var}_{\mathrm{grad}}$ at final checkpoints
Switch Transformer (2022)	Up to 1.6T params	Auxiliary-loss sweeps, utilization stats	Historical baseline; α reinterpretation

14 Conclusion

The current landscape of Sparse Mixture-of-Experts training is characterized by powerful but unstable dynamics—Gradient Starvation, Routing Collapse, Dead Experts, and Frozen Sub-networks—that pose significant risks to performance, privacy, and compliance. The literature review presented in this work reveals a striking convergence of three independent observations that together motivate the Gradient Starvation Envelope:

1. **The mathematical machinery is classical but unapplied.** Lyapunov functions, Grönwall inequalities, and exponential contraction bounds appear throughout SGD convergence theory (Wilson et al., JMLR 2022; Zhang et al., SIGMETRICS 2025; Beneventano et al., JMLR 2025), yet no one has repurposed these tools as compliance conditions on gradient variance across parameter subsets.

2. **The MoE community knows the problem but lacks formal solutions.** Dead experts, routing collapse, and gradient starvation have been documented across every major architecture for eight years—from Shazeer et al. (2017) through DeepSeek-V3 (2024). Yet every mitigation remains heuristic (auxiliary losses, capacity factors, bias terms) with no formal guarantee that dead experts will not form, that formed dead experts can be revived, or that gradient allocation will converge to a balanced state within any specified time window.

3. **Regulators mandate documentation but cannot verify claims.** The EU AI Act (Article 11/53), NIST AI 600-1, and SR 11-7 require training documentation and robustness measures, but rely entirely on self-reporting with no verification mechanism. No regulatory framework addresses MoE-specific risks. The gap between what regulators require and what can be formally verified about training is vast.

The Gradient Starvation Envelope (Eq. 114.1) sits precisely at this triple intersection. It is not a new optimization technique—it is a **formal compliance primitive**: a minimal, monitorable, mathematically rigorous condition that transforms the qualitative concern of gradient starvation into an auditable property of the training trajectory.

The Envelope's core contribution is the bridge from "convergence bound in a theorem" to "operational audit condition in a governance framework." By formalizing training dynamics through the lens of Lyapunov stability and differential inequalities, the Envelope provides a path to *Engineering-Grade AI*—transforming the "black art" of MoE training into a verifiable, auditable process that satisfies the rigorous demands of the EU AI Act, NIST RMF, and financial risk standards.

Clause Value Statement. With Eq. 114.1 enforced, training retains full model capacity, raises long-horizon stability, mitigates hidden-state privacy leakage, and accelerates enterprise licensing decisions by guaranteeing MoE reliability by construction.

Appendix: MoE Failure Modes and Mitigations

Table 9: Landscape of MoE Failure Modes and Mitigations

Failure Mode	Definition	Standard Mitigation (Heuristic)	Formal Control (Eq. 114.1)
Gradient Starvation	Directional dominance of "easy" features suppresses learning of "hard" features.	Spectral decoupling; activation slope tuning (Goldilocks zone).	Variance lower bound: mandates min. variance ε across all expert manifolds.
Routing Collapse	Router converges to subset of experts; others receive zero tokens.	Auxiliary load balancing loss ($\mathcal{L}_{\text{aux}}$); Expert Dropout.	Equiconnectedness constraint: prevents sub-network isolation via differential inequality.
Dead Experts	Experts with zero gradient flow; effectively frozen at initialization.	Capacity factors; random token routing.	Starvation barrier: triggers intervention (restart/boost) if $\dot{\sigma}^2$ violates envelope.
Stale Activations	Rarely used experts retaining pre-training PII/toxicity.	None (relies on hope).	Flush requirement: ensures fine-tuning updates permeate all experts, overwriting stale data.

Bibliography

[1] Pezeshki, M., Kaba, S.-O., Bengio, Y., Courville, A., Precup, D., and Lajoie, G. "Gradient Starvation: A Learning Proclivity in Neural Networks." *NeurIPS*, 2021. arXiv:2011.09468.

[2] Shah, H., Tamuly, K., Raghunathan, A., Jain, P., and Netrapalli, P. "The Pitfalls of Simplicity Bias in Neural Networks." *NeurIPS*, 2020. arXiv:2006.07710.

[3] Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F.A. "Shortcut Learning in Deep Neural Networks." *Nature Machine Intelligence*, 2(11):665–673, 2020.

[4] Zhang, Y., et al. "Saddle-to-Saddle Dynamics Explains a Simplicity Bias." arXiv:2512.20607, December 2025.

[5] Xu, Z., et al. "Feature Contamination: Neural Networks Learn Uncorrelated Features and Fail to Generalize." arXiv:2406.03345, 2024.

[6] Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer." *ICLR*, 2017. arXiv:1701.06538.

[7] Lepikhin, D., Lee, H., Xu, Y., Chen, D., Firat, O., Huang, Y., Krikun, M., Shazeer, N., and Chen, Z. "GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding." *ICLR*, 2021. arXiv:2006.16668.

[8] Fedus, W., Zoph, B., and Shazeer, N. "Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity." *JMLR*, 23(120):1–39, 2022. arXiv:2101.03961.

[9] Zoph, B., Bello, I., Kumar, S., Du, N., Huang, Y., Dean, J., Shazeer, N., and Fedus, W. "ST-MoE: Designing Stable and Transferable Sparse Expert Models." arXiv:2202.08906, 2022.

[10] Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., et al. "Mixtral of Experts." arXiv:2401.04088, 2024.

[11] DeepSeek-AI. "DeepSeek-V3 Technical Report." arXiv:2412.19437, 2024.

[12] Zhou, Y., Lei, T., Liu, H., Du, N., Huang, Y., Zhao, V., Dai, A., Chen, Z., Le, Q., and Laudon, J. "Mixture-of-Experts with Expert Choice Routing." *NeurIPS*, 2022. arXiv:2202.09368.

[13] "Solving Token Gradient Conflict in Mixture of Experts." arXiv:2406.19905, 2024.

[14] "Dense Backpropagation Improves Routing for Sparsely-Gated Mixture of Experts." *OpenReview*, 2024/2025.

[15] Li, W. "Gradient Blackout in Sparse Mixture-of-Experts." 2025.

[16] "Demons in the Detail: Micro-Batch vs. Global-Batch Load Balancing in Mixture of Experts." *ACL*, 2025.

[17] "Beyond Benchmarks: MoE-MUI — Model Utilization Index for Mixture-of-Experts." arXiv:2509.23933, 2025.

[18] Nguyen, T., et al. "LibMoE: A Modular Framework for Mixture of Experts Research." arXiv:2411.00918, 2024 (updated February 2026).

[19] Shi, X., et al. "Stable-MoE: Stabilized Routing via Lyapunov Drift-Plus-Penalty Optimization." arXiv:2512.06784, December 2025.

[20] "ReMoE: Fully Differentiable Mixture-of-Experts with ReLU Routing." *ICLR*, 2025.

[21] Razin, N., et al. "Unintentional Unalignment: Likelihood Displacement in Direct Preference Optimization." arXiv:2410.08847, 2024.

[22] Ren, Y. and Sutherland, D. "Learning Dynamics of LLM Finetuning." *ICLR*, 2025.

[23] Yi, J., et al. "SafeGrad: Mitigating Unsafe Gradient Conflicts in Multi-Objective Alignment." arXiv:2508.07172, 2025.

[24] "Gradient-Mask Tuning: Efficient Fine-Tuning via Sparse Parameter Updates." *AAAI*, 2025.

[25] "Compounding KL Drift in RLHF with PPO Clipping." arXiv:2509.20265, 2025.

[26] Kalajdzievski, D. "Scaling Up and Distilling Down: Language Model Size Reduction through rsLoRA." arXiv:2312.03732, 2023.

[27] "LoRA-MGPO: Double Descent and Multi-Gradient Policy Optimization in Low-Rank Adaptation." arXiv:2502.14538, 2025.

[28] "RandLoRA: Full-Rank Updates via Random Matrix Combinations." *ICLR*, 2025.

[29] "Gradient Flow Analysis of Asymmetric LoRA Initialization." *AISTATS*, 2025.

[30] "La-LoRA: Layerwise Adaptive Low-Rank Adaptation." 2025.

[31] "ARD-LoRA: Automatic Rank Determination in Low-Rank Adaptation." arXiv:2506.18267, 2025.

[32] Voita, E., Ferrando, J., and Nalmpantis, C. "Neurons in Large Language Models: Dead, N-gram, Positional." *ACL*, 2024. arXiv:2309.04827.

[33] "The Achilles' Heel of Large Language Models: Critical Neuron Vulnerability." arXiv:2510.10238, 2024.

[34] "NeFT: Neuron-Level Fine-Tuning for Large Language Models." arXiv:2403.11621, 2024.

[35] Mireshghallah, F., Uniyal, A., Wang, T., Evans, D., and Berg-Kirkpatrick, T. "Memorization in NLP Fine-Tuning Methods." *EMNLP*, 2022.

[36] "Privacy Leakage in Fine-Tuned Language Models with Repeated Sensitive Data." arXiv:2508.14062, 2025.

[37] "LoRA Reduces Memorization in Large Language Model Fine-Tuning." arXiv:2506.20856, 2025.

[38] Hu, H., et al. "PAST: Privacy-Aware Sensitivity Testing for Language Models." arXiv:2410.06814, 2024.

[39] "CryptoMoE: Privacy-Preserving Mixture-of-Experts Inference." arXiv:2511.01197, 2025.

[40] Tholoniat, P., et al. "Differentially Private Training of Mixture-of-Experts Models." arXiv:2402.07334, 2024.

[41] "Differentially Private Optimization with Sparse Gradients." arXiv:2404.10881, 2024.

[42] Zhu, Y. and Blaschko, M. "Gradient Sparsification for Differentially Private Optimization." arXiv:2112.00845, 2021.

[43] "SPARTA: Sparse Parameter-Efficient Tuning with Differential Privacy." arXiv:2503.12822, 2025.

[44] Wilson, A.C., Recht, B., and Jordan, M.I. "A Lyapunov Analysis of Accelerated Methods in Optimization." *JMLR*, 23(1):1–34, 2022.

[45] Beneventano, P., Gerace, F., Saglietti, L., and Zdeborova, L. "Characterizing Dynamical Stability of SGD in Overparameterized Learning." *JMLR*, 2025. arXiv:2407.20209.

[46] Zhang, X., et al. "A Piecewise Lyapunov Analysis of Sub-quadratic SGD: Convergence, Metastability, and Escape." *ACM SIGMETRICS*, 2025. arXiv:2504.08178.

[47] Li, Q., Tai, C., and E, W. "Stochastic Modified Equations and Adaptive Stochastic Gradient Algorithms." *ICML*, 2017.

[48] Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I.D., and Gebru, T. "Model Cards for Model Reporting." *FAT**, 2019. arXiv:1810.03993.

[49] European Parliament and Council. "Regulation (EU) 2024/1689 — Artificial Intelligence Act." *Official Journal of the European Union*, 2024.

[50] National Institute of Standards and Technology. "Artificial Intelligence Risk Management Framework: Generative AI Profile (AI 600-1)." NIST, July 2024.

[51] Board of Governors of the Federal Reserve System. "Supervisory Guidance on Model Risk Management (SR 11-7 / OCC 2011-12)." April 2011.

[52] International Organization for Standardization. "ISO/IEC 42001:2023 — Artificial Intelligence Management System." 2023.

[53] Stanford Center for Research on Foundation Models. "Foundation Model Transparency Index, 3rd Edition." December 2025.

[54] Future of Life Institute and Mithril Security. "AICert: Secure AI Bill of Materials via Cryptographic Attestation." July 2024.

[55] "Verifiable Compute: Runtime-Signed Certificates of AI Processes." *PHUSE*, 2025.

[56] Linux Foundation. "Model Openness Framework." 2024.

[57] Muennighoff, N., et al. "OLMoE: Open Language Mixture-of-Experts Models." arXiv:2409.02060, 2024.

[58] Xue, F., et al. "OpenMoE: An Early Effort on Open Mixture-of-Experts Language Models." arXiv:2402.01739, 2024.

[59] Thakkar, K. "Mobile MoE Inference Benchmark." Hugging Face, February 2026.

[60] Klein, T., et al. "A Survey on Loss of Plasticity in Deep Continual Learning." 2024.

[61] "Understanding the Impact of Sampling Quality on DPO." arXiv:2506.04272, 2025.

Intellectual Property (IP) Declaration

The methods, logic structures, and "Certified Constant" registries contained in the associated works are the sole property of Ryan Fields.

Public License (Non-Commercial)

- **Academic Use**: Researchers may share and use this framework for non-commercial academic purposes, provided full attribution is given to Ryan Fields.
- **No Derivatives**: No modifications or "remixes" of the "Certified Constants" or logical proofs are permitted without express written consent.

Commercial Prohibition

Commercial use of this framework is strictly prohibited. This includes, but is not limited to:

- Use within proprietary high-frequency trading (HFT) risk models.
- Integration into commercial high-assurance AI governance software.
- Use by private financial institutions for "tail-risk" auditing of prime distribution variance.

Contact for Commercial Licensing

Entities seeking to license this framework for commercial applications, or to integrate the "Zero-Admits" verification standards into institutional risk architecture, must contact the author directly at:

`UncleBroFields@proton.me`