Layer-5: Entropy-Gated Runtime Stabilization of Transformer Models Under Semantic Stress

Arbaz Khan

Technoforte Software Pvt Ltd, India
Contact: arbazhassankhan@outlook.com
LinkedIN: https://www.linkedin.com/in/arbazhkhan/

Abstract

Transformer language models exhibit recurring failure modes under semantic stress, including repetition loops, instruction conflict, and overconfident degeneration. These failures are often attributed to numerical instability or decoding heuristics. In this work, we empirically study internal transformer behavior under semantic stress using a NanoGPT-style model. We show that while residual norms, activation statistics, and train–validation drift remain stable, token-level entropy collapses catastrophically in deeper layers. Based on this observation, we introduce **Layer-5**, a runtime monitoring and intervention mechanism that detects entropy collapse and applies a conservative, entropy-gated correction to internal representations. Across multiple stress conditions, Layer-5 consistently restores internal entropy by an order of magnitude without affecting normal generation. Our results demonstrate the feasibility of internal control planes for transformer stability and safety, independent of training-time alignment or decoding-level heuristics.

1. Introduction

Large language models are increasingly deployed in systems where reliability and safety are critical. Despite strong average-case performance, these models exhibit well-known failure modes under semantic stress, such as repetitive generation, instruction conflict, and self-reinforcing loops. Existing mitigation strategies primarily operate either at training time (e.g., RLHF) or at inference time via decoding heuristics (e.g., temperature scaling, repetition penalties).

However, these approaches do not directly address internal representational failures that occur during model execution. This work asks a foundational question:

How do transformer models fail internally under semantic stress, and can such failures be detected and stabilized at runtime without altering normal behavior?

We investigate this question through controlled experiments on a compact transformer model and introduce **Layer-5**, a minimal runtime mechanism that monitors and stabilizes internal entropy collapse.

2. Experimental Setup

2.1 Model Overview

All experiments are conducted on a NanoGPT-style decoder-only transformer trained on a standard text corpus. The model is intentionally compact to enable detailed inspection of internal activations and residual streams.

2.2 NanoGPT-A2 Architecture

We refer to the specific implementation used in this work as **NanoGPT-A2**. The architecture consists of:

- Learned token embeddings and positional embeddings
- A stack of 6 transformer blocks
- Each block includes:
  - Pre-LayerNorm multi-head self-attention
  - Pre-LayerNorm feed-forward network with GELU activation
  - Residual connections around both submodules
- A final LayerNorm followed by a linear output projection tied to the token embedding matrix

Causal masking is applied to all attention heads, enforcing autoregressive behavior.

Formally, each transformer block computes:

$$x_{l+1} = x_l + \mathrm{MHA}(\mathrm{LN}(x_l)) + \mathrm{MLP}(\mathrm{LN}(x_l))$$

The residual stream dimensionality is fixed at 384 across all layers. No auxiliary losses, safety mechanisms, or decoding-time constraints are used during baseline evaluation.

This architecture enables direct observation and modification of the residual stream at each layer, which is central to the Layer-5 design.

2.3 Training and Tokenization

- Tokenization: Byte Pair Encoding (BPE)
- Context length: 256
- Optimization objective: next-token cross-entropy
- Weight tying between token embedding and output projection

The model is trained until convergence, achieving low loss and perplexity on both training and validation data.

---

3. Baseline Evaluation

3.1 Baseline Metrics

We evaluate the trained model using:

- Training and validation loss
- Perplexity
- Residual stream mean norms per layer
- Residual standard deviation norms per layer
- Activation drift between training and validation distributions

Baseline Results

Metric	Train	Validation
Loss	0.4018	0.4066
Perplexity	1.49	1.50

Residual norms increase smoothly with depth, and standard deviations remain stable across layers. Activation drift between training and validation representations is small, indicating a numerically stable and well-generalized model.

---

4. Stress Testing and Observation

4.1 Semantic Stress Conditions

We apply three classes of semantic stress known to induce degeneration in language models:

1. **Repetition loops** (e.g., repeated tokens or phrases)
2. **Self-referential prompts**
3. **Instruction conflict prompts**

These stresses are applied without modifying decoding parameters unless explicitly stated.

4.2 Initial Hypothesis: Numerical Instability

A natural hypothesis is that semantic stress causes numerical instability, observable through:

- Residual norm explosion
- High z-scores
- Large activation drift

This hypothesis is falsified.
Across all stress conditions, residual norms and z-scores remain within baseline distributions.

---

5. Core Observation: Entropy Collapse

Despite numerical stability, we observe **catastrophic token-level entropy collapse** under semantic stress.

5.1 Entropy Measurement

Token entropy is computed from the output distribution at each transformer block:

$$H(p) = -\sum_{i} p_i \log\, p_i$$

5.2 Layer Localization

Entropy collapse is strongly localized to deeper layers:

Layer	Minimum Entropy (Stress)
0–1	~5.3
2	~0.2
3	~1e-2
4	~1e-5
5	~1e-9

This pattern is consistent across stress types and sampling temperatures.

5.3 Key Insight

Transformer failure under semantic stress manifests as representational overconfidence, not numerical instability.

Entropy is therefore a more appropriate internal signal for failure detection.

---

6. Layer-5: Runtime Monitoring and Intervention

6.1 Design Principles

Layer-5 is designed to be:

- Runtime-only (no training modification)
- Non-invasive
- Entropy-gated
- Layer-localized
- Inactive under normal operation

6.2 Entropy-Gated Intervention

Layer-5 activates when:

$$H < \epsilon \text{and} l \geq 3$$

with $\epsilon = 10^{-3}$.

When triggered, Layer-5 applies a conservative correction to the residual stream:

$$x' = \alpha x + (1 - \alpha)\mu$$

where:

- $x$is the current residual stream
- $\mu$is a rolling mean of past residuals
- $\alpha = 0.9$

A soft norm cap is applied as a safety fallback. No model parameters are modified.

---

7. Results

7.1 Internal Entropy Recovery

Across 20 entropy-collapse events in layer 5:

- Entropy increases by **5–10×** after intervention
- Recovery is consistent across stress types
- No intervention is triggered under normal prompts

7.2 Behavioral Impact

Despite internal stabilization, surface-level generation often remains repetitive. This is expected, as decoding feedback loops may persist even after internal entropy recovery. Importantly, Layer-5 does not degrade normal generation quality.

---

8. Discussion

8.1 Why Output Does Not Immediately Improve

Layer-5 intervenes conservatively and late in the network to avoid unintended side effects. While entropy recovery reopens internal degrees of freedom, stronger or earlier interventions are required to visibly alter generated text.

8.2 Scope for Improvement

Potential extensions include:

- Earlier layer intervention
- Stronger representation re-centering
- Logit-level entropy regularization

These directions are intentionally excluded to preserve a clean and defensible scope.

9. Limitations

- Experiments are conducted on a single model scale
- No agentic or tool-using loops are evaluated
- Output-level recovery is limited

These limitations define clear avenues for future research.

10. Conclusion

We show that semantic stress in transformer models leads to entropy collapse in deep layers despite numerical stability. By detecting and correcting this collapse at runtime, Layer-5 provides a minimal internal control plane that stabilizes representations without affecting normal behavior. This work establishes a foundation for internal safety and stability mechanisms in transformer-based systems.

References

[1] **Vaswani, A., et al.**
Attention Is All You Need.
Advances in Neural Information Processing Systems (NeurIPS), 2017.

[2] **Radford, A., et al.**
Improving Language Understanding by Generative Pre-Training.
OpenAI Technical Report, 2018.

[3] **Radford, A., et al.**
Language Models are Unsupervised Multitask Learners.
OpenAI Blog, 2019.

[4] **Holtzman, A., et al.**
The Curious Case of Neural Text Degeneration.
International Conference on Learning Representations (ICLR), 2020.

[5] **Welleck, S., et al.**
Neural Text Generation with Unlikelihood Training.
International Conference on Learning Representations (ICLR), 2020.

[6] **Kadavath, S., et al.**
Language Models (Mostly) Know What They Know.
arXiv preprint arXiv:2207.05221, 2022.

[7] **Elhage, N., et al.**
A Mathematical Framework for Transformer Circuits.
Anthropic Technical Report, 2021.

[8] **Karpathy, A.**
NanoGPT: A Minimal GPT Training Codebase.
GitHub Repository, 2022.