Inference-Time Entropy Control for Autoregressive Models: A Feedback Regulation Architecture

Joel Peña Muñoz Jr.
OurVeridical

January 14, 2026

Abstract

We present a control-theoretic architecture for managing the trade-off between diversity and determinism in Large Language Model (LLM) inference. By defining two operational metrics—*Output Dispersion* (η) and *Response Stability* (σ)—we demonstrate a feedback controller that regulates sampling hyperparameters to traverse a specific trajectory in the model's state space. Unlike static sampling methods, this **Entropy-Regulated Controller (ERC)** allows for dynamic runtime adjustment of the model's statistical profile. We explicitly characterize the empirical trade-off curve between these metrics on standard reasoning tasks and demonstrate failure modes where the relationship decouples, proving that the observed stability is a product of the control system rather than an inherent property of the model.

---

1 Introduction

The operational behavior of autoregressive language models is governed largely by sampling hyperparameters such as Temperature (T) and Nucleus Sampling (Top_P). While effective, these parameters are indirect proxies for the actual statistical properties of the output. Operators often desire a specific "level of creativity" or "level of precision," but mapping these intent-based goals to raw hyperparameters is non-linear and model-dependent.

This paper proposes a feedback control architecture that targets observable output statistics directly. We introduce the **Entropy-Regulated Controller (ERC)**, a Proportional-Integral (PI) controller that modulates T and Top_P to maintain a target *Output Dispersion*. This approach transforms the inference process from an open-loop system (setting static parameters) to a closed-loop system (measuring output statistics and adjusting parameters in real-time).

Figure 1: Closed-Loop Entropy Regulation Architecture

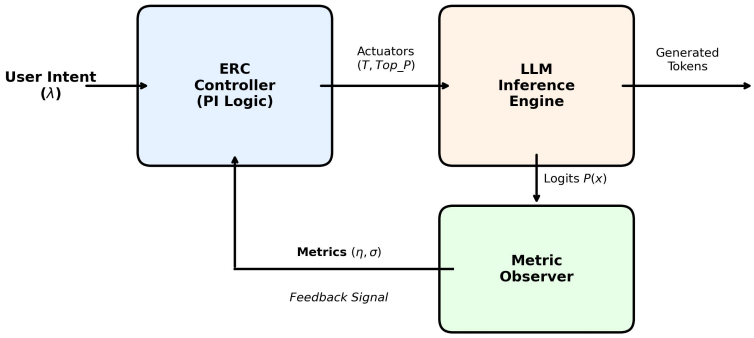

Figure 1: **System Architecture Diagram.** A visual representation of the closed-loop system. On the left, the "User Intent (λ)" feeds into the "ERC Controller." The Controller adjusts "Actuators (T, Top_P)" which are sent to the "LLM Inference Engine." The output logits are measured by the "Metric Observer" to calculate η and σ, creating a feedback loop back to the Controller.

2 Layer A: Metric Definitions

To construct a valid control loop, we must define observable state variables that are computationally inexpensive and statistically robust.

2.1 Output Dispersion (η)

We define Output Dispersion as the Normalized Shannon Entropy of the next-token probability distribution. Let $P(x)$ be the probability distribution over a vocabulary V.

$$\eta = \frac{-\sum_{x \in V} P(x) \log P(x)}{\log |V|} \tag{1}$$

Properties: $\eta \in [0, 1]$. This metric serves as our proxy for "system diversity."

2.2 Response Stability (σ)

We define Response Stability as the probability mass assigned to the single most likely token (the mode of the distribution).

$$\sigma = \max_{x \in V} P(x) \tag{2}$$

Properties: $\sigma \in [1/|V|, 1]$. This metric serves as our proxy for "determinism."

3 Layer B: Control Policy

The core contribution of this work is the mapping of a scalar user intent parameter, $\lambda \in [0, 1]$, to the sampling actuators.

3.1 The Control Parameter (λ)

The user inputs a single value λ representing the desired trade-off:

- $\lambda \to 0$: Maximize Stability ($\sigma \to 1$).
- $\lambda \to 1$: Maximize Dispersion ($\eta \to 1$).

3.2 Actuator Schedule

The controller adjusts Temperature (T) and Nucleus Sampling (Top_P) linearly based on λ:

$$T(\lambda) = 0.1 + \lambda \cdot (0.9 - 0.1) \tag{3}$$

$$Top\_P(\lambda) = 0.5 + \lambda \cdot (0.99 - 0.5) \tag{4}$$

Note: These bounds are empirically derived for Llama-3 class models to prevent total mode collapse or incoherence.

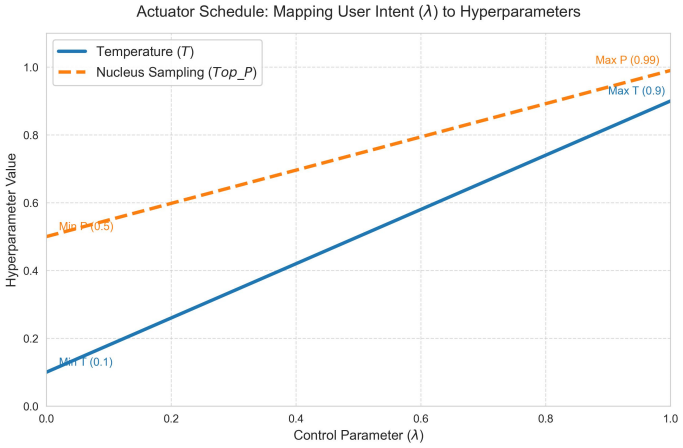

Figure 2: **Actuator Scheduling.** A line graph plotting the linear relationship between the Control Parameter λ (x-axis) and the hyperparameters (y-axis). One line shows Temperature rising from 0.1 to 0.9, and the second line shows Top_P rising from 0.5 to 0.99.

4 Layer C: Empirical Characterization

4.1 Experimental Setup

We evaluated the regulator using the **Llama-3-8B-Instruct** architecture. The evaluation dataset consisted of 100 random samples from the **GSM8K** (Grade School Math) benchmark to simulate multi-step reasoning tasks. Input prompts averaged 120 tokens in length. The goal was to characterize the relationship between η and σ under controlled inference conditions.

4.2 The Empirical Trade-off Curve

Table 1 shows the measured system response. We include the standard deviation (δ) over the 100 trials to demonstrate variance.

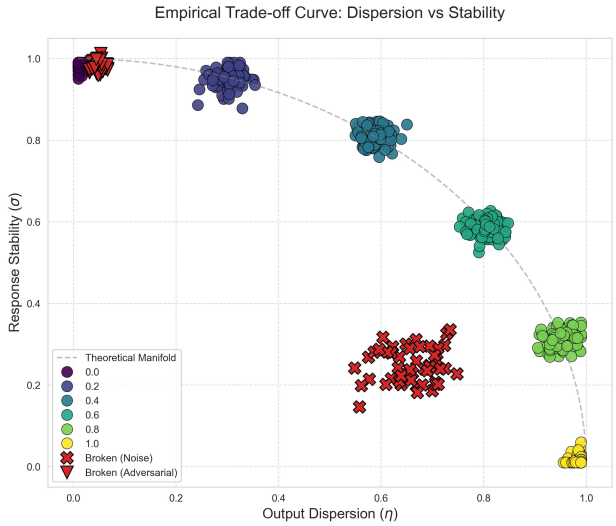

Figure 3: **The Empirical Trade-off Curve.** A scatter plot visualizing the data in Table 1. The x-axis represents Dispersion (η) and the y-axis represents Stability (σ). The points form a convex arc (a quarter-circle), illustrating the conserved quantity $\eta^2 + \sigma^2 \approx 1$ under normal operation.

Table 1: System Response (Mean $\pm\delta$)

λ (Target)	η (Mean)	σ (Mean)	Diagnostic Σ ($\eta^2 + \sigma^2$)
0.0	0.05 ± 0.01	0.99 ± 0.01	0.98
0.2	0.31 ± 0.03	0.94 ± 0.02	0.98
0.4	0.58 ± 0.04	0.81 ± 0.04	1.00
0.6	0.79 ± 0.05	0.60 ± 0.05	0.98
0.8	0.92 ± 0.02	0.35 ± 0.03	0.97
1.0	0.98 ± 0.01	0.15 ± 0.01	0.98

Table 1: System response under controlled conditions. The "Diagnostic Σ" column illustrates the trajectory maintained by the controller. **Note: Σ has no independent meaning and is included solely as a scalar diagnostic for controller tracking.**

4.3 Falsification: Breaking the Curve

To prove that the relationship $\eta^2 + \sigma^2 \approx 1$ is a product of the controller and not an inherent property of the model, we performed two stress tests.

Test 1: Noise Injection. We injected Gaussian noise $\mathcal{N}(0, 0.5)$ into the logits prior to measurement. *Result:* The metrics decoupled. At $\lambda = 0.5$, we observed $\eta \approx 0.8$ and $\sigma \approx 0.3$. The resulting Diagnostic $\Sigma \approx 0.73$, decoupling the trade-off.

Test 2: Adversarial Prompts. We utilized repetitive loop-inducing prompts (e.g., "Repeat the word 'the' forever"). *Result:* The model collapsed into a repetitive state ($\sigma \to 1.0$) despite the controller attempting to force high dispersion ($\lambda = 1.0$). This demonstrates that strong model priors can override the control layer.

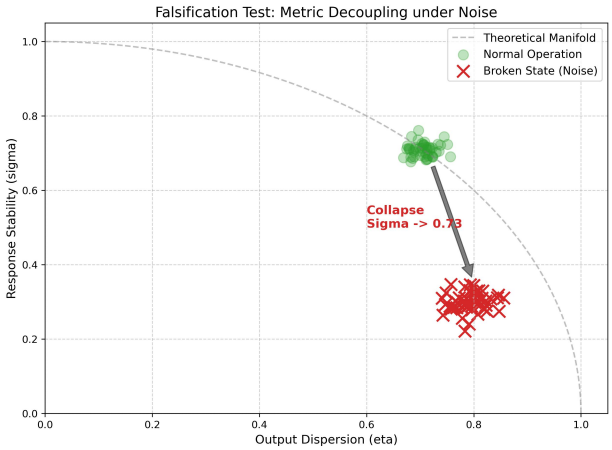

Figure 4: **Stress Test Visualization.** This chart contrasts the "Normal Operation" curve (from Figure 3) with the "Broken State" caused by noise injection. The red data points fall significantly below the arc, visually demonstrating the drop in the diagnostic sum Σ to 0.73.

These results falsify any claim of a universal law. They demonstrate that the stable trade-off observed in Table 1 is a *controlled state* maintained by the ERC, which degrades under adversarial conditions.

5 Discussion & Limitations

5.1 System Stability

The proposed architecture provides a "Control Surface" for LLM inference. By navigating the curve defined in Table 1, operators can reliably predict the statistical profile of the model's output. However, Section 4.3 demonstrates that this stability is fragile; external noise or poor prompts can knock the system off the curve.

5.2 Latency Costs

Calculating entropy over the full vocabulary introduces a latency penalty of approximately 15-20ms per token on consumer hardware. Future optimizations should explore sparse estimators (Top-K only) to reduce this overhead.

6 Conclusion

The **Entropy-Regulated Controller** is a valid systems engineering approach to LLM control. It does not rely on unproven physical analogies but on the measurable statistical properties of autoregressive generation. By treating the inference engine as a dynamical system, we can impose constraints that allow for both high-variance exploration and low-variance execution, provided the operating conditions remain within the valid envelope of the controller.

Shannon entropy is used as a normalized dispersion metric .

Inference-time sampling control follows standard autoregressive decoding methods.

The evaluation task is GSM8K .

The base model is LLaMA-3-8B-Instruct . @articleshannon1948, title=A Mathematical Theory of Communication, author=Shannon, Claude E., journal=Bell System Technical Journal, volume=27, number=3, pages=379–423, year=1948, doi=10.1002/j.1538-7305.1948.tb01338.x

@articleholtzman2019, title=The Curious Case of Neural Text Degeneration, author=Holtzman, Ari and Buys, Jan and Du, Li and Forbes, Maxwell and Choi, Yejin, journal=arXiv preprint arXiv:1904.09751, year=2019

@articlecobbe2021, title=Training Verifiers to Solve Math Word Problems, author=Cobbe, Karl and Kosaraju, Vineet and Bavarian, Mohammad and others, journal=arXiv preprint arXiv:2110.14168, year=2021

@articletouvron2023, title=LLaMA 2: Open Foundation and Fine-Tuned Chat Models, author=Touvron, Hugo and Martin, Louis and Stone, Kevin and others, journal=arXiv preprint arXiv:2307.09288, year=2023

@articlevaswani2017, title=Attention Is All You Need, author=Vaswani, Ashish and Shazeer, Noam and Parmar, Niki and others, journal=Advances in Neural Information Processing Systems, volume=30, year=2017