Inference Stack Degradation Theory

Load-Coupled Numerical Drift and Monitoring in LLM Inference

Version 1.5

Roble Mumin
Independent AI Researcher and AI Consultant
https://roblemumin.com

December 2025

Contents

Abstract

This paper proposes the Inference Stack Degradation Theory (ISDT), a falsifiable hypothesis extending recent findings on nondeterminism in large language model (LLM) inference. Prior work demonstrates that output divergence at temperature=0 arises from batch-size variability and floating-point non-associativity in non batch-invariant transformer kernels.

ISDT hypothesizes that such numerical deviations may, under sustained concurrent load, propagate autoregressively and correlate with measurable task-level changes in output stability. To operationalize this hypothesis, the Prompt Weather Index (PWI) is introduced as a monitoring construct designed to track temporal inference stability. Established empirical findings are clearly separated from conjecture, with explicit scope, null hypotheses, and validation paths defined.

1 Introduction

Reproducibility is a foundational requirement for trustworthy AI systems. Nevertheless, modern large language models frequently exhibit output divergence even when sampling parameters are fully deterministic, including temperature=0.

Recent work by He et al. demonstrates that this behavior arises from batch-size variability induced by inference server load. Dynamic batching alters kernel execution paths and floating-point reduction order. Due to floating-point non-associativity, these changes introduce small numerical deviations that can cascade into token-level divergence.

This paper builds on those findings by asking whether such divergence is always operationally neutral, or whether under sustained load it correlates with observable task-level effects. To support empirical investigation, a monitoring construct, the Prompt Weather Index, is proposed.

2 Scope and Non-Claims

This paper does not claim that batch-induced numerical drift necessarily degrades model quality, nor that inference nondeterminism implies unreliability of large language models.

ISDT does not assert:

- that all nondeterminism results in quality loss,
- that batch-invariant kernels are universally required,
- or that the Prompt Weather Index constitutes a measure of model capability.

The theory is limited to examining whether infrastructure-induced numerical variability can correlate with downstream behavior under sustained load, and whether such correlations are observable and falsifiable.

3 Established Findings from Prior Work

3.1 Floating-Point Non-Associativity

Floating-point arithmetic is not associative:

$$(a + b) + c \neq a + (b + c)$$

due to rounding effects inherent to finite-precision representations.

3.2 Batch Dependence as Practical Cause

Dynamic batching alters kernel execution order and reduction paths. He et al. report that identical prompts at temperature=0 produced multiple unique outputs under default kernels. When batch-invariant kernels were enforced, identical outputs were restored, at a measured performance cost of approximately 1.6× to 2.1×.

4 Relation to ML Systems Research

ISDT aligns with prior observations that machine learning system behavior is shaped not only by training data and model architecture, but also by deployment infrastructure. Sculley et al. describe how hidden technical debt manifests independently of model updates.

ISDT reframes inference nondeterminism as a system-level dynamic rather than an implementation defect, complementing work on reproducibility, evaluation variance, and deployment effects.

5 Inference Stack Degradation Theory

ISDT hypothesizes that batch-induced numerical deviations may propagate autoregressively and, under sustained load, correlate with task-level effects for sufficiently long or precision-sensitive inference sequences.

5.1 Formal Framing

Let p_t denote the ideal token distribution at step t, and $\tilde{p}_t$ the realized distribution under batch-dependent perturbations. Drift is defined as:

$$D_t = \mathrm{KL}(p_t \,\|\, \tilde{p}_t)$$

A recursive structure is proposed:

$$D_t = g(B_t, L_t, K) + \alpha_t D_{t-1}$$

where B_t represents batch characteristics, L_t system load, and K kernel properties.

5.2 Null Hypothesis

Batch-induced numerical drift affects reproducibility only and does not measurably alter task-level correctness or accuracy.

6 Autoregressive Sensitivity Considerations

Autoregressive generation amplifies small distributional differences over long sequences. Even minimal numerical perturbations may cross token selection thresholds when probability mass is redistributed near decision boundaries.

ISDT relies on this sensitivity without assuming that all drift is harmful or persistent.

7 Prompt Weather Index

7.1 Motivation

The Prompt Weather Index (PWI) is a monitoring framework designed to track temporal inference stability. A targeted literature search conducted in December 2025 did not identify prior use of this term.

7.2 Definition

PWI is composed of three observable sub-metrics:

- **Deterministic Consistency Score (DCS)**
- **Semantic Consistency Delta (SCD)**
- **Error Density Factor (EDF)**

A composite index is defined as:

$$\mathrm{PWI} = w_1 \cdot \mathrm{DCS} + w_2 \cdot \mathrm{SCD} + w_3 \cdot (1 - \mathrm{EDF})$$

7.3 Stability Versus Correctness

PWI measures stability, not correctness. High stability does not imply correctness, and low stability does not imply error.

7.4 Falsifiability

Fixed prompts queried periodically yield reproducible datasets. Absence of correlation between PWI and load proxies would falsify ISDT-based monitoring assumptions.

8 Minimal Empirical Validation Path

A minimal validation requires repeated fixed-prompt inference under varying concurrency with controlled model version, routing, and decoding parameters. Even small-scale studies could confirm or refute predicted correlations.

9 Operational and Ethical Considerations

Monitoring inference stability raises fairness considerations. PWI is proposed as an observability signal rather than a gating mechanism. Uniform application and transparency are essential to avoid selective advantage.

10 Implications if Validated

If ISDT and PWI are validated:

- Evaluation may become explicitly load-aware.
- Benchmarks may better separate model capability from infrastructure effects.
- Monitoring may complement, but not replace, batch-invariant kernels.

11 Open Research Questions

- Under what prompt lengths or task classes does autoregressive sensitivity become operationally relevant?
- How do quantization and mixed precision interact with batch-induced drift?
- Can decoding strategies mitigate drift without enforcing batch invariance?

12 Conclusion

ISDT reframes inference nondeterminism as a potentially observable system dynamic rather than a purely stochastic artifact. PWI extends this framing into a practical monitoring proposal. Together, they invite empirical investigation into how infrastructure dynamics shape deployed LLM behavior.

13 Experiments

This section outlines a set of reproducible experiments designed to test the Inference Stack Degradation Theory (ISDT) and evaluate the Prompt Weather Index (PWI). The protocols are explicitly structured for execution within an agentic framework swarm, where specialized LLM agents can autonomously handle setup, code execution, data collection, statistical analysis, and synthesis using available tools (e.g., code interpreters, web search, dataset retrieval).

All experiments use open-source models and tools to ensure verifiability without proprietary dependencies. Recommended models include Meta-Llama-3-8B-Instruct or Qwen2-7B-Instruct. Inference is performed with vLLM ($\geq 0.6.0$), which supports dynamic batching and is widely used in production settings. Agents can parallelize experiments across the swarm, with one agent per sub-experiment and a synthesis agent aggregating results.

13.1 Experiment 1: Reproducing Batch-Induced Token-Level Divergence

13.1.1 Objective

Confirm the baseline nondeterminism reported in prior work and quantify token-level divergence under varying batch sizes and load conditions.

13.1.2 Setup

- Model: `meta-llama/Meta-Llama-3-8B-Instruct`
- Engine: vLLM with dynamic batching enabled
- Prompt: Single fixed prompt requiring ≥ 50 tokens (e.g., from GSM8K: "Solve step by step: What is 137 + 428?" extended with reasoning instructions)
- Conditions: Low load (max_num_seqs=4), High load (max_num_seqs=32)
- Repetitions: 100 identical prompts per condition, temperature=0, max_tokens=128

13.1.3 Expected Metrics

- Number of unique outputs
- Average Levenshtein (edit) distance from the most common output
- Token position of first divergence

13.1.4 Null Hypothesis

No significant difference in divergence between low- and high-load conditions.

13.1.5 Agentic Execution Notes

Verifier Agent confirms model availability on Hugging Face; Experiment Agent runs generations and computes distances; Critique Agent checks for divergence accumulation over sequence length.

13.2 Experiment 2: Measuring Autoregressive Drift Accumulation

13.2.1 Objective

Test whether small numerical perturbations propagate and amplify over long autoregressive sequences under sustained load.

13.2.2 Setup

- Extend Experiment 1 to max_tokens=512
- Use precision-sensitive tasks: multi-step arithmetic chains, code generation (HumanEval prompts), long-form reasoning
- Compare outputs under (a) batch-invariant mode (if supported) vs. (b) dynamic batching with simulated concurrent load

13.2.3 Expected ISDT Outcome

Greater divergence and earlier branching in high-load dynamic batching, especially beyond 100–200 tokens.

13.2.4 Agentic Execution Notes

Experiment Agent measures KL divergence between softmax distributions at each token step (requires access to logits); Critique Agent tests correlation between sequence length and divergence magnitude.

13.3 Experiment 3: Prompt Weather Index (PWI) Time-Series Monitoring

13.3.1 Objective

Compute PWI over a simulated 24-hour period and test correlation with load proxies.

13.3.2 Setup

- Fixed prompt set: 50 prompts (20 GSM8K arithmetic, 20 HumanEval coding, 10 TriviaQA factual)
- Query interval: Every 30 minutes (48 measurements)
- Load schedule: Low (00:00–08:00, 20:00–24:00), Peak (08:00–20:00)
- Sub-metrics:
  - DCS: Exact-match rate across 5 repetitions per prompt
  - SCD: Average cosine similarity of sentence embeddings (`all-MiniLM-L6-v2`)
  - EDF: Task-specific error rate (1 - accuracy)
- Composite PWI:

$$\text{PWI} = 0.4 \cdot \text{DCS} + 0.4 \cdot \text{SCD} + 0.2 \cdot (1 - \text{EDF}) \times 100$$

13.3.3 Null Hypothesis

No significant correlation between PWI and load level (Pearson $|r| < 0.3$).

13.3.4 Elaboration: Metric Sensitivity and Weight Robustness

PWI is intended as an observability signal for temporal *stability*, not a direct measure of correctness or model capability. The inclusion of EDF is optional and should be treated as a diagnostic supplement that helps detect whether instability *co-occurs* with task failure rates, without implying causality.

To reduce arbitrariness in the composite index, the evaluation should include a weight sensitivity analysis:

- Report each sub-metric (DCS, SCD, EDF) separately in addition to the composite.
- Recompute PWI across multiple plausible weightings (e.g., $w_1, w_2, w_3 \in \{0.2, 0.3, 0.4, 0.5\}$ with $w_1 + w_2 + w_3 = 1$) and verify whether correlation conclusions are stable.
- Treat correlation as robust only if direction and magnitude remain consistent across weightings and across prompt subsets (math, code, QA).

If a correlation is observed only for a narrow set of weights or a single prompt class, it should be interpreted as weak evidence and reported as such.

13.3.5 Agentic Execution Notes

Experiment Agent automates periodic queries and metric computation; Synthesis Agent produces time-series plots and correlation statistics, including sub-metric-only analyses and weight sensitivity summaries.

13.4 Experiment 4: Task-Level Correctness Degradation

13.4.1 Objective

Determine whether batch-induced drift correlates with measurable declines in task accuracy.

13.4.2 Setup

- Benchmarks: GSM8K (math), HumanEval (code), TriviaQA (factual QA)
- 200 prompts per benchmark
- Conditions: (a) Low-load deterministic mode, (b) High-load dynamic batching
- Accuracy measured via standard evaluators (exact match for math/QA, pass@1 for code)

13.4.3 Null Hypothesis

Accuracy difference between conditions $\leq 2\%$ (within typical evaluation variance).

13.4.4 Expected ISDT Outcome

Statistically significant accuracy drop ($\geq 5\%$) in the high-load condition, particularly on longer or precision-sensitive problems. This effect size is task-dependent and should not be treated as a universal expectation across models or benchmarks.

13.4.5 Elaboration: Effect Size Interpretation and Confounders

Observed differences between low-load and high-load conditions may reflect multiple interacting infrastructure effects beyond batch-induced reduction-order changes. The experimental design should therefore include explicit confounder controls and reporting:

- **Environment control:** Pin model commit hash, tokenizer version, vLLM version, CUDA and driver versions, and hardware type. Use fixed container images when possible.
- **Scheduling and contention:** Record concurrency, GPU utilization, memory pressure, and request queue depth. Note that scheduler variability and memory contention can alter latency and caching behavior.
- **KV cache dynamics:** Track KV cache reuse or eviction behavior under load, since cache pressure can change effective execution paths and timing.
- **Routing and placement:** Ensure requests are routed to the same replica and hardware, or explicitly model replica variance as a factor.
- **Multiple comparisons:** When testing multiple benchmarks and slices (by length, topic, difficulty), apply correction or clearly label analyses as exploratory.

Falsification boundaries must be explicit: failure to observe an accuracy difference beyond the predefined margin falsifies ISDT for the tested regime, model, and infrastructure configuration, but does not generalize to all deployments.

13.4.6 Agentic Execution Notes

Experiment Agent runs evaluations; Critique Agent performs statistical tests (McNemar where paired decisions apply, or bootstrap for accuracy differences and confidence intervals); Synthesis Agent reports effect sizes, confidence intervals, and confounder logs.

13.5 Implementation Guidelines for Agentic Swarm

- All code should be Python-based, using Hugging Face `datasets`, vLLM, `sentence-transformers`, and standard scientific libraries.
- Agents log raw outputs, metrics, seeds, environment versions, and hardware identifiers for reproducibility.
- Confounders (model updates, routing, replica variance, hardware changes) must be controlled via fixed environment snapshots or explicitly modeled as factors.
- Falsification criteria are explicit per experiment; absence of predicted effects refutes ISDT for the tested regime.

These experiments provide a complete, verifiable pathway from reproduction of known nondeterminism to testing the novel claims of ISDT and PWI. Results can be aggregated by a Synthesis Agent into a structured report for human review.

Acknowledgments

The author thanks the Thinking Machines Lab for advancing reproducibility research in LLM inference.

14 References

[R1] He, Horace, et al. *Defeating Nondeterminism in LLM Inference*. Thinking Machines Lab Technical Blog, September 2025. Available at: https://thinkingmachines.ai/blog/defeating-nondeterminism-in-llm-inference/

[R2] IEEE Computer Society. *IEEE Standard for Floating-Point Arithmetic (IEEE Std 754-2019)*. IEEE, 2019. DOI: 10.1109/IEEESTD.2019.8766229.

[R3] Goldberg, David. *What Every Computer Scientist Should Know About Floating-Point Arithmetic*. ACM Computing Surveys, Vol. 23, No. 1, pp. 5–48, 1991. DOI: 10.1145/103162.103163.

[R4] Sculley, D., Holt, G., Golovin, D., et al. *Hidden Technical Debt in Machine Learning Systems*. Advances in Neural Information Processing Systems (NeurIPS), Workshop on Software Engineering for Machine Learning, 2015. arXiv:1506.03425.

[R5] Brown, Tom B., et al. *Language Models are Few-Shot Learners*. Advances in Neural Information Processing Systems (NeurIPS 33), 2020. arXiv:2005.14165.

[R6] OpenAI. *GPT-4 Technical Report*. arXiv preprint, 2023. arXiv:2303.08774.

15 Glossary

Autoregressive Generation A sequence generation process in which each output token is conditioned on all previously generated tokens, making later outputs sensitive to earlier perturbations.

Batch-Invariant Kernel A computational kernel whose numerical results are independent of batch size, request grouping, or execution order, typically at the cost of reduced throughput.

Dynamic Batching An inference optimization technique that aggregates incoming requests into variable-size batches to improve hardware utilization, potentially altering execution order and numerical behavior.

Deterministic Consistency Score (DCS) The proportion of identical outputs produced from repeated executions of the same prompt under fixed decoding parameters.

Floating-Point Non-Associativity The property that floating-point arithmetic does not guarantee associative behavior due to rounding and finite-precision representation, such that $(a + b) + c \neq a + (b + c)$.

Inference Stack The combined hardware, kernel, runtime, scheduling, and software layers involved in executing model inference in a deployed system.

Inference Stack Degradation Theory (ISDT) A falsifiable hypothesis proposing that batch-induced numerical perturbations may propagate autoregressively and correlate with observable task-level effects under sustained system load.

Prompt Weather Index (PWI) A proposed composite monitoring index intended to track temporal variability and stability in LLM inference behavior across repeated prompts.

Reproducibility The ability to obtain identical outputs from repeated executions of a model under identical inputs, parameters, and system conditions.

Semantic Consistency Delta (SCD) An embedding-based metric quantifying semantic variation between outputs that may differ lexically but share meaning.

Temperature = 0 A decoding configuration in which stochastic sampling is disabled and the most probable token is selected at each generation step.

Task-Level Correctness A measure of whether a model output satisfies the functional or semantic requirements of a given task, independent of stylistic variation.

A Agentic Peer Review Protocol

This appendix outlines a structured multi-agent protocol for evaluating ISDT using verifier, novelty, critique, experiment, and synthesis agents. It complements, but does not replace, human peer review.

B Illustrative Simulation Scripts

B.1 Toy Autoregressive Drift Simulation

```python
import random
import numpy as np

def simulate_drift(steps=10, high_load=False, seed=42):
    random.seed(seed)
    np.random.seed(seed)

    true_sum = 0.0
    low_load_sum = 0.0
    high_load_sum = 0.0
    deviations = []

    for step in range(1, steps + 1):
        value = 1.01 * step
        true_sum += value
        low_load_sum += value

        if high_load:
            perturbed = value + random.uniform(-1e-6, 1e-6)
            high_load_sum += perturbed
        else:
            high_load_sum += value

        deviations.append((
            step,
            abs(low_load_sum - true_sum),
            abs(high_load_sum - true_sum)
        ))

    return deviations
```

B.2 Mock Prompt Weather Index Generator

```
import random
from datetime import datetime, timedelta

def generate_pwi_day(start_date="2025-12-13", seed=42):
    random.seed(seed)
    base_time = datetime.strptime(start_date, "%Y-%m-%d")
    data = []

    for hour in range(24):
        if 13 <= hour <= 19:
            dcs, scd, edf = random.uniform(75,85), random.uniform(70,80), random.
    uniform(0.25,0.35)
        else:
            dcs, scd, edf = random.uniform(90,100), random.uniform(75,90), random.
    uniform(0.15,0.25)

        pwi = (0.4 * dcs) + (0.4 * scd) + (0.2 * (1 - edf) * 100)
        data.append((hour, dcs, scd, edf, pwi))

    return data
```

Appendix Notes

All scripts are illustrative only and do not model real transformer kernels or GPU execution.