Telemetry Degradation and Sequential Stability in LLM-Integrated IoT Decision Systems

Julio C. Amador Díaz López, Christopher Abou-Najm, Saksham Dawhan, Alina-Irina Serban, Payam Barnaghi and Ravi Vaidyanathan

***Abstract*—AI systems for human performance/interface must transform raw IoT telemetry into consistent actionable output. Large language models (LLMs) function as such interpreters. However behavioral stability through real-world data degradation remains unexplored; applications demanding consistent output (e.g. medicine, athletics) remain unrealized.**

We introduce a framework to evaluate behavioral stability of LLM decision systems transforming IoT telemetry to semantic output. The framework is validated in a case study of AI-tennis coaching using stroke-level IoT smartwatch streams. Local perturbations are mapped to global output through graded dropout, resolution downsampling and reliability signaling. We establish baseline temporal stability under idealized conditions, then quantify impact of increasing degradation severity on sequential decision behavior. Temporal volatility (FlipRate), directional escalation and cumulative safety outcomes are assessed.

We show telemetry degradation significantly increases decision instability in already unstable systems. In a balanced regime, FlipRate rises from 0.144 under clean conditions to 0.207 under downsampling (+44%), while in a serve-dominant regime it reaches 0.300. Regimes characterized by strong feature dominance exhibit amplified sensitivity to information loss. A workload accumulation model shows observed volatility constitutes a structural pathway to safety-bound violations. Reliability signaling alone produces smaller effects than structural information loss, indicating informational cues cannot compensate for lost telemetry. Structured prompting partially mitigates instability but does not eliminate degradation-induced effects.

We argue behavioral stability in LLM-integrated IoT systems must be evaluated jointly with sensing reliability and regime structure. Our robustness analysis provides a basis for LLM-telemetry interpretation, opening new AI-implementation avenues in human performance and digital health.

***Index Terms*—Internet of Things, Large Language Models, Telemetry Reliability, Sequential Decision Stability, Edge Intelligence, Cyber-Physical Systems, Transport-Layer Degradation, System Robustness Evaluation.**

This work was supported by The U.K. Dementia Research Institute (DRI) which receives its funding from U.K. DRI Ltd., funded by the U.K. Medical Research Council (MRC), Alzheimer's Society and Alzheimer's Research U.K. (Grant UKDRI-7005), by SiftyML LTD and the Department of Mechanical Engineering at Imperial College London (ICL).

Julio C. Amador Díaz López, Christopher Abou-Najm, Saksham Dawhan, Alina-Irina Serban, and Ravi Vaidyanathan are with the Department of Mechanical Engineering, Imperial College London and the U.K. DRI Care Research & Technology Centre, SW7 1AL London, U.K.

Payam Barnaghi is with the Department of Brain Sciences, Imperial College London and the U.K. DRI Care Research & Technology Centre, W12 0BZ London, U.K.

Julio C. Amador Díaz López is also with SiftyML LTD, London, UK. Correspondence should be addressed to: j.amador@imperial.ac.uk and/or r.vaidyanathan@imperial.ac.uk

Code and data to reproduce our results are available at https://github.com/julioadl/telemetry-degradation

I. Introduction

Large language models (LLMs) are increasingly deployed beyond text generation and into decision-support roles within cyber-physical and IoT-integrated systems. In such settings, models consume structured telemetry streams and produce actionable outputs that influence physical behavior. While extensive evaluation exists for reasoning benchmarks and static prompt robustness [1], [2], comparatively little attention has been paid to how LLM decision policies behave under degraded streaming telemetry conditions typical of real-world IoT deployments.

Beyond traditional automation contexts, this challenge arises in a growing class of applications where LLMs interpret longitudinal sensor data to produce human-facing recommendations. Examples include digital health monitoring, rehabilitation and therapy support, sports analytics, and other human performance systems in which wearable or environmental sensors stream behavioral telemetry to AI-driven decision modules (see [3] for examples). In particular, this study was motivated by the potential applicability of LLMs in Living Labs deployments like the UKDRI CR&T[1]. Living Labs deploy multimodal sensing for personalized health monitoring, turning home-sensor data into dementia care guidance [4], [5]. In all of these settings, LLM outputs function not merely as descriptive summaries but as actionable guidance that may influence training, treatment, or behavioral adaptation. Understanding how telemetry reliability affects sequential decision behavior is therefore critical for translating LLM-based telemetry interpretation into real-world human-interaction systems. In operational environments, telemetry is subject to packet loss, resolution reduction, and transport-layer instability [6], [7]. These impairments alter the informational content of aggregated sensor summaries before they reach the decision layer. When an LLM functions as a controller within such a pipeline, degradation at the sensing or transport layer may perturb decision stability. Crucially, this instability need not manifest as overt semantic failure; instead, it may appear as increased volatility among otherwise syntactically valid and semantically plausible outputs.

LLM robustness evaluation typically centers on a fundamentally local question: *does output remain correct under perturbation?* While appropriate for single-turn NLP benchmarks, this is insufficient for systems wherein an LLM functions as a sequential controller. When outputs are consumed repeatedly within an IoT pipeline, stability must be evaluated at the

[1]UK Dementia Research Institute Care Research and Technology Centre

level of *decision trajectories*, not individual responses. The key distinction is between *local correctness*; i.e., whether each recommendation is individually plausible, and *global stability*, whether the sequence of recommendations constitutes a coherent, trustworthy policy over time. This work makes operational that distinction within a controlled degradation environment.

Our investigative goal is to quantify behavioral sensitivity of an LLM-based decision module under graded telemetry degradation. As a case study, we examine a structured tennis coaching scenario with synthetic yet temporally coherent stroke-level telemetry, providing a basis for evaluation of LLM output from human data stream. We evaluate how structural information loss in the form of random dropout, systematic downsampling, and their interaction, affects decision stability across multiple regime structures. The LLM produces discrete action classes at each telemetry window, enabling quantitative measurement of temporal volatility, directional drift, and cumulative workload dynamics.

Our analysis yields several findings. First, decision volatility is non-zero even under clean telemetry, indicating that LLM policies exhibit intrinsic stochasticity. Telemetry degradation amplifies this baseline variability rather than introducing instability into an otherwise deterministic controller. Second, degradation sensitivity is regime-dependent: in dominance-structured regimes where outcomes depend heavily on a small subset of features, decision volatility increases more sharply. Third, the structure of missingness matters; i.e., systematic downsampling produces substantially larger instability than random dropout at comparable nominal information loss. Fourth, degradation effects vary across stress dimensions and interact with decoding temperature, revealing a multi-parameter stability landscape rather than a single severity axis. Structured prompting reduces volatility but does not eliminate degradation-induced sensitivity.

Despite rising decision volatility, cumulative safety violations do not occur within our evaluation window. In our setup, the LLM processes each telemetry window independently without access to prior decisions or full telemetry history, and safety outcomes are computed post hoc through the workload accumulation model. Consequently, the absence of violations in some conditions should not be interpreted as evidence of intrinsic safety. Rather, the results suggest that sequential decision variability can increase systematically with telemetry degradation, even when safety thresholds are not triggered under the specific workload model and evaluation configuration.

By isolating the LLM decision layer within an IoT-style telemetry pipeline and sweeping graded degradation regimes, this study demonstrates that evaluation under clean data conditions is insufficient for deployment assurance. Stability emerges from interactions among sensing fidelity, feature dominance structure, sampling configuration, and interface constraints. These findings motivate system-level robustness evaluation when integrating LLMs into IoT decision workflows.

II. Related Work

A. LLMs in Decision-Support and Cyber-Physical Systems

LLMs are no longer confined to text generation and are increasingly embedded as decision-support modules within cyber-physical and IoT-integrated systems. In embodied and robotics contexts, language models have been used for high-level task planning and policy generation, translating natural language objectives into executable control strategies [8]–[11]. Beyond robotics, emerging IoT-oriented deployments integrate LLMs into telemetry-to-decision pipelines, where structured sensor summaries or symbolic state representations are consumed to produce actionable outputs that influence downstream processes such as orchestration, monitoring, and adaptive control [12], [13].

Advances in LLM capabilities have expanded their role into decision-support and agentic contexts. Khan et al. [14] survey developments across reasoning, adaptability, efficiency, and ethics, highlighting techniques such as Chain-of-Thought prompting, instruction tuning, and reinforcement learning from human feedback. They also note challenges including high computational cost, bias, and limited interpretability. This framing of LLMs as agentic systems motivates evaluation beyond single-instance correctness, as considered in the present work for streaming IoT telemetry under degraded sensing conditions. A foundational property motivating this work is the intrinsic inconsistency of pretrained language models under semantically equivalent inputs. Elazar et al. [15] introduce PARAREL, a benchmark of paraphrased factual queries, and show that models often produce different answers to equivalent inputs, with substantial variability across relations. This result indicates that output variability can arise even under semantically identical inputs, directly supporting the non-zero baseline FlipRate observed in the present work under clean telemetry conditions. Stureborg et al. [16] provide complementary evidence for output inconsistency in LLMs, showing low agreement across repeated evaluations of the same input and sensitivity to minor prompt variations. While their setting focuses on evaluation tasks, the core finding that repeated queries on equivalent inputs can yield inconsistent outputs directly parallels the non-zero baseline FlipRate observed in the present work and reinforces the conclusion that sequential decision variability can arise from stochastic decoding rather than only from degraded input conditions.

Existing evaluation of LLMs primarily focuses on reasoning benchmarks, prompt robustness, hallucination behavior, and static input performance [17]–[19]. While these studies characterize semantic reliability, they typically assume well-formed and stable inputs. Recent work argues that traditional LLM evaluation frameworks are insufficient for real-world deployment. Shukla [20] proposes a multidimensional framework that goes beyond standard accuracy-based benchmarks by incorporating aspects such as safety, explainability, and robustness. In particular, the emphasis on robustness under non-ideal or perturbed inputs aligns with the present work, which evaluates LLM behavior under structured telemetry degradation in streaming IoT pipelines. Systematic evaluation of LLM sensitivity to input variation has been advanced

through robustness benchmarking. Zhu et al. [21] introduce PromptRobust, showing that small, semantically plausible perturbations to prompts can lead to substantial performance degradation across tasks and settings. This highlights the fragility of LLM behavior under minor input variations at the textual level. While these perturbations operate directly on the prompt surface, the present work considers a complementary setting in which instability arises from upstream telemetry degradation, demonstrating that sensitivity can originate from the sensing layer even when prompt semantics remain unchanged. Complementary work has addressed safety at the interface level rather than within the model itself. Shvetsova et al. [22] propose a modular API-layer filtering framework that performs bidirectional control over LLM interactions, including prompt injection mitigation, toxicity reduction, and contextual grounding through retrieval. Their results show that such external filtering layers can substantially improve robustness and safety while introducing manageable latency overhead. This perspective reinforces the view adopted in the present work that reliable LLM behavior depends on the full system pipeline, not only the model, particularly in settings where upstream data quality and interface design jointly influence decision stability.

When LLMs operate as decision layers within streaming systems, input variability arises not only from semantic ambiguity but also from transport-layer degradation. Recent work has begun exploring LLMs in embodied and control-adjacent contexts [8]–[11], yet systematic evaluation of their behavioral stability under degraded telemetry remains limited.

Recent surveys explicitly document emerging integration patterns between LLMs and IoT pipelines (e.g., natural-language interfaces over sensing/actuation and telemetry-to-decision workflows), while highlighting deployment constraints driven by edge compute, memory, and latency budgets [12]. Complementing these overviews, IoT-focused frameworks demonstrate how LLM reasoning can be augmented with real-world sensor data, and edge-LLM surveys summarize design and runtime techniques required for resource-constrained on-device inference [13], [23].

B. Telemetry Reliability and Degradation in IoT Systems

IoT pipelines operate under imperfect sensing and communication conditions. Packet loss, sampling irregularity, resolution reduction, and edge-compute constraints alter the informational content of telemetry streams. Classical IoT robustness research addresses fault detection, data imputation, and network resilience [6], [24]. In control systems literature, degradation is analyzed in terms of observability, stability, and noise propagation.

Recent work highlights that deploying LLMs in edge and mobile environments introduces systems-level constraints beyond model accuracy. Qu et al. [25] survey mobile edge intelligence for LLMs, showing that edge deployment offers benefits such as reduced latency, improved privacy, and lower bandwidth use, while being constrained by limited compute, memory, and communication resources. Their work emphasizes resource-efficient design and deployment architectures that balance device and network capabilities, underscoring the need to evaluate LLM systems under realistic IoT constraints.

These analyses, however, typically assume deterministic controllers or probabilistic filters with explicitly modeled uncertainty. When an LLM replaces or augments a traditional inference module, degradation affects the statistical structure of token-level generation conditioned on compressed telemetry summaries. The resulting behavioral consequences are not directly captured by traditional signal-processing or control-theoretic models.

C. Sequential Stability and Accumulated Risk

Safety analysis in sequential decision systems emphasizes trajectory-level behavior rather than single-instance correctness. In reinforcement learning and control contexts, small local errors can accumulate into drift or instability over time [26].

Recent work connects LLM safety with control-theoretic ideas. Miyaoka and Inoue [27] propose CBF-LLM, which applies a control barrier function as a safety filter over token generation, intervening on model outputs to enforce safer behavior. This formulation treats alignment as a constrained control problem rather than purely a prompting or fine-tuning task, providing a principled mechanism for filtering unsafe trajectories during generation. Evaluation therefore considers cumulative state evolution and constraint violations.

In contrast, most LLM evaluation remains largely instance-based, focusing on prompt–response pairs [17], [18]. However, when deployed in embodied, agentic, or decision-support settings, LLM outputs are consumed as part of sequential interaction or control loops [11], [28]. Even if individual output is syntactically coherent and semantically plausible, temporal volatility or directional bias may propagate into unsafe trajectories. There is a severe dearth in the investigation of trajectory-level effects of LLMs embedded within IoT telemetry pipelines.

D. Literature Synopsis

In its current state, the field has established: 1) LLM outputs are intrinsically inconsistent under semantically equivalent inputs [15], [16], implying non-zero baseline volatility even under ideal conditions; 2) IoT and edge deployments impose sensing and transport constraints that systematically degrade the informational content of inputs before they reach the decision layer [6], [25]; and 3) safety in sequential decision systems must be evaluated at the trajectory level, not the instance level [26], [27]. However, no prior work has systematically examined the intersection of these three principles: *How does intrinsic LLM instability interact with upstream telemetry degradation to create trajectory-level behavioral instability in IoT-integrated systems*? This investigation bridges that gap.

E. Research Contributions

AI systems in human performance/interface must transform raw IoT telemetry into consistent human actionable output. Large language models (LLMs) emerge as such interpreters. However behavioral stability through real-world

data degradation remains unexplored; potential in applications demanding consistent output (e.g. medicine, athletics) remains unrealized. We introduce a framework to evaluate behavioral stability of LLM decision systems transforming IoT telemetry to semantic output. The framework is validated in a case study of tennis coaching using stroke-level IoT smartwatch streams. Local perturbations are mapped to global output through graded dropout, resolution downsampling, and reliability signaling. We establish baseline temporal stability under idealized conditions, then quantify impact of increasing degradation severity on sequential decision behavior. Temporal volatility (FlipRate), directional escalation, and cumulative safety outcomes are assessed. Telemetry degradation significantly increases decision instability in already unstable systems. In a balanced regime, FlipRate rises from 0.144 under clean conditions to 0.207 under downsampling (44% increase), while in a serve-dominant regime it reaches 0.300. Regimes characterized by strong feature dominance exhibit amplified sensitivity to information loss. A workload accumulation model shows observed volatility constitutes a structural pathway to safety-bound violations. Reliability signaling alone produces smaller effects than structural information loss, indicating informational cues cannot compensate for lost telemetry. Structured prompting partially mitigates instability but does not eliminate degradation-induced effects. We argue behavioral stability in LLM-integrated IoT systems must be evaluated jointly with sensing reliability and regime structure. Our robustness analysis provides a basis for LLM- telemetry interpretation, opening new implementation avenues in human performance and digital health.

We establish the veracity of our behavioral stability framework through: (i) formulation of a baseline temporal stability structure under clean telemetry; (ii) quantification of how graded degradation severity modulates decision volatility; (iii) examination of how regime structure (e.g. feature dominance) amplifies sensitivity to information loss; and (iv) demonstration of the impact of local decision instability on trajectory-level safety outcomes through a cumulative workload model.

By separating structural degradation from declarative reliability signaling and by embedding decisions in a sequential safety framework, this study provides empirical evidence that telemetry reliability and regime structure jointly determine behavioral stability. To our knowledge, prior work has not systematically characterized these interactions. More fundamentally, the field lacks a clear conceptual separation between *local* correctness (individual output plausibility) and *global* stability (trajectory-level policy coherence). Existing robustness benchmarks, for example, prompt perturbation, factual consistency, hallucination rate, evaluate local behavior. Deployment in embodied, health-monitoring, or coaching contexts requires the more demanding criterion: *Does local coherence compose into global stability under perturbation?* Our experiments provide the first structured evidence that it may not, and characterize the conditions under which the local–global gap widens.

F. Investigative Hypotheses

We approach the question of local coherence mapping to global stability under the umbrellas of the local–global stability framework. We offer proof/disproof of 3 working hypothesis to establish each (*i.e.* Local Hypothesis (HL1, HL2, HL3) and Global Hypothesis (HG1, HG2, HG3). We summarize each set of hypothesis:

Local Stability; Key question A: *Does degradation perturb individual decisions?*

HL1 (Intrinsic stochasticity): FlipRate is non-zero under idealized telemetry, therefore LLM decision policy exhibits baseline local variability independent of any degradation.

HL2 (Local amplification): Local decision volatility (FlipRate) increases monotonically with the severity of real-world telemetry degradation.

HL3 (Missingness structure): At comparable nominal information loss, systematic downsampling perturbs local decisions more severely than random dropout due to causal relationships between temporal regularity and local output stability.

Global Stability; Key question B: *Do locally plausible decisions map to coherent sequential policy?*

HG1 (Regime gain): Regime dominance structure modulates the propagation from local perturbation to trajectory-level divergence. Balanced regimes buffer, dominant regimes amplify.

HG2 (Stochasticity interaction): Higher decoding temperature amplifies trajectory-level instability under degradation. Local sampling uncertainty and input uncertainty compound globally, not independently.

HG3 (Interface damping): Structured prompting partially restores global stability, but the local–global gap persists under sufficient degradation. Interface constraints act as a damping layer, not a corrective mechanism.

The central question linking both groups is whether local volatility (HL1–HL3) remains contained at the individual decision level, or whether it composes into trajectory-level divergence (HG1–HG3).

III. System Architecture

Fig. 1 illustrates the end-to-end telemetry-to-decision pipeline for taking in streaming biomechanical data data transformed into human actionable output to test the behavioral stability framework.

A. Deployment Context

The evaluated system represents an LLM-based decision module embedded within a streaming IoT telemetry pipeline. In a realistic deployment scenario, stroke-level motion data from wearable sensors would be processed into compact kinematic summaries and transmitted to a decision layer responsible for generating actionable coaching recommendations in near real time.

The present study isolates this decision layer in order to evaluate its behavioral stability under controlled telemetry

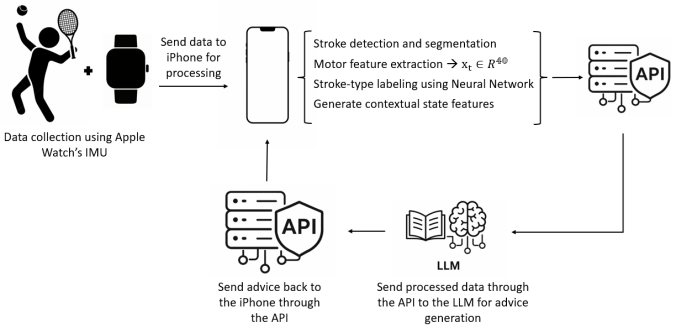


Fig. 1. End-to-end system pipeline: Apple Watch IMU telemetry is processed on iPhone, summarized, sent via API to the LLM for coaching advice, and returned to the iPhone.

degradation. The system operates in an open-loop configuration: telemetry windows are processed independently, and no adaptive learning or closed-loop control is active during experimentation. The objective is to characterize how degraded telemetry affects structured LLM decisions.

B. Telemetry Representation

Stroke-level telemetry is represented as a fixed-dimensional motor feature vector and contextual state variables.

Each stroke t is associated with:

- A motor feature vector $\mathbf{x}_t \in \mathbb{R}^{40}$,
- A stroke-type label (e.g., serve, forehand, backhand),
- Contextual state features (e.g., rally phase, score pressure, momentum indicators),
- A binary outcome label indicating point success.

The motor feature vector consists of aggregated inertial descriptors derived from acceleration and gyroscope signals. Features include per-axis statistical summaries (mean, variance), magnitude-based aggregates, variability indicators, and derived motion intensity proxies. These features are intended to approximate compact stroke-level representations typically computed at the edge of an IoT sensing pipeline. Full feature definitions are provided in Appendix A.

C. Window Aggregation

A sliding window of fixed length (20 strokes, stride 5 strokes) is applied to the telemetry stream. For each window, summary statistics are computed, including:

- Mean and variance of motor features across strokes,
- Stroke-type distribution within the window,
- Win/loss rate within the window,
- A global motor intensity proxy derived from aggregated motion statistics.

All downstream decisions are based on these window-level summaries rather than raw stroke sequences.

D. Prompt Construction and Decision Interface

For each window, a structured prompt is constructed containing:

- Window summary statistics,
- Contextual state information,
- Optional telemetry reliability metadata (a `data_quality` block).

The prompt instructs the model to return a structured JSON object consisting of:

```
{ "action_class": one of {increase_intensity,
maintain, decrease_intensity, rest,
technique_focus},
"cue": "...",
"uncertainty_acknowledged": true/false }
```

The discrete `action_class` constitutes the evaluated decision variable. The textual cue is retained for interpretability but is not directly used in quantitative metrics.

E. Telemetry Stress Injection

To evaluate robustness under degraded streaming conditions, controlled stressors are applied at the stroke level prior to window aggregation. Degradation severity is parameterized along two axes:

- **Dropout (D):** A fraction D of strokes within a window is randomly removed.
- **Downsampling (K):** Only every K-th stroke is retained prior to aggregation.

These stressors reduce the informational resolution of window-level summaries while preserving prompt structure and decision interface. Combined stress experiments apply dropout followed by downsampling.

In addition to structural degradation, an informational reliability condition is evaluated:

- **Reliability Signaling:** The telemetry content remains unchanged, but a metadata block indicating delay or staleness (e.g., `delay_seconds=5`) is included in the prompt.

No temporal misalignment or stale-window substitution is implemented. The reliability manipulation therefore isolates the effect of declarative uncertainty cues from structural information loss.

F. Evaluation Configuration

The system operates in an open-loop evaluation mode. Each telemetry window is processed independently by the LLM. No prior decisions, cumulative workload state, or historical outputs are included in subsequent prompts. Safety and trajectory-level effects are evaluated offline from the sequence of structured action outputs.

IV. System Implementation

A. LLM Configuration

All experiments are conducted using the Groq-hosted LLaMA-family model `llama-3.3-70b-versatile`. At the time this research is carried out, this model is considered *state-of-the-art* within the Open Source Software models. The model is accessed exclusively via the Groq API; no local inference, fine-tuning, or mixed-model configuration is used.

The API is invoked in structured JSON mode with:

- `response_format={"type":"json_object"}`,
- Deterministic schema validation of returned fields.

Unless otherwise specified, decoding temperature is set to $T = 0.2$. A dedicated control experiment sweeps temperature $T \in \{0.0, 0.2, 0.5\}$ to evaluate interaction between telemetry degradation and decoding stochasticity. All other experiments retain $T = 0.2$.

Each model invocation returns a structured JSON object containing: `action_class`, `cue`, and `uncertainty_acknowledged`. Responses are parsed and validated against a predefined action ontology. Malformed or schema-inconsistent outputs are logged; fallback parsing is applied where possible to preserve traceability.

B. Windowing and Aggregation

Telemetry is processed using a sliding window:

- Window size: 20 strokes,
- Step size: 5 strokes.

For each window, summary statistics are computed after stress injection (if applicable). These include:

- Aggregated motor feature statistics,
- Stroke-type distributions,
- Outcome rates,
- Global intensity proxies.

No prior decisions, workload state, or historical outputs are included in prompt construction. Each window is evaluated independently in open-loop mode.

C. Telemetry Stress Injection

Stress is applied at the stroke level prior to window aggregation. Rather than fixed stress categories, degradation severity is parameterized along two axes:

- **Dropout level (D):** A fraction D of strokes within each window is randomly removed. Severity sweeps evaluate $D \in \{0.00, 0.10, 0.20, 0.30, 0.40, 0.60\}$.
- **Downsampling factor (K):** Only every K-th stroke is retained prior to aggregation. Severity sweeps evaluate $K \in \{1, 2, 3, 5, 10\}$, where $K = 1$ corresponds to clean telemetry.

Combined stress experiments apply dropout followed by downsampling. These operations reduce the informational resolution of window summaries without altering prompt structure.

D. Reliability Signaling

In addition to structural degradation, a reliability signaling condition injects metadata into the prompt's `data_quality` block (e.g., `delay_seconds=5`) without modifying telemetry content.

No temporal misalignment, stale-window substitution, or transport delay simulation is implemented. This condition isolates the effect of declarative reliability cues from structural information loss.

E. Experimental Replication

To account for stochastic decoding variability at nonzero temperature, each experimental configuration is repeated 10 times with independent API calls. Variability across replicates arises solely from stochastic decoding (model sampling at nonzero temperature); all other factors; i.e. telemetry seed, stress parameters, prompt content, are held identical across replicates within a condition.

All replicates share identical telemetry seeds, stress parameters, and prompts. Variability across replicates arises solely from model sampling.

F. Logging and Reproducibility

For each window and replicate, the following artifacts are logged:

- Window summary statistics,
- Applied stress parameters (D, K, T, prompt variant),
- Full raw JSON model response,
- Parsed decision fields.

Aggregated metrics (FlipRate, EscalationRate, IncoherenceRate, ViolationRate, Time-to-Violation, NetLoadDrift defined in section V) are computed post-hoc from stored logs.

Datasets, regime generation procedures, and random seeds remain fixed across severity sweeps to ensure comparability.

API credentials and model identifiers are configured via environment variables to support reproducible execution under controlled conditions.

V. Experimental Design

This section formalizes the experimental structure used to evaluate behavioral stability of the LLM decision layer under controlled telemetry degradation. The design isolates three interacting factors:

1) Telemetry regime structure,
2) Transport-layer degradation severity,
3) Decoding stochasticity and prompt structure.

All evaluations are conducted in open-loop mode: each telemetry window is processed independently, and no historical decisions or cumulative state variables are included in prompts. Trajectory-level effects are computed post-hoc from logged action sequences.

A. Experimental Matrix

Experiments follow a fully parameterized design across the following axes:

- **Regimes (R):**
  - R1: Balanced baseline,
  - R2: Variance-perturbed,
  - R3: Serve-dominant.

  Regime construction is detailed in Appendix A.
- **Degradation Severity (S):** Structural stress is applied prior to window aggregation:
  - Dropout levels: $D \in \{0.00, 0.10, 0.20, 0.30, 0.40, 0.60\}$,
  - Downsampling factors $K \in \{1, 2, 3, 5, 10\}$,

  - Combined dropout + downsampling.
- **Temperature (T):** Decoding temperature $T \in \{0.0, 0.2, 0.5\}$ for stochasticity sensitivity.
- **Prompt Variants (P):**
  - Baseline prompt,
  - Structured prompt with reliability framing.

Dropout and downsampling are applied at the stroke level before window aggregation, ensuring that degradation alters the informational content of summary statistics rather than merely the textual prompt representation. Prompts are presented in Appendix B.

For each configuration (R, S, P, T), we run $R_{\text{rep}} = 10$ independent replicates using identical telemetry inputs but independent model invocations.

Let:

$$R = \text{number of regimes}, \quad S = \text{number of stress conditions},$$
$$P = 2, \quad W = \text{windows per dataset}.$$

The total number of LLM evaluations is:

$$N_{\text{calls}} = R \times S \times P \times W \times R_{\text{rep}}.$$

We present a summary of all of our experiments in Table I. The full experimental pipeline is publicly available at `https://github.com/julioadl/telemetry-degradation`.

B. Decision Volatility Metrics

Let $a_t^{(r)}$ denote the discrete action selected at window index t in replicate r, with T total windows per dataset.

FlipRate: Temporal instability is measured as:

$$\text{FlipRate} = \frac{1}{T-1} \sum_{t=2}^{T} \mathbb{1}\left[a_t \neq a_{t-1}\right].$$

Across the 10 replicates per condition, FlipRate is reported as the mean over replicates: $\overline{\text{FlipRate}} = \frac{1}{N_{\text{rep}}} \sum_{r=1}^{N_{\text{rep}}} \text{FlipRate}^{(r)}$. To quantify variability, we report the standard error of the mean $\widehat{\text{SE}} = \sigma_{\text{FR}} / \sqrt{N_{\text{rep}}}$ across replicates as shaded bands in figures show $\pm 2\,\widehat{\text{SE}}$, providing approximate 95% confidence intervals under the assumption of normality.

FlipRate captures local decision volatility independent of direction. A change in recommendation is undesirable when it is induced by model instability rather than a genuine change in the underlying physiological state. Given the 20-stroke window with a 5-stroke step, adjacent windows share a large proportion of telemetry, variability that correlates with telemetry degradation rather than a real change in athlete condition constitutes the form of instability FlipRate is designed to detect. FlipRate counts all consecutive changes in recommended action, including minor adjustments that might be contextually appropriate. This makes it conservative; i.e., it over-counts relative to a metric restricted to disruptive transitions only. However, this conservatism strengthens the argument: even with a permissive metric, degradation produces measurably higher instability. When an LLM loses informational grounding, the resulting flips tend to be large-magnitude transitions (e.g., `increase_intensity` → `decrease_intensity`), so FlipRate captures the most operationally relevant instability in practice. The EscalationRate and IncoherenceRate metrics, to be introduced next, provide complementary directional and coherence-based views.

EscalationRate: To isolate directional drift toward higher workload intensity, define $\mathcal{E}$ as the set of transitions corresponding to upward intensity changes (e.g., maintain → increase_intensity, decrease → increase_intensity).

$$\text{EscalationRate} = \frac{1}{T-1} \sum_{t=2}^{T} \mathbb{1}\left[(a_{t-1}, a_t) \in \mathcal{E}\right].$$

EscalationRate measures systematic bias toward higher workload actions.

IncoherenceRate: We additionally measure structural inconsistency in action sequences. Let $\mathcal{I}$ denote implausible or logically inconsistent transitions (e.g., rest → increase_intensity without intermediate stabilization):

$$\text{IncoherenceRate} = \frac{1}{T-1} \sum_{t=2}^{T} \mathbb{1}\left[(a_{t-1}, a_t) \in \mathcal{I}\right].$$

This metric captures breakdowns in sequential behavioral coherence.

C. Cumulative Safety Model

To evaluate trajectory-level implications, action sequences are embedded in a workload accumulation model. Let L_t denote the latent workload state at window t, initialized as $L_0 = 0$. The state evolves as:

$$L_{t+1} = L_t + \delta(a_t),$$

where $\delta(\text{increase_intensity}) = 1.0$, $\delta(\text{maintain}) = -0.2$, $\delta(\text{decrease_intensity}) = -0.2$, $\delta(\text{technique_focus}) = -0.2$, $\delta(\text{rest}) = -0.2$, and L has a floor of zero. A safety violation occurs when:

$$L_t > L_{\max}.$$

From this model we compute:

- **ViolationRate:** Proportion of replicates in which L_t exceeds $L_{\max}$,
- **Time-to-Violation:** Mean window index of first threshold crossing,
- **NetLoadDrift:** $L_T - L_0$.

Importantly, cumulative workload state is *not* included in prompt construction. All safety metrics are computed offline from logged action sequences to isolate the intrinsic stability of the LLM decision layer under degraded telemetry.

VI. Results

This section reports behavioral stability of the LLM decision layer under graded telemetry degradation. We analyze (i) temporal decision volatility (FlipRate), (ii) directional drift (EscalationRate), and (iii) safety-style outcomes from the cumulative workload model (ViolationRate). All values reflect mean performance across 10 replicates per condition.

TABLE I
SUMMARY OF EXPERIMENTAL CONDITIONS

Experiment	Regimes	Degradation	Temperature	Prompt	Rep.	Primary Metric
Dropout severity sweep	R1, R2, R3	$D \in \{0.00, 0.10, 0.20, 0.30, 0.40, 0.60\}$	$T = 0.2$	P0	10	FlipRate, ViolationRate
Downsampling severity sweep	R1, R2, R3	$K \in \{1, 2, 3, 5, 10\}$	$T = 0.2$	P0	10	FlipRate, ViolationRate
Dropout vs. downsampling comparison	R1	D matched to $\approx$ equal loss; $K \in \{1, 2, 3, 5, 10\}$	$T = 0.2$	P0	10	FlipRate
Dropout–downsampling interaction	R1, R3	$D \in \{0, 0.2, 0.4\} \times K \in \{1, 3, 5\}$	$T = 0.2$	P0	10	FlipRate (heatmap)
Temperature sensitivity	R1	$D \in \{0, 0.4\}$	$T \in \{0.0, 0.2, 0.5\}$	P0	10	FlipRate, ΔFlipRate
Prompt mitigation	R1	$D \in \{0.00, 0.10, 0.20, 0.30, 0.40, 0.60\}$	$T = 0.2$	P0 vs. P1	10	FlipRate, Δ_{mit}
Reliability signalling	R1	None (structural)	$T = 0.2$	P0 + delay metadata	10	FlipRate vs. structural loss

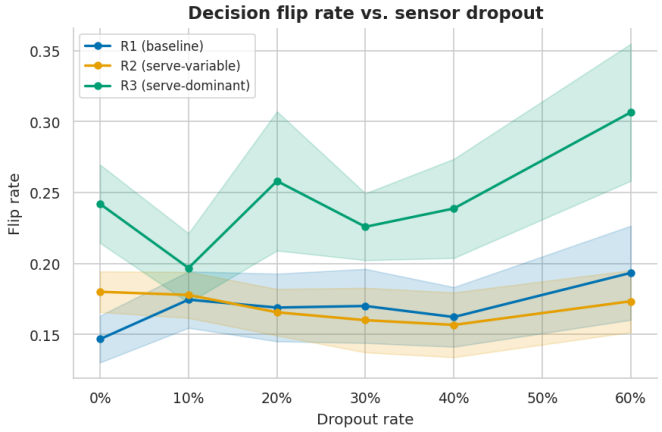


Fig. 2. FlipRate versus dropout level for regimes R1, R2, and R3.

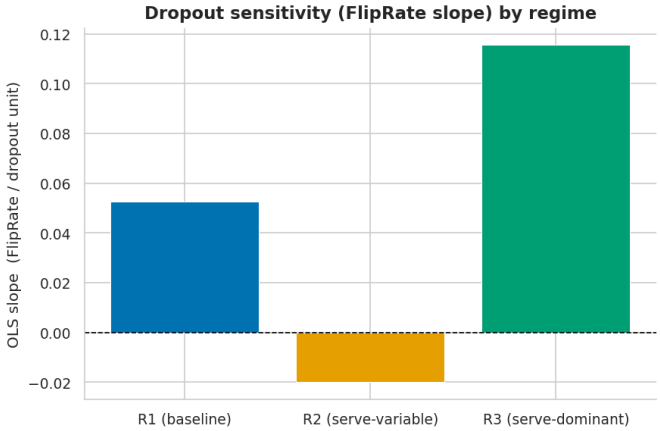


Fig. 3. Estimated slope of FlipRate versus dropout level by regime. Higher values indicate greater sensitivity to increasing dropout.

A. Dropout Severity Sweep

We perform a graded dropout sweep $D \in \{0.00, 0.10, 0.20, 0.30, 0.40, 0.60\}$ applied at the stroke level prior to window aggregation.

FlipRate increases with dropout in a regime-dependent manner: Figure 2 shows FlipRate versus dropout level across regimes R1 balanced, R2 variance-perturbed, R3 serve-dominant). Baseline volatility under clean telemetry ($D = 0$) is non-zero in all regimes. FlipRate generally increases with dropout severity, and the serve-dominant regime (R3) exhibits both the highest baseline FlipRate and the strongest growth under degradation. To quantify degradation sensitivity, we compute the linear slope $\partial\text{FlipRate}/\partial D$ for each regime.

As shown in Figure 3, R3 exhibits the largest positive slope, R1 moderate sensitivity, and R2 near-flat behavior.

Safety metrics remain inactive under current bounds: Figures 4 and 5 show ViolationRate and EscalationRate versus dropout. Both remain zero across all dropout levels and regimes under the current workload bound and horizon.

Consequently, the FlipRate–ViolationRate scatter collapses onto the zero-violation axis (see Figure 6).

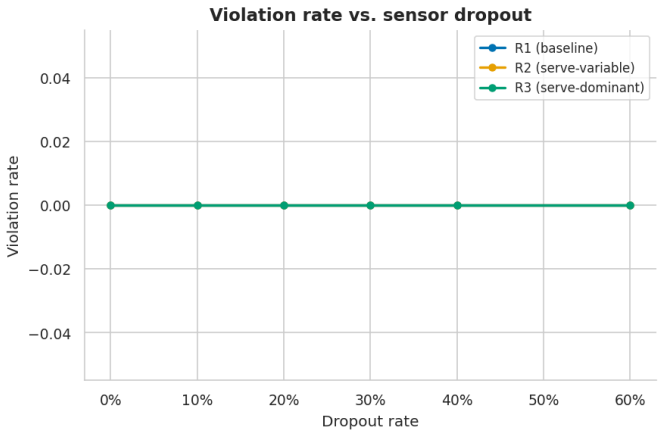


Fig. 4. ViolationRate versus dropout level across regimes. Under the current workload bound, violations are not observed across the sweep.

B. Downsampling Severity Sweep

We replace fixed "keep every 5th stroke" downsampling with graded factors $K \in \{1, 2, 3, 5, 10\}$.

Downsampling produces stronger destabilization than dropout: Figure 7 shows that FlipRate rises sharply with K, particularly in R3. At $K = 10$, instability is substantially higher than in the dropout sweep, indicating that structured temporal thinning is more destabilizing than random deletion.

ViolationRate again remains zero across all downsampling

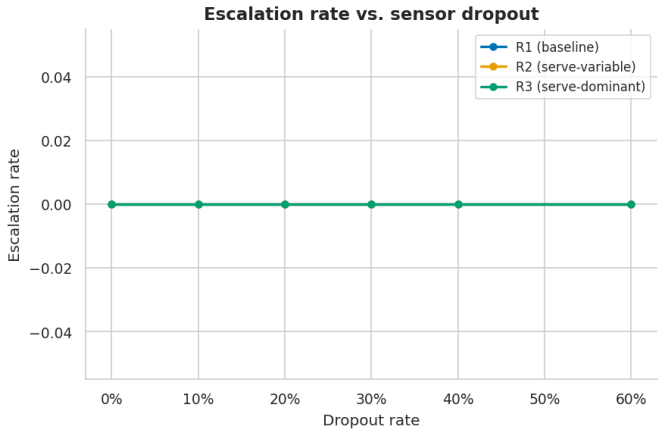


Fig. 5. EscalationRate versus dropout level across regimes. Under the current configuration, escalation remains inactive across the sweep.

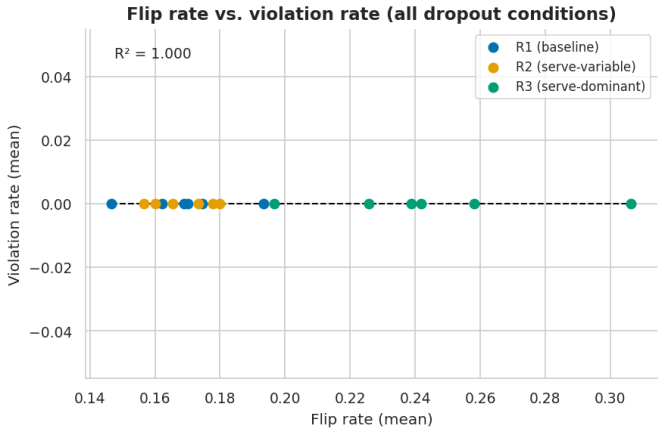


Fig. 6. FlipRate versus ViolationRate across regime–dropout configurations. The $R^2 = 1.000$ annotation reflects a degenerate constant fit and carries no inferential meaning.

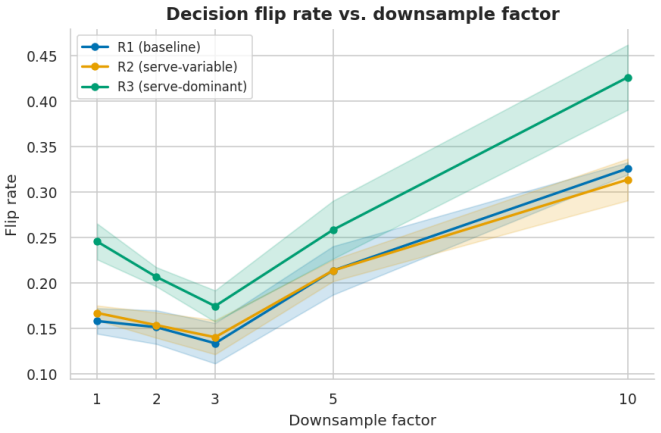


Fig. 7. FlipRate versus downsampling factor (K) across regimes. Downsampling induces larger increases in FlipRate than dropout at comparable nominal information loss.

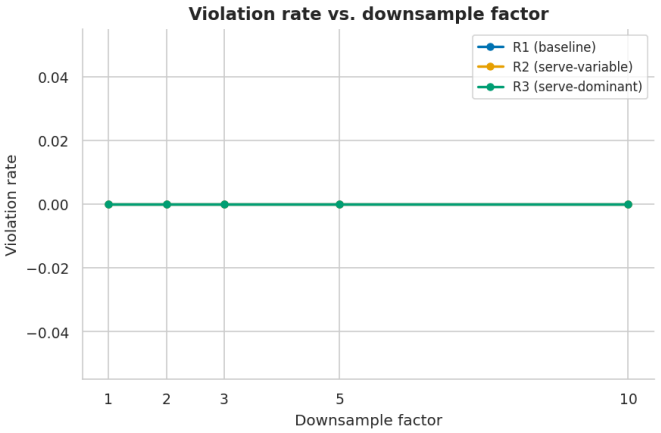


Fig. 8. ViolationRate versus downsampling factor across regimes. No violations are observed under the current bound.

factors (see Figure 8).

Dropout vs. downsampling comparison: Figure 9 overlays dropout and downsampling under approximately matched nominal information loss in the balanced regime (R1). Downsampling produces substantially larger increases in FlipRate than dropout, suggesting that the *structure* of missingness (regular subsampling vs. random dropout) is a primary driver of volatility.

C. Dropout–Downsampling Interaction

To test compositional effects, we evaluate a small grid $D \in \{0, 0.2, 0.4\}$ and $K \in \{1, 3, 5\}$ in R1 and R3.

Figure 10 indicate that the combined degradation conditions show no consistent superadditivity. Residuals from a purely additive prediction are mixed in sign across the tested combinations: some conditions exceed additive expectations and some fall below them. Dropout and downsampling therefore appear to act largely independently at the levels tested, with no evidence of systematic compounding in the high-stress corner of the parameter grid.

D. Temperature Sensitivity Control

To separate degradation sensitivity from decoding stochasticity, we sweep temperature $T \in \{0.0, 0.2, 0.5\}$ under R1 for dropout levels $D \in \{0, 0.4\}$.

Figure 11 shows that higher temperature increases baseline FlipRate and alters degradation response.

We summarize the degradation delta $\Delta = \text{FlipRate}(0.4) - \text{FlipRate}(0.0)$ on Figure 12.

Figure 12 indicates that degradation sensitivity is largest at $T = 0.5$, consistent with an interaction between telemetry loss and decoding stochasticity.

E. Prompt Mitigation Under Severity

We compare prompt variants P0 (baseline) and P1 (structured reliability framing) across the dropout sweep in R1.

Figure 13 shows $\text{FlipRate}_{P1} < \text{FlipRate}_{P0}$ across all dropout levels. ViolationRate remains zero for both prompts.

We quantify mitigation as $\Delta_{\text{mit}}(D) = \text{FlipRate}_{P0} - \text{FlipRate}_{P1}$ on Figure 15.

As shown in Figure 15, mitigation increases with dropout severity and peaks around $D = 0.4$, indicating that structured

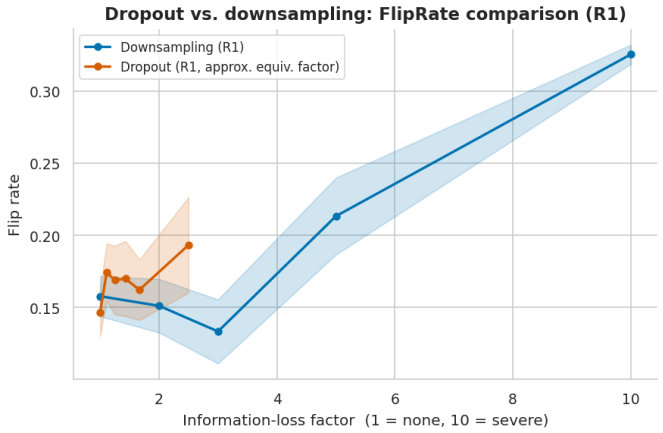


Fig. 9. FlipRate under dropout versus downsampling in R1 plotted against an approximate information-loss factor. Downsampling induces larger instability than dropout at comparable nominal loss.

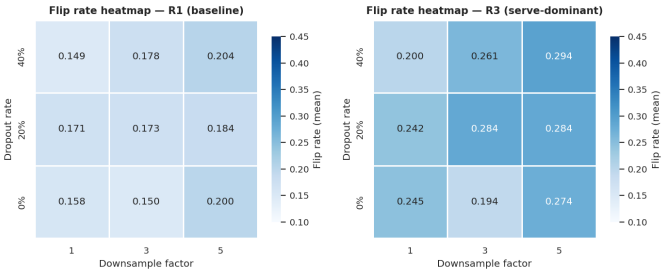


Fig. 10. FlipRate heatmap over the dropout–downsampling interaction grid for R1 and R3.

prompting yields larger stabilization benefits under harsher telemetry loss.

F. Summary of Empirical Findings

Across the expanded sweeps and interaction tests, we observe:

- Baseline FlipRate is non-zero under clean telemetry (Fig. 2).
- Dropout increases FlipRate in a regime-dependent manner, with strongest sensitivity in the dominant regime R3 (Figs. 2–3).
- Downsampling induces substantially larger instability than dropout at comparable nominal information loss (Figs. 7 and 9).
- Dropout and downsampling interact weakly, with mixed-sign residuals and no robust superadditivity (Fig. 10).
- Degradation sensitivity increases with decoding temperature, indicating a degradation×stochasticity interaction (Figs. 11–12).
- Structured prompting (P1) reduces FlipRate across the full dropout sweep, with mitigation strongest at higher dropout (Figs. 13 and 15).
- Under the current workload bound and horizon, ViolationRate and EscalationRate remain inactive (Figs. 4, 5, 8, 14), so observable degradation effects manifest primarily as local decision volatility rather than cumulative safety violations.

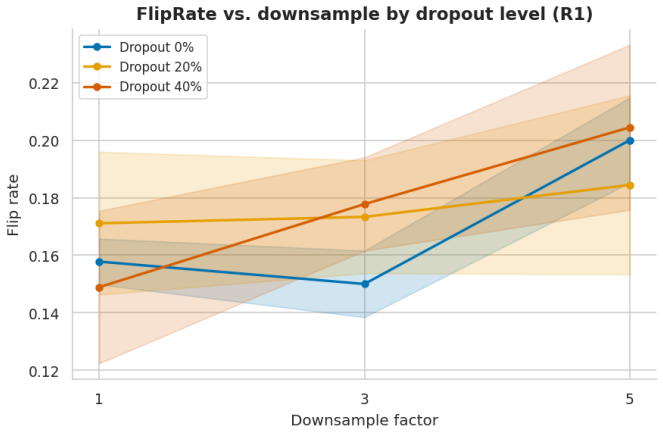


Fig. 11. FlipRate versus dropout in R1 at different decoding temperatures. Higher temperature increases baseline volatility and amplifies degradation sensitivity.

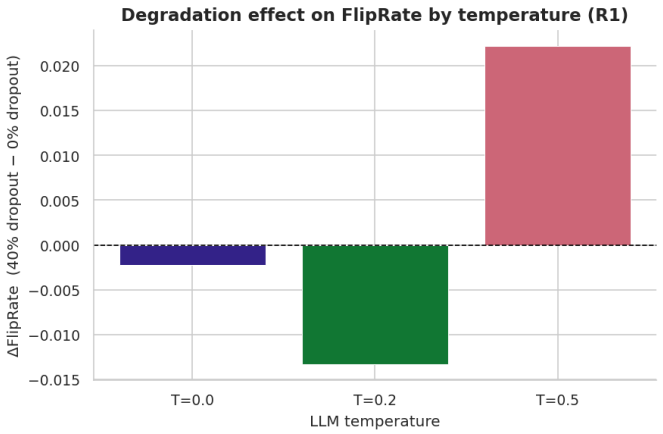


Fig. 12. Degradation effect size in R1 as a function of temperature, measured as FlipRate(D=0.4) − FlipRate(D=0.0).

VII. Discussion

A. Baseline Volatility as an Intrinsic Property

A central observation of the experiment set is that FlipRate is non-zero even under clean telemetry ($D = 0$, $K = 1$). This indicates that the LLM decision layer behaves as a stochastic policy even in the absence of transport degradation.

Degradation therefore does not introduce instability into an otherwise stable controller; rather, it amplifies an already-present baseline volatility.

This distinction is important. The relevant systems question thus how much does telemetry loss amplify intrinsic policy variability and under what structural conditions.

B. The Local–Global Stability Distinction

The results can be interpreted through a unifying lens: the distinction between *local stability* and *global stability* in LLM-integrated sequential systems.

Local stability refers to properties of individual outputs:

- Outputs remain syntactically coherent.
- Individual recommendations remain semantically plausible.

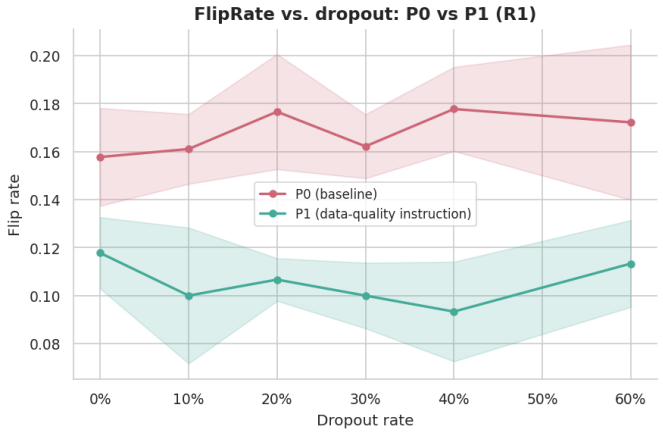


Fig. 13. FlipRate versus dropout in R1 for baseline prompt (P0) and structured prompt (P1). Structured prompting reduces FlipRate across the sweep.

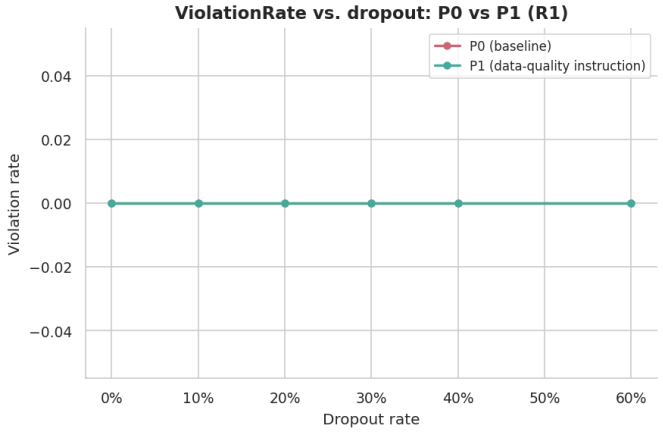


Fig. 14. ViolationRate versus dropout in R1 for prompt variants P0 and P1. No violations occur under the current workload bound.

- JSON schemas remain valid.

In our experiments, local stability is largely preserved even under substantial telemetry degradation. Catastrophic failures (e.g., malformed output, nonsensical recommendations) are rare. This is consistent with the established finding that LLMs are robust to surface perturbations [16].

Global stability refers to properties of the decision trajectory:

- Consecutive recommendations remain consistent given similar inputs.
- Policy behavior is reproducible under repeated evaluation.
- Degradation does not alter the characteristic action distribution.

At the global level, a different picture emerges. FlipRate increases, trajectory-level behavior diverges across replicates, and equivalent degradation conditions produce distinct action sequences. These effects emerge *despite* local outputs remaining individually plausible.

The central observation is therefore: **local coherence can mask global instability.** Instability does not manifest as overt semantic failure; it appears through divergence of sequential behavior over time. A system may appear to function correctly

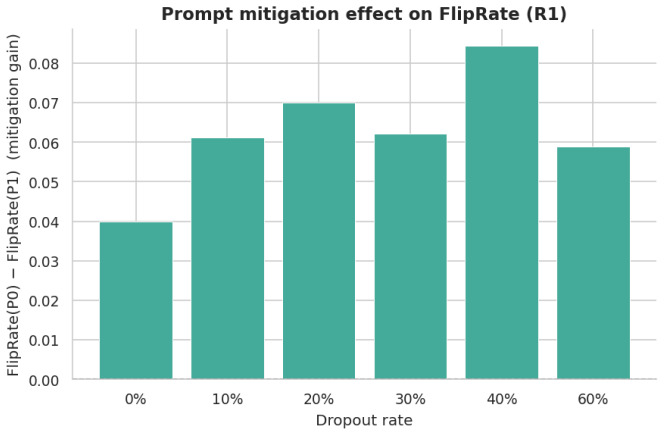


Fig. 15. Prompt mitigation effect across dropout severity in R1. Larger values indicate greater FlipRate reduction from structured prompting (P1) relative to baseline (P0).

when evaluated one decision at a time, yet exhibit clinically or operationally meaningful volatility when evaluated as a controller.

As such, we argue that *locally coherent decisions do not necessarily compose into a globally coherent sequential policy* under perturbation, and characterizes the structural conditions (regime dominance, degradation type, decoding temperature) that govern how large the local–global gap becomes.

C. Regime-Dependent Sensitivity and Dominance Amplification

The graded sweeps reveal that degradation sensitivity is strongly regime-dependent. In the serve-dominant regime (R3), FlipRate increases more sharply with both dropout and downsampling than in balanced (R1) or variance-perturbed (R2) regimes. The slope comparison across regimes makes this difference explicit.

This pattern suggests a dominance-amplification effect: when decision boundaries rely heavily on a small subset of features, reducing telemetry resolution disproportionately perturbs those boundaries. In balanced regimes, information loss is partially buffered by feature diversity; in dominant regimes, the same loss produces larger decision shifts.

Thus, robustness is not a property of the model alone but of the interaction between model policy and feature dominance structure.

D. Structure of Missingness Matters

The comparison between dropout and downsampling provides an additional nuance. Downsampling induces substantially larger increases in FlipRate than random dropout at comparable nominal information loss.

This indicates that the temporal structure of missingness is critical. Regular subsampling distorts window-level summaries in a systematic way, whereas random dropout perturbs them more diffusely. The LLM's decision instability is therefore sensitive not just to how much information is removed, but to how it is removed.

In IoT systems, this implies that packet thinning or resolution reduction may have qualitatively different behavioral consequences than sporadic loss.

E. Interaction in Degradation Space

The dropout–downsampling interaction grid shows that degradation effects are not strictly additive. Increasing dropout has a weak, mixed impact when downsampling is already present, and vice versa.

This interaction indicates that telemetry degradation defines a multi-dimensional stress space rather than a single scalar severity axis. Robustness evaluation must therefore consider combined stressors; testing isolated degradations may underestimate worst-case behavioral instability.

F. Degradation–Stochasticity Interaction

The temperature sweep introduces another layer of nuance. Higher decoding temperature increases baseline FlipRate and amplifies degradation sensitivity. At low temperature, degradation effects are attenuated; at higher temperature, they are magnified.

This interaction demonstrates that instability arises from the joint effect of input uncertainty (telemetry degradation) and policy stochasticity (sampling temperature). Robust deployment cannot treat sensing reliability and decoding configuration as independent design axes.

G. Prompt-Level Mitigation Scales with Severity

Structured prompting consistently reduces FlipRate across dropout levels in the balanced regime. Importantly, the mitigation effect grows with degradation severity, peaking at moderate-to-high dropout levels.

This indicates that interface-level constraints can partially stabilize decision behavior under stress. However, the severity curve is not flattened: FlipRate still increases with dropout under structured prompting. Prompt engineering therefore acts as a damping layer rather than a corrective mechanism. This characterization of prompt constraints as damping rather than corrective is consistent with findings in the robustness literature. Qiang et al. [29] show that fine-tuned LLMs degrade under input perturbations such as paraphrasing and synonym variation, and propose a consistency regularization approach that partially recovers performance. However, recovery remains incomplete, indicating persistent sensitivity to input variation. This mirrors the results reported here, where structured prompting reduces FlipRate but does not restore clean-telemetry stability, suggesting that interface-level interventions attenuate but do not eliminate degradation effects.

H. Local Volatility Without Safety Activation

ViolationRate remains zero across all conditions in this study. This outcome is primarily a structural consequence of the open-loop evaluation design: because each LLM call is stateless, the model has no memory of prior recommendations and no mechanism to sustain directional escalation across windows. Crucially, this same oscillatory behavior suppresses safety violations: triggering a violation requires a sustained run of `increase_intensity` decisions, and a stateless model that keeps reversing course will not produce one. There is therefore an inherent tension in the current results: the same instability that FlipRate measures acts as a structural damper on the accumulation pathway.

The confirmed accumulation parameters are: $\delta(increase_intensity) = +1.0$; $\delta(all_other_actions) = -0.2$ with floor at zero. Under this asymmetric mapping, violations require a strongly directional action sequence, a pattern that stateless stochastic decoding will not reliably produce over the evaluated horizon.

This does not constitute evidence of safety under deployment conditions. In a closed-loop system where prior recommendations influence athlete state, or where prompts include session history, degradation-induced biases would compound over time rather than oscillate. The floor at $L = 0$ additionally introduces a ratchet effect under any positive directional bias. The safety model in this study establishes the structural pathway from decision volatility to workload accumulation; calibrating violation probability under realistic deployment conditions requires closed-loop evaluation over multi-session horizons and is a priority for future work.

I. Contribution

Taken together, these results characterize system-level stability:

- Baseline policy volatility exists even under clean telemetry.
- Degradation amplifies this volatility in a regime-dependent manner.
- Structured temporal loss (downsampling) is more destabilizing than random dropout.
- Degradation effects are approximately additive across stressor combinations, with no robust superadditivity observed in the tested parameter grid.
- Decoding stochasticity modulates degradation sensitivity.
- Prompt constraints mitigate but do not eliminate instability.

Quantification results prove the local (HL1-HL3) and global (HG1-HG3) hypothesis proposed in **Section II-F.** Robustness of LLM-driven IoT decision systems depend jointly on sensing fidelity, feature dominance structure, decoding configuration, and interface design. Stability cannot be assessed by evaluating the model in isolation under clean inputs.

J. Limitations

Several limitations remain. First, telemetry is synthetically generated, enabling controlled regime manipulation but not capturing the full diversity of real-world sensor artifacts. Second, safety evaluation uses an open-loop workload model with a fixed threshold. Alternative bounds or longer horizons may reveal stronger coupling between local volatility and cumulative risk. Third, this study evaluates a single model (LLaMA-3.3-70b-versatile via Groq API). Absolute quantitative findings; i.e., FlipRate magnitudes, degradation slopes,

mitigation effect sizes, and temperature sensitivity curves, are properties of this specific model under the evaluated interface and should not be extrapolated without replication. However, the model was selected as a strong baseline: it represents current state-of-the-art performance among open-source instruction-following models, and observed instability therefore constitutes a lower bound on what would be expected from comparable architectures. Furthermore, the qualitative failure modes documented here: input-induced output volatility, non-linear degradation thresholds, stressor interaction effects, are consistent with the established prompt robustness literature (see Section II), which documents LLM sensitivity to input perturbation across model families. The structural findings of this study (regime-dependent sensitivity, dropout–downsampling asymmetry, weak stressor interaction effects) are argued to generalize because they follow from the interaction between information loss mechanisms and window-level aggregation, not from model-specific behaviour. Replication across model families (e.g., GPT-4o, Gemini, Mistral) remains an important direction for future work and would confirm or refine this partition.

K. Implications for System Design

The findings translate into six actionable design principles for practitioners deploying LLMs as decision layers over IoT telemetry.

1. Test under degraded sensing, not just clean inputs. Clean-data evaluation underestimates real-world behavioral volatility. Testing should include graded dropout, systematic downsampling, and reliability signaling as minimum conditions.

2. Characterize regime structure before deployment. Identify which features drive the LLM's decisions in your domain. If a small subset of features dominates (analogous to serve statistics in tennis), those features are high-risk, their degradation will disproportionately amplify instability.

3. Treat downsampling and dropout as distinct failure modes. Regular temporal thinning (e.g., sensor resolution reduction) has qualitatively different consequences than sporadic packet loss. They require separate robustness evaluation, not a single information-loss metric.

4. Co-design sensing fidelity and decoding temperature. Higher decoding temperature amplifies degradation sensitivity. For safety-critical applications, lower temperature ($T \leq 0.2$) should be used and validated jointly with the expected sensing degradation range.

5. Design for trajectory stability, not just output correctness. In multi-session or longitudinal deployments (e.g., digital health monitoring, dementia care), single-output evaluation is insufficient. Monitor sequential consistency metrics (FlipRate-equivalent) in production alongside accuracy metrics.

VIII. Conclusion

This study characterizes the behavioral stability of a large language model deployed as a decision layer within an IoT telemetry pipeline under graded degradation. Rather than evaluating only coarse stress conditions, we examine structured severity sweeps in dropout and downsampling, interaction effects, decoding temperature sensitivity, and prompt-level mitigation. Several core findings emerge.

First, baseline decision volatility is non-zero even under clean telemetry. Degradation therefore amplifies intrinsic policy variability rather than introducing instability into an otherwise stable controller.

Second, degradation sensitivity is regime-dependent. In dominance-structured regimes, where decisions rely heavily on a small subset of features, FlipRate increases more sharply with telemetry loss. Robustness is thus jointly determined by model policy and feature dominance structure.

Third, the structure of missingness matters. Regular downsampling induces substantially larger instability than random dropout at comparable nominal information loss, indicating that temporal thinning perturbs decision stability more strongly than diffuse deletion.

Fourth, degradation effects do not compound systematically across stress dimensions. The dropout–downsampling interaction grid shows mixed-sign residuals with no robust superadditivity.

Fifth, decoding stochasticity modulates degradation sensitivity. Higher temperature increases baseline volatility and amplifies degradation-induced effects, demonstrating that sensing reliability and sampling configuration interact at the systems level.

Structured prompting reduces FlipRate across degradation levels and provides larger stabilization benefits under higher dropout. However, mitigation does not eliminate severity dependence, indicating that interface constraints act as damping layers rather than corrective mechanisms.

Importantly, under the present workload bound and evaluation horizon, cumulative safety violations remain absent despite measurable increases in local decision volatility. This outcome reflects both the conservative safety bound and the open-loop evaluation design. Each telemetry window is evaluated independently by the LLM without access to prior decisions or the full telemetry history, and safety metrics are computed post hoc from the resulting action sequence. Consequently, the experiment characterizes volatility in the decision layer rather than the closed-loop evolution of the athlete–coach system. The results therefore reveal a layered stability profile in which syntactic coherence is preserved while sequential decision variability increases systematically with telemetry degradation, highlighting that rising policy volatility can occur even when safety thresholds are not activated within the evaluated horizon.

Taken together, these findings indicate that LLM-based decision layers integrated into IoT systems cannot be evaluated solely under clean-data assumptions. Behavioral stability emerges from the interaction among sensing fidelity, feature structure, decoding configuration, and interface design. As LLMs increasingly function as interpreters of longitudinal telemetry in human-facing systems such as digital health monitoring, rehabilitation support, performance analytics, and AI-assisted coaching, understanding how sensing reliability shapes sequential decision behavior becomes essential for ensuring stable and trustworthy recommendations.

In summary, this investigation introduces a framework to evaluate behavioral stability of LLM decision systems transforming telemetry to semantic output within an IoT-style pipeline. We address the real-world issue of stability under stream degradation common to all IoT wireless interfaces. A distinguishing factor in this study is the shift from subjective semantic correctness or static prompt robustness to empirical quantification of the impact of structural information on sequential decision behavior when transforming streaming data into actionable output. This foundation provides a new basis on which to use LLM models to provide consistent output when transforming telemetry data into human-actionable output which is particularly applicable in human-interface applications such as athletics and digital health. Our current work involves implementing the architecture to fuse heterogeneous sensor streams in living laboratory environments (e.g. at the UKDRI-CR&T) for system evaluation across all four degradation types evaluated here (WiFi packet loss, sensor duty cycling, synchronization drift). The combination of high deployment stakes (safety-critical care guidance) and unreliable sensing makes trajectory-level stability evaluation a prerequisite, not an add-on. This basis opens avenues for implementation of medical feedback language models into clinical care pipelines. Future stability analysis should examine closed-loop architectures, tighter or adaptive safety bounds, real sensor deployments, and decision layers that explicitly couple uncertainty estimation with safety enforcement.

ACKNOWLEDGMENT

We express our warmest gratitude to the participants who contributed their time and allowance for data collection to this study.

REFERENCES

[1] C. Siska, K. Marazopoulou, M. Ailem, and J. Bono, "Examining the robustness of LLM evaluation to the distributional assumptions of benchmarks," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 10 406–10 421. [Online]. Available: https://aclanthology.org/2024.acl-long.560/

[2] M. T. R. Laskar, S. Alqahtani, M. S. Bari, M. Rahman, M. A. M. Khan, H. Khan, I. Jahan, A. Bhuiyan, C. W. Tan, M. R. Parvez, E. Hoque, S. Joty, and J. Huang, "A systematic survey and critical review on evaluating large language models: Challenges, limitations, and recommendations," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 13 785–13 816. [Online]. Available: https://aclanthology.org/2024.emnlp-main.764/

[3] N. Nguyen, "I tested an ai fitness coach who wouldn't accept any of my excuses," *The Wall Street Journal*, January 4 2026, accessed: March 4, 2026. [Online]. Available: https://www.wsj.com/tech/personal-tech/ai-fitness-coach-1ca345ec

[4] A.-I. Serban, S. Horrocks, C. Walsh, M. Harrison, E. Soreq, A. Marzuki, M. Woodbridge, R. Nilforooshan, R. A. Calvo, and D. J. Sharp, "Monitoring night-time safety in households with dementia using ambient sensing technologies: a rule-based decision algorithm," *Alzheimer's & Dementia*, vol. 21, no. S9, p. e110614, 2025. [Online]. Available: https://alz-journals.onlinelibrary.wiley.com/doi/abs/10.1002/alz70863_110614

[5] O. A. Shvetsova and S.-K. Lee, "Living labs in university-industry cooperation as a part of innovation ecosystem: Case study of south korea," *Sustainability*, vol. 13, no. 11, 2021. [Online]. Available: https://www.mdpi.com/2071-1050/13/11/5793

[6] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[7] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013, including Special sections: Cyber-enabled Distributed Computing for Ubiquitous Cloud and Network Services Cloud Computing and Scientific Applications — Big Data, Scalable Analytics, and Beyond. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X13000241

[8] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, M. Yan, and A. Zeng, "Do as i can, not as i say: Grounding language in robotic affordances," 2022. [Online]. Available: https://arxiv.org/abs/2204.01691

[9] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," 2022. [Online]. Available: https://arxiv.org/abs/2209.07753

[10] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence, "Palm-e: An embodied multimodal language model," 2023. [Online]. Available: https://arxiv.org/abs/2303.03378

[11] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar, "Voyager: An open-ended embodied agent with large language models," 2023. [Online]. Available: https://arxiv.org/abs/2305.16291

[12] K. Khatiwada, J. Hopper, J. Cheatham, A. Joshi, and S. Baidya, "Large language models in the iot ecosystem – a survey on security challenges and applications," 2025. [Online]. Available: https://arxiv.org/abs/2505.17586

[13] Y. Zheng, Y. Chen, B. Qian, X. Shi, Y. Shu, and J. Chen, "A review on edge large language models: Design, execution, and applications," 2024, accepted by ACM Computing Surveys (CSUR) per arXiv record. [Online]. Available: https://arxiv.org/abs/2410.11845

[14] A. Khan, M. Z. Khan, A. Zainab, S. Jamshed, S. Ahmad, K. Khatib, F. Bibi, and A. Rehman, "Advances in llms with focus on reasoning, adaptability, efficiency and ethics," 2026. [Online]. Available: https://arxiv.org/abs/2506.12365

[15] Y. Elazar, N. Kassner, S. Ravfogel, A. Ravichander, E. Hovy, H. Schütze, and Y. Goldberg, "Measuring and improving consistency in pretrained language models," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1012–1031, 2021. [Online]. Available: https://aclanthology.org/2021.tacl-1.60/

[16] R. Stureborg, D. Alikaniotis, and Y. Suhara, "Large language models are inconsistent and biased evaluators," 2024. [Online]. Available: https://arxiv.org/abs/2405.01724

[17] A. Webson and E. Pavlick, "Do prompt-based models really understand the meaning of their prompts?" 2022. [Online]. Available: https://arxiv.org/abs/2109.01247

[18] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," 2023. [Online]. Available: https://arxiv.org/abs/2203.11171

[19] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. V. Dillon, B. Lakshminarayanan, and J. Snoek, "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," 2019. [Online]. Available: https://arxiv.org/abs/1906.02530

[20] A. K. Shukla, "Large language model evaluation in 2025: Smarter metrics that separate hype from trust," *TechRxiv*, 2025. [Online]. Available: https://www.techrxiv.org/doi/abs/10.36227/techrxiv.175021940.06537275/v1

[21] K. Zhu, J. Wang, J. Zhou, Z. Wang, H. Chen, Y. Wang, L. Yang, W. Ye, Y. Zhang, N. Z. Gong, and X. Xie, "Promptrobust: Towards evaluating the robustness of large language models on adversarial prompts," 2024. [Online]. Available: https://arxiv.org/abs/2306.04528

[22] O. Shvetsova, D. Katalshov, and S.-K. Lee, "Innovative guardrails for generative ai: Designing an intelligent filter for safe and responsible llm deployment," *Applied Sciences*, vol. 15, no. 13, p. 7298, 2025. [Online]. Available: https://www.mdpi.com/2076-3417/15/13/7298

[23] T. An, Y. Zhou, H. Zou, and J. Yang, "Iot-llm: A framework for enhancing large language model reasoning from real-world sensor data," *Patterns*, vol. 7, no. 1, p. 101429, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666389925002776
[24] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
[25] G. Qu, Q. Chen, W. Wei, Z. Lin, X. Chen, and K. Huang, "Mobile edge intelligence for large language models: A contemporary survey," 2025. [Online]. Available: https://arxiv.org/abs/2407.18921
[26] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018.
[27] Y. Miyaoka and M. Inoue, "Control barrier function for aligning large language models," *IEEE Transactions on Control Systems Technology*, pp. 1–10, 2026. [Online]. Available: http://dx.doi.org/10.1109/TCST.2026.3675329
[28] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," in *International Conference on Learning Representations (ICLR)*, 2023, available at OpenReview. [Online]. Available: https://openreview.net/forum?id=WE_vluYUL-X
[29] Y. Qiang, S. Nandi, N. Mehrabi, G. Ver Steeg, A. Kumar, A. Rumshisky, and A. Galstyan, "Prompt perturbation consistency learning for robust language models," in *Findings of the Association for Computational Linguistics: EACL 2024*. St. Julian's, Malta: Association for Computational Linguistics, 2024, pp. 1357–1370. [Online]. Available: https://aclanthology.org/2024.findings-eacl.91/

APPENDIX A
SYNTHETIC TELEMETRY GENERATION AND FEATURE SCHEMA

A. Overview

All experiments use synthetic stroke-level telemetry designed to preserve realistic match structure while enabling controlled distributional manipulations. Each dataset corresponds to a simulated set comprising an ordered sequence of strokes. For each stroke t, the generator outputs (i) a stroke label, (ii) a 40-dimensional motor feature vector derived from inertial signals, (iii) match-context variables, and (iv) a binary success label used for window-level outcome summaries.

The experimental stressors in the main paper (dropout, downsampling) are applied *at evaluation time* on the stroke stream prior to window aggregation; they do not modify the underlying generated ground-truth sequences.

B. Motor Feature Schema (IMU-Derived Summaries)

Motor telemetry is represented as a fixed vector $\mathbf{x}_t \in \mathbb{R}^{40}$ composed of accelerometer- and gyroscope-derived summary statistics computed per stroke. Features are grouped by signal channel (axis and magnitude) and by statistic.

The `simpson_`$\star$ terms represent a Simpson-rule numerical integral of the corresponding signal over the stroke duration, used as an energy/intensity proxy. The magnitude channels (`accMag`, `gyroMag`) summarize the Euclidean norm of tri-axial signals.

C. Match Context Variables

In addition to motor features, each stroke carries match context variables sufficient to construct window-level summaries and strategic regimes. Table III lists the match context fields used in the generated datasets.

TABLE II
MOTOR FEATURE FAMILIES (40 DIMENSIONS) DERIVED PER STROKE FROM INERTIAL SIGNALS.

Family	Variables (per stroke)
Acceleration axes	`mean_accX`, `std_accX`, `max_accX`, `min_accX`, `simpson_accX`
	`mean_accY`, `std_accY`, `max_accY`, `min_accY`, `simpson_accY`
	`mean_accZ`, `std_accZ`, `max_accZ`, `min_accZ`, `simpson_accZ`
Acceleration magnitude	`mean_accMag`, `std_accMag`, `max_accMag`, `min_accMag`, `simpson_accMag`
Gyroscope axes	`mean_gyroX`, `std_gyroX`, `max_gyroX`, `min_gyroX`, `simpson_gyroX`
	`mean_gyroY`, `std_gyroY`, `max_gyroY`, `min_gyroY`, `simpson_gyroY`
	`mean_gyroZ`, `std_gyroZ`, `max_gyroZ`, `min_gyroZ`, `simpson_gyroZ`
Gyroscope magnitude	`mean_gyroMag`, `std_gyroMag`, `max_gyroMag`, `min_gyroMag`, `simpson_gyroMag`

TABLE III
MATCH-CONTEXT VARIABLES EMITTED PER STROKE AND USED FOR STATE/CONTEXT SUMMARIES.

Category	Variables
Indexing	`set_index`, `game_index`, `point_index`, `stroke_index_in_point`, `row_index`, `isoTime`
Score state	`sets_you`, `sets_opp`, `games_you`, `games_opp`, `points_you_raw`, `points_opp_raw`, `point_score`
Outcomes	`point_outcome` (point-level result), `label` (binary success label used in evaluation summaries)

D. Generative Process and Regime Construction

The generator produces a temporally coherent sequence of points. Serves initiate points, rally strokes follow a simple alternating structure, and rally length is sampled from a regime-dependent distribution (shorter for serve-dominant regimes, longer for defensive/slice-heavy regimes). Each stroke is then assigned motor features by sampling from a stroke-conditioned distribution. Regimes are implemented by manipulating (i) stroke mixture/temporal structure, (ii) motor feature distribution parameters, and/or (iii) the outcome coupling to specific stroke types.

The synthetic telemetry is designed to be *behaviorally realistic* rather than signal-accurate. Match structure (rally lengths, stroke distributions, scoring dynamics) follows probabilistic models grounded in published tennis biomechanics literature. Motor feature distributions are calibrated to produce kinematic statistics broadly consistent with published wearable IMU studies of racket sports. The key validity requirement is not signal fidelity to a specific athlete but behavioral validity: that regimes differ in feature dominance structure in ways that are representative of the structural variation present in real telemetry. This is verified by confirming that regime-conditioned action distributions differ significantly under clean telemetry, establishing that regime labels carry behavioral meaning for the LLM decision layer.

Baseline (Scenario 2 Baseline). The baseline regime defines stroke-conditioned motor distributions

$$\mathbf{x}_t \sim \mathcal{N}(\boldsymbol{\mu}_{s_t}, \Sigma_{s_t}),$$

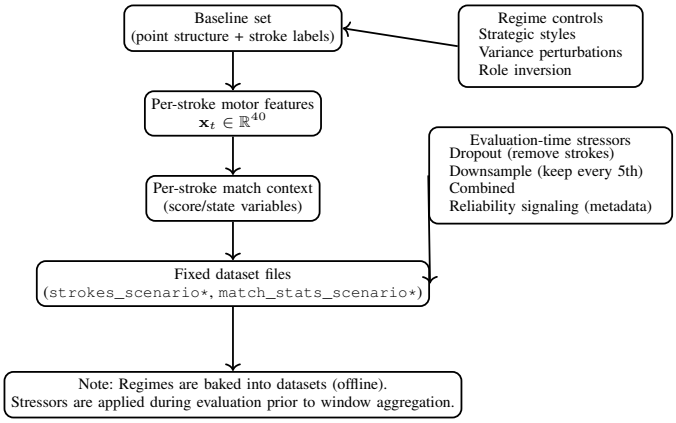


Fig. 16. Schematic of the synthetic telemetry generation process...

with stable parameters across the set and without a single stroke type dominating success patterns.

Strategic styles (Scenario 1 family). Strategic style regimes modify the *dominant contributor* to successful points by altering stroke mixture patterns and coupling success to subsets of motor features associated with specific stroke types: balanced all-court, serve-dominant, defensive slicer (long rallies), backhand weapon, forehand weapon, and inconsistent server. These regimes preserve point structure while changing which stroke types and motor patterns are preferentially associated with success.

Variance perturbations (Scenario 2 BackhandVar / ServeVar). These regimes preserve the baseline set structure and mean motor patterns but selectively inflate variance for one targeted stroke type:

$$\Sigma'_s = \alpha \Sigma_s, \quad \alpha > 1,$$

applied only when s is the perturbed stroke category (backhand or serve). This isolates sensitivity to increased sensor noise without altering the overall sequence structure.

Role inversion (Scenario 3 family). These regimes preserve the set structure while changing the role of the forehand in point success: forehand-bad, forehand-neutral, and forehand-weapon. Operationally, the generator adjusts the association between forehand-associated motor patterns and stroke outcomes, thereby inducing a controlled shift in feature dominance without changing the feature schema.

E. Schematic of Data Generation and Experimental Manipulations

Figure 16 summarizes the generation pathway and where regime and stress manipulations act.

APPENDIX B
LLM PROMPT SPECIFICATIONS

Prompt 0 (Baseline)

```
You are an expert tennis coach. Given the recent
    telemetry summary and data quality indicators,
    output ONE coaching action and cue.

Telemetry window summary:
[window_summary]

Data quality:
[data_quality]

Game context (raw):
[game_context  only if game_context is non-empty]

Choose action_class from exactly: increase_intensity
    , maintain, decrease_intensity, rest,
    technique_focus.
If data_quality indicates missing or stale data (e.g
    . dropout_rate > 0, staleness_seconds > 0, non-
    empty missing_fields), set
    uncertainty_acknowledged=true and mention it
    briefly in the cue.

Return ONLY valid JSON with no other text:
{"action_class": "<one of increase_intensity,
    maintain, decrease_intensity, rest,
    technique_focus>", "cue": "1-2 sentences", "
    uncertainty_acknowledged": true or false}
```

Prompt 1 (Reliability-Constrained)

```
You are an expert tennis coach. Given the recent
    telemetry summary and data quality indicators,
    output ONE coaching action and cue.

Telemetry window summary:
[window_summary]

Data quality:
[data_quality]

Game context (raw):
[game_context  only if game_context is non-empty]

Choose action_class from exactly: increase_intensity
    , maintain, decrease_intensity, rest,
    technique_focus.
If data_quality indicates missing or stale data (e.g
    . dropout_rate > 0, staleness_seconds > 0, non-
    empty missing_fields), set
    uncertainty_acknowledged=true and mention it
    briefly in the cue.

IMPORTANT: If data_quality shows dropout_rate > 0,
    delay_seconds > 0, staleness_seconds > 0,
    downsample_factor > 1, or non-empty
    missing_fields, you MUST set
    uncertainty_acknowledged=true and briefly
    mention the data issue in your cue.

Return ONLY valid JSON with no other text:
{"action_class": "<one of increase_intensity,
    maintain, decrease_intensity, rest,
    technique_focus>", "cue": "1-2 sentences", "
    uncertainty_acknowledged": true or false}
```