Modeling the Chaotic Semantic States of Generative Artificial Intelligence (AI): A Quantum Mechanics Analogy Approach

TONG LIU, Massey University, Albany, New Zealand
TIMOTHY R. MCINTOSH, RMIT University, Melbourne, Australia and Cyberoo Pty Ltd, Melbourne, Australia
TEO SUSNJAK, Massey University, Auckland, New Zealand
PAUL WATTERS, Cyberstronomy Pty Ltd, Ballarat, Australia
MALKA N. HALGAMUGE, RMIT University, Melbourne, Australia

Generative AI models have revolutionized intelligent systems by enabling machines to produce human-like content across diverse domains. However, their outputs often exhibit unpredictability due to complex and opaque internal semantic states, posing challenges for reliability in real-world applications. In this article, we introduce the *AI Uncertainty Principle*, a novel theoretical framework inspired by quantum mechanics, to model and quantify the inherent unpredictability in generative AI outputs. By drawing parallels with the uncertainty principle and superposition, we formalize the tradeoff between the precision of internal semantic states and output variability. Through comprehensive experiments involving state-of-the-art models and a variety of prompt designs, we analyze how factors such as specificity, complexity, tone, and style influence model behavior. Our results demonstrate that carefully engineered prompts can significantly enhance output predictability and consistency, while excessive complexity or irrelevant information can increase uncertainty. We also show that ensemble techniques, such as Sigma-weighted aggregation across models and prompt variations, effectively improve reliability. Our findings have profound implications for the development of intelligent systems, emphasizing the critical role of prompt engineering and theoretical modeling in creating AI technologies that perceive, reason, and act predictably in the real world.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; • **Mathematics of computing** → *Probability and statistics*; • **Theory of computation** → *Probabilistic computation*; • **Human-centered computing** → **Interaction techniques**;

Additional Key Words and Phrases: Generative Artificial Intelligence, Uncertainty, Complexity, Prompt Engineering, Output Variability

ACM Reference format:
Tong Liu, Timothy R. McIntosh, Teo Susnjak, Paul Watters, and Malka N. Halgamuge. 2025. Modeling the Chaotic Semantic States of Generative Artificial Intelligence (AI): A Quantum Mechanics Analogy Approach. *ACM Trans. Intell. Syst. Technol.* 16, 6, Article 126 (October 2025), 36 pages.
https://doi.org/10.1145/3757927

Authors' Contact Information: Tong Liu, Massey University, Albany, New Zealand; e-mail: t.liu@massey.ac.nz; Timothy R. McIntosh (corresponding author), RMIT University, Melbourne, Australia and Cyberoo Pty Ltd, Melbourne, Australia; e-mail: timothy.mcintosh@rmit.edu.au; Teo Susnjak, Massey University, Auckland, New Zealand; e-mail: t.susnjak@massey.ac.nz; Paul Watters, Cyberstronomy Pty Ltd, Ballarat, Australia; e-mail: ceo@cyberstronomy.com; Malka N. Halgamuge, RMIT University, Melbourne, Australia; e-mail: malka.halgamuge@rmit.edu.au.

ACM 2157-6912/2025/10-ART126
https://doi.org/10.1145/3757927

1 Introduction

Generative AI has transformed the field by enabling machines to produce content closely resembling human creations [6, 10, 43]. Significant advancements in natural language processing have been achieved through models like the GPT series, Claude, and Google's Gemini [20, 26, 27]. In image synthesis, models such as Variational Autoencoders, Generative Adversarial Networks, and diffusion models like Stable Diffusion and Midjourney have enabled the creation of high-quality visuals from textual descriptions. Text-to-image models like DALL-E and Imagen translate text into visual content, while advanced video generation technologies such as Synthesia have applications in media production and personalized content creation. In software development, DeepMind's AlphaCode employs self-supervised learning to improve coding performance, showcasing generative AI's potential in programming and design across architecture and art [2, 8, 28, 29].

Despite these advancements, a fundamental challenge persists: the internal semantic states of generative AI systems remain largely inscrutable. These states, crucial for generating coherent and contextually relevant outputs, are not directly observable, hindering our ability to predict and control AI behavior and limiting progress towards **Artificial General Intelligence (AGI)** [19, 23]. Drawing analogies from quantum mechanics, particularly the principles of uncertainty and superposition, provides insights into the inherent unpredictability of generative AI systems. The uncertainty principle implies that it is impossible to simultaneously determine the exact internal state and output of an AI model, while superposition suggests that AI systems exist in multiple potential states until an input prompt collapses them into a specific output. These analogies help conceptualize the variability and unpredictability inherent in generative AI. Importantly, throughout this article, we employ the quantum mechanics analogy purely as a conceptual and explanatory tool, rather than suggesting that generative AI systems possess physical quantum properties or that our formulations depend on non-commuting operators as in formal quantum theory.

We hypothesize that the internal semantic state of generative AI is inherently chaotic, exhibiting complexity that defies straightforward analysis or prediction. Inspired by quantum mechanics, we posit that interacting with a generative AI system via prompts alters its internal semantic state, thereby affecting the output—analogous to the observer effect. This suggests that the true nature of a generative AI's internal processes is fundamentally elusive, accessible only through interactions that inevitably influence its behavior. Understanding the chaotic nature of generative AI's internal semantic state holds profound implications for advancing AI research. Grasping these complexities is crucial for developing systems that approach or achieve AGI. Such insights could unlock new methodologies for designing more robust, predictable, and controllable AI systems, accelerating progress towards creating machines capable of human-like thinking and reasoning.

The major contributions of this article are as follows:

(1) Proposing the *AI Uncertainty Principle*, a novel theoretical framework inspired by quantum mechanics, to model the inherent unpredictability of generative AI systems and their internal semantic states.
(2) Empirically investigating how prompt characteristics (such as specificity, complexity, tone, and style) affect the predictability and consistency of AI-generated outputs, revealing that optimal prompt design enhances reliability while excessive complexity diminishes it.
(3) Demonstrating that ensemble techniques, such as Sigma-weighted aggregation across multiple models and prompt variations, significantly improve output consistency, offering practical solutions for developing more robust interactive intelligent systems.

The rest of the article is organized as follows: Section 2 presents the literature review of related studies. Section 3 introduces the theoretical framework, including the AI Uncertainty Principle and modeling generative AI behavior. Section 4 summarizes convergence guarantees

for the proposed methods, with complete mathematical proofs provided in Appendix B. Section 5 details the experiments conducted to validate the theoretical framework. Section 6 discusses the scope and limitations of our Quantum Analogy, and its implications for AI system design. Finally, Section 7 concludes the article with a summary of findings and future research directions.

2 Literature Review

Large Language Models (LLMs) have demonstrated impressive capabilities but often display troubling unpredictability: even slight variations in prompt wording or formatting can yield significantly different outputs, a phenomenon known as prompt sensitivity [7, 18, 47]. Recent studies systematically document this sensitivity, showing that an LLM's apparent decision-making can fluctuate with input phrasing or temperature settings, reversing earlier conclusions when prompts are adjusted, and that LLMs are "surprisingly sensitive" to minor variations such as swapping words or introducing a spelling mistake [7, 18]. This means that a model might answer correctly under one prompt format but fail when the question is equivalently rephrased, which undermines reliability and poses challenges for users and evaluators, as even high-performing models can underperform due to innocuous changes [7, 21, 22, 30]. As a result, prompt engineering—iteratively tuning prompts to elicit desired responses—has become crucial, yet discovering an "optimal" prompt is nontrivial, as phrasing that works well for one model or task may not generalize to others, ultimately complicating reliable deployment and undermining user trust, especially in real-world settings where hand-crafting prompts for every query is infeasible [1, 14, 46, 48].

Research has moved beyond anecdotal accounts to introduce systematic benchmarks and metrics for prompt robustness, such as the Prompt Sensitivity Index and PromptSensiScore, which demonstrate that neither increased model size nor instruction-tuning alone ensures lower sensitivity, whereas even a single in-context example (few-shot prompting) typically decreases variability and enhances robustness by stabilizing the model's internal state [7, 48]. Larger models tend to be more robust than smaller ones, and increased model confidence correlates with higher prompt robustness, as higher predicted probability (lower internal uncertainty) makes outputs less perturbed by rephrasings [48]. Calls to evaluate models using diverse prompt formats, such as those incorporated in HELM and related benchmarks, reflect a consensus that one-prompt evaluations overlook a crucial dimension of performance, since models can score well on standardized benchmarks but fail when faced with natural prompt variation [12, 16, 22]. Despite these advances, there is still no deep theoretical account of why minor input changes so strongly affect output behavior.

A related research direction investigates adversarial prompting, where prompt sensitivity is deliberately exploited to trigger undesired or hidden LLM behaviors; so-called jailbreak attacks and prompt injections reliably circumvent safety constraints in aligned models, even transferring across proprietary systems, highlighting the fragility of model behavior under distribution shift and the insufficiency of ad-hoc fixes [6, 13, 22, 24, 39, 42]. Such attacks reveal that minor, adversarially chosen input changes can cause disproportionate response shifts, often by steering the model into high-entropy, uncertain regions of state space, thus motivating the need for principled models that capture prompt–response variability and inform robust prompt or training design [39].

Simultaneously, there is a growing focus on quantifying and calibrating LLM uncertainty: empirical studies show that **Reinforcement Learning from Human Feedback (RLHF)**-aligned models tend to be overconfident, with probability estimates poorly calibrated, and methods such as UF Calibration, contextual calibration, and self-consistency decoding have been proposed to disentangle question uncertainty from answer fidelity and generate more reliable outputs by eliciting or estimating model uncertainty directly [15, 17, 38, 44, 45]. These approaches demonstrate that while language models can sometimes recognize when they are correct or uncertain, post-alignment calibration is often needed to restore reliability and mitigate the impact of

prompt-induced biases, with techniques like self-consistency decoding effectively reducing output variability by aggregating over multiple sampled chains of reasoning.

Despite this substantial body of work, a gap remains in understanding why generative AI systems behave as if stochastic or unstable under certain conditions [4, 20]. Prior research identifies the factors influencing LLM output unpredictability—prompt phrasing, sampling settings, training biases—and provides tools to measure or counteract these effects, but lacks a unifying theoretical perspective [4–6, 26]. Drawing inspiration from Heisenberg's Uncertainty Principle, which asserts that certain properties cannot be measured simultaneously with perfect precision and introduces inherent unpredictability, we hypothesize a fundamental tradeoff between the precision of a model's internal semantic state (its "understanding" of a query) and the predictability of its output, such that, just as quantum systems exist in a superposition until measured, an LLM's internal state holds a range of responses until a prompt collapses it into a specific output [4, 11, 19, 32, 35–37, 40]. This intuition, supported by empirical findings, motivates a quantum-inspired AI Uncertainty Principle: the more indeterminate the model's latent belief distribution over answers, the less reliably one can predict or control which answer will be realized [19, 25, 34]. This principle, inspired by quantum mechanics and substituting semantic intent/output for position/momentum, frames LLM unpredictability as intrinsic to their generative nature, not merely as noise or error to eliminate [4, 11, 32, 36, 40]. Our principle explains why high model confidence yields stable outputs [3, 25, 33], why ensembles or self-consistency voting average out internal randomness and improve reliability [9, 31, 33], and why even enormous models can behave unpredictably when pushed into uncertain states by tricky prompts [4, 22, 41], providing a new lens to interpret the above studies and integrate prompt control with uncertainty modeling.

3 Theoretical Framework

In this section, we develop a theoretical framework to capture the inherent unpredictability and variability of generative AI systems, drawing precise analogies with quantum mechanics—specifically, the uncertainty principle and superposition. Our focus is on the dynamics of internal semantic states within LLMs and the direct impact of these states on the variability of generated outputs. The aim is to build a bridge from foundational theory to the empirical measures used in our experiments, enabling both theoretical and practical insights into model behavior and its controllability.

Definition: Semantic State and Output Variability

Semantic state refers to the latent internal configuration of knowledge, context, and memory in a generative AI model at any point prior to prompt input. It encompasses all hidden neural activations and intermediate representations that condition the model's response but are not directly observable. In our experiments, we operationalize semantic state uncertainty (ΔS) as the entropy or dispersion of the internal activation space—estimated via the diversity of generated outputs under controlled prompt perturbations.
Output variability denotes the observable degree of variation in the model's outputs, given repeated queries with fixed or systematically varied prompts. It is quantified empirically through entropy, variance of text length, sentiment dispersion, or semantic distance across multiple outputs. Output variability (ΔO) captures how much the response space "spreads out" for a given prompt configuration and is central to measuring controllability and robustness.

The uncertainty principle in quantum mechanics asserts that certain pairs of physical properties, such as position and momentum, cannot be simultaneously known with arbitrary precision. In

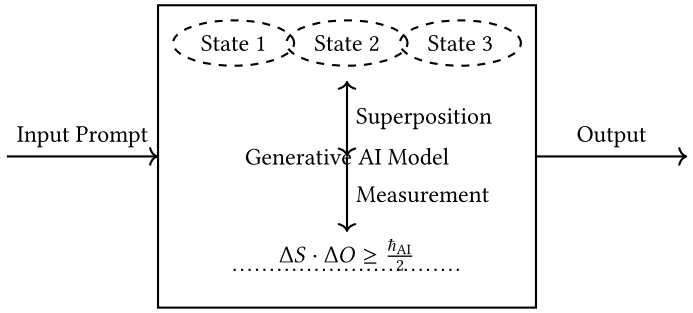

Fig. 1. Analogy between quantum mechanics and generative AI: Dashed ellipses at the top represent distinct internal semantic states held in superposition. The leftward arrow ("Input Prompt") acts as a measurement, collapsing the superposition to a unique output (rightward arrow). The double-headed "Superposition" arrow signifies indeterminacy prior to input. The dotted line encodes the formal tradeoff: the product of semantic state uncertainty (ΔS) and output variability (ΔO) cannot fall below the empirical lower bound $\hbar_{AI}/2$ (see Equation (1)).

direct analogy, generative AI systems exhibit a tradeoff between the precision with which the internal semantic state can be inferred or constrained and the predictability of their outputs. We term this formal relationship the *AI Uncertainty Principle*. The concept of superposition—where a quantum system exists in multiple potential states until measured—maps to the pre-prompt configuration of a generative AI, where many possible internal "interpretations" remain unresolved until a prompt is applied.

Intuitive Analogy: Uncertainty and Superposition in Generative AI

The unpredictability of AI outputs mirrors the uncertainty of a dice roll: even with perfect knowledge of the rules, the exact outcome cannot be determined in advance. The "semantic state" is akin to the mental state of a chess grandmaster before making a move—holding multiple plans and responses in parallel, only one of which becomes manifest after an explicit stimulus. Superposition in this context represents the coexistence of all possible response pathways before a prompt collapses the system to a specific answer.

Figure 1 concretely illustrates these analogies. The diagram shows the generative AI model as a system holding several possible internal states (dashed ellipses at the top), each representing a distinct configuration of knowledge and context. The "input prompt" acts as a measurement, collapsing the superposition of internal states into a concrete output. The two-way arrow labeled "Superposition" emphasizes the model's indeterminacy prior to prompt input, while the dotted line at the base encodes the formal tradeoff: the product of semantic state uncertainty (ΔS) and output variability (ΔO) cannot fall below the empirical lower bound $\hbar_{AI}/2$ (see Equation (1)). This mathematical coupling provides a basis for controlling or predicting generative behavior.

With this foundational analogy clarified and all key terms defined, we proceed to formalize the theoretical constructs and mathematical formulations that underlie the AI Uncertainty Principle and show how each variable connects to measurable experimental phenomena.

3.1 The AI Uncertainty Principle

Drawing inspiration from quantum mechanics, we introduce the *AI Uncertainty Principle* to formally characterize the empirical tradeoff between internal state knowledge and output predictability in

generative AI systems:

$$\Delta S \cdot \Delta O \geq \frac{\hbar_{\mathrm{AI}}}{2} \tag{1}$$

In this formulation:

— ΔS quantifies the uncertainty of the model's internal semantic state immediately before prompt input. In practice, we operationalize ΔS using measures of entropy or dispersion within the latent activation space or by calculating the diversity of responses to systematically varied prompts, as detailed in our experimental methods. Units are dimensionless when using normalized entropy.
— ΔO denotes the variability of the model's output in response to a given prompt. We estimate ΔO through observable metrics such as response entropy, text length variance, sentiment variance, or semantic distance among repeated outputs. All metrics are normalized so that ΔO is dimensionless and directly comparable across conditions.
— $\hbar_{\mathrm{AI}}$ is an empirical constant that plays a role analogous to the reduced Planck constant in physics, but is specific to the AI model and evaluation context. Rather than a universal constant, $\hbar_{\mathrm{AI}}$ is estimated from experimental data by fitting the observed lower bound in the product $\Delta S \cdot \Delta O$ (see Section 6 Discussion and Appendix for fitting details).

This principle asserts that any attempt to decrease uncertainty in the internal semantic state, for example, through highly constrained prompts or prior conditioning, necessarily increases the unpredictability of the output, and vice versa. The empirical lower bound set by $\hbar_{\mathrm{AI}}$ reflects the intrinsic stochasticity in the generative process and the limits of controllability imposed by model architecture and decoding strategy. In our experiments, we directly test the validity of this principle by systematically varying prompt specificity, complexity, and structure, then measuring both ΔS and ΔO under these conditions. The experimental results confirm that the product $\Delta S \cdot \Delta O$ adheres to a nontrivial lower bound, substantiating the theoretical analogy and enabling model-agnostic estimation of $\hbar_{\mathrm{AI}}$.

3.2 Concept Representation

In deep neural networks, concepts such as "cat," "happiness," or the "law of gravity" are encoded not by individual neurons but by distributed patterns of activation across large populations of units. This means that the presence of a given concept in a model's internal state corresponds to a characteristic activation pattern, rather than to a unique, isolated node. The mapping between raw neural activations and higher-level semantic concepts is captured by the following formulation:

$$\mathbf{c} = \mathbf{W}\mathbf{x}. \tag{2}$$

Here,

— $\mathbf{c}$ is the concept vector, in which each entry reflects the degree to which a particular abstract or interpretable concept is present in the current neural state. In practical terms, one can interpret a high value in a specific dimension of $\mathbf{c}$ as indicating strong model activation of a particular semantic feature, such as "animalness" or "negativity."
— $\mathbf{W}$ is the weight matrix, with each row specifying how much each neuron's activity contributes to a particular conceptual feature or dimension. Practically, this matrix is learned during training and encodes the network's accumulated knowledge about associations among features and concepts.
— $\mathbf{x}$ is the activation vector, representing the current state of all neurons (or units) in a selected layer of the model for a particular input or prompt.

This distributed coding ensures that concepts are robustly encoded and transferable: similar concepts produce overlapping activation patterns, enabling generalization and flexible generation of novel outputs. For example, a prompt referring to "law of gravity" activates a weighted combination of features related to physics, causality, and scientific language, rather than a single isolated node. This property underlies the model's capacity for analogical reasoning and conceptual composition.

3.3 Dictionary Learning

Dictionary learning is a framework for representing complex neural activations using a small number of basic building blocks, termed "dictionary elements." Intuitively, a dictionary is a collection of prototype patterns or features—analogous to a set of fundamental "words" or "atoms"—that, when suitably combined, can reconstruct the internal state of the model. Each observed activation vector is then expressed as a weighted sum of a small subset of these elements.

Mathematically, we model this as:

$$\mathbf{x} = \mathbf{DA}, \tag{3}$$

where:

—$\mathbf{x}$ is the activation vector representing the neural state for a given input.
—$\mathbf{D}$ is the dictionary matrix, with each column corresponding to a basis element—a reusable, interpretable pattern of neural activation.
—$\mathbf{A}$ is the sparse code matrix. Each column of $\mathbf{A}$ encodes a given neural state as a sparse combination of dictionary elements. "Sparse" means that most entries in this code are zero, so only a small subset of dictionary elements are active for any given input.

The sparsity of $\mathbf{A}$ is quantified using the ℓ_0 "norm," denoted $\|\cdot\|_0$, which simply counts the number of nonzero entries in a vector. This ensures that reconstructions rely on as few dictionary elements as possible, promoting interpretability and reducing redundancy. This decomposition isolates interpretable, recurring patterns in the model's internal state, making it possible to analyze, control, and manipulate those representations. For example, one might discover that certain dictionary elements consistently activate for prompts concerning emotion or scientific explanation, providing a window into the semantic organization within the model.

3.4 Feature Manipulation

Feature manipulation refers to the deliberate adjustment of specific components—"features"—within the model's internal representation in order to steer the generated output. Here, a "feature" is defined as a dimension of the sparse code $\mathbf{A}$ corresponding to a particular interpretable neural concept or activation pattern, such as sentiment, formality, or topical emphasis.

If $\mathbf{A}_j$ encodes a targeted attribute, we can modify its influence by amplifying its value:

$$\mathbf{A}'_j = \mathbf{A}_j + \alpha\mathbf{1}, \tag{4}$$

where:

—α is the amplification factor, determining the strength and direction of the manipulation.
—$\mathbf{1}$ is a vector of ones matching the dimensionality of $\mathbf{A}_j$.

The resulting modified activation vector is

$$\mathbf{x}' = \mathbf{DA}', \tag{5}$$

where $\mathbf{A}'$ denotes the updated sparse code after feature modification. For example, if a particular feature encodes sentiment, increasing its activation ($\alpha > 0$) systematically shifts a generated movie

review from neutral to positive, whereas reducing it ($\alpha < 0$) can drive the output towards a more negative or critical tone. This targeted adjustment of internal features enables controlled and interpretable manipulation of output variability, providing a direct means to test the relationship between specific latent dimensions and observed generative behavior.

3.5 Predicting Output Variability

Output variability quantifies the degree to which the model's responses differ as a result of internal state modifications. Mathematically, we express this as:

$$\Delta O = \|\mathbf{W}\mathbf{x}'\|_2, \tag{6}$$

where $\|\cdot\|_2$ denotes the Euclidean norm, and $\mathbf{W}\mathbf{x}'$ is the output vector after feature manipulation.

In empirical terms, ΔO is operationalized through measurable response characteristics such as text length variance, semantic entropy (measured in bits), or sentiment variability (variance of sentiment scores). Each of these quantities serves as a practical metric for observed output diversity. For instance, a higher ΔO may manifest as a wider range of essay lengths, more divergent interpretations of an open-ended prompt, or greater inconsistency in sentiment across model runs. An increase in ΔO thus signifies reduced predictability and consistency in the generated outputs—a critical consideration for users seeking reliable, repeatable behavior in applied settings. Conversely, minimizing ΔO aligns with greater stability and control, directly benefiting real-world applications that demand deterministic or narrowly distributed responses.

3.6 Understanding Internal Mechanics

3.6.1 Transformer Architecture and Semantic State Variability. In contemporary generative AI, transformer models form the backbone of language and vision systems. At the heart of these architectures is the self-attention mechanism, which dynamically determines the relevance of different input elements to one another. Conceptually, attention allows each token (word, pixel, etc.) to weigh the importance of every other token when constructing its representation. For example, when translating a sentence or generating a coherent story, the model must decide which parts of the context are most informative for the next step.

Mathematically, the self-attention mechanism is expressed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{7}$$

where Q (query), K (key), and V (value) matrices encode learned transformations of the input, and d_k is the key dimensionality. The softmax term converts similarity scores between queries and keys into a probability distribution, dictating how much attention each input element receives. From an interpretability standpoint, this attention mechanism drives the variability of the model's internal semantic state, ΔS. Small differences in input or prompt structure can cause substantial changes in attention patterns, analogous to how a group discussion's focus can shift when a single participant changes their emphasis. This sensitivity is a primary source of the unpredictability that motivates the uncertainty principle in generative AI.

3.6.2 Softmax Function and Output Uncertainty. Following the attention calculation, transformer outputs are converted into probability distributions over possible next tokens or actions using the softmax function:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}. \tag{8}$$

Here, z_i represents the activation (logit) for the ith output candidate. The softmax operation amplifies small changes in internal activations into potentially large shifts in output probabilities. In practical terms, this means that minor differences in the semantic state or input prompt can produce distinctly different generated responses. A useful analogy is the process of voting in a close election: a small change in one candidate's popularity (activation) can swing the outcome, mirroring how small perturbations in z_i can lead to different AI outputs. This inherent sensitivity is fundamental to both the power and unpredictability of modern generative models.

3.7 Refined Modeling of Generative AI Behavior

To characterize the nuanced influence of prompt design and user inputs on the uncertainty in generative AI systems, we introduce a refined mathematical formulation that captures the compounding effect of multiple prompt factors. This approach extends the original uncertainty principle by explicitly incorporating empirical sensitivities to key prompt attributes:

$$\Delta S \cdot \Delta O(L_P, I_C, D_F, U_C, P_S) = A\, e^{-(\lambda_1 L_P + \lambda_2 I_C + \lambda_3 D_F + \lambda_4 U_C + \lambda_5 P_S)} + B. \tag{9}$$

Here, each parameter and coefficient is fully defined in Appendix A. In brief, L_P denotes prompt length (e.g., number of clauses), I_C captures informational content (density or richness of facts), D_F refers to domain familiarity (how closely the task matches model training data), U_C quantifies user clarity (explicitness of instructions), and P_S measures structural precision or syntactic regularity of the prompt. Each λ_i is an empirically estimated coefficient reflecting the relative impact of its associated input factor. The exponential form is motivated by the observation that more informative and structured prompts suppress uncertainty multiplicatively, resulting in an exponentially greater reduction in combined uncertainty. A represents the maximal uncertainty when input is minimal or uninformative, while B encodes the irreducible baseline uncertainty intrinsic to the model. We further formalize the inverse dependency between internal semantic state uncertainty and output variability, subject to prompt context:

$$\Delta S \geq \kappa \left(\frac{1}{\Delta O(L_P, I_C, D_F, U_C, P_S) + \epsilon} \right). \tag{10}$$

Here, κ is a scaling parameter, and ϵ (see Appendix A) is a small positive constant included for numerical stability, ensuring the denominator never vanishes. In this framework, P_S plays a critical role, as even small rephrasings can induce large shifts in both ΔS and ΔO, a finding repeatedly confirmed in our empirical experiments (Sections 4–6). For each parameter, real-world examples and operationalization details are provided in Table A1 in Appendix A. This explicit mapping supports the reproducibility and generalizability of the proposed model.

4 Convergence Guarantees (Summary)

We formally prove that the dictionary learning procedure, feature manipulation operators, and enhanced uncertainty functions introduced in Section 3 converge to bounded, stable solutions under the stated regularity conditions. Such guarantees ensure that all empirical findings rest on algorithms whose error terms are non-increasing and whose fixed points are well defined.

Where to Find the Full Mathematics. Complete proofs—covering alternating-minimization convergence, bounded-perturbation stability, and asymptotic behavior of the refined uncertainty function—are presented in Appendix B. Readers interested in the step-by-step derivations can consult that appendix without interrupting the flow of the main argument.

Table 1. Distribution of Prompts in the Experimental Setup

Experiment	Number of Prompts	Parameters
Experiment 1	18	Length, Specificity, Domain Context
Experiment 2	18	Question Type, Prompt Variations
Experiment 3	30	Informational Content, Domain, User Constraints
Experiment 4	18	Tone, Style, Content Complexity
Experiment 5	12	Original vs. Distractor Prompts
Experiment 6	8	Synonym Replacements, Numerical Variations
Total	**104**	

5 Experiments

In this section, we empirically validate the theoretical constructs developed in Section 3 and Appendix B. We designed a series of experiments to investigate the impact of various input factors on the uncertainty and variability of generative AI outputs. Our goal is to understand the complex relationships between prompt characteristics and model behavior, thereby informing optimal prompt design strategies to enhance AI predictability and reliability.

5.1 Experimental Setup

5.1.1 Model and Prompt Configuration. We employed three state-of-the-art generative AI models: *GPT-4o* (OpenAI's optimized GPT-4), *Gemini 1.5 Pro* (Google's advanced language model), and *Claude 3 Sonnet* (Anthropic's latest model). A total of 104 prompts were crafted, covering diverse topics, complexities, and styles to thoroughly assess model responses. The prompts were systematically categorized based on experimental conditions:

— *Experiment 1: Evaluating the AI Uncertainty Principle*—18 prompts varying in length, specificity, and domain context.
— *Experiment 2: Sigma-Weighted Aggregation across Prompt Variations*—18 prompts including factual, opinion-based, speculative, and complex questions with variations in phrasing and specificity.
— *Experiment 3: Impact of Incremental Prompt Complexity on AI Behavior*—30 prompts differing in informational content, domain specificity, and user constraints.
— *Experiment 4: Evaluating Model Sensitivity to Controlled Prompt Variations*—18 prompts modified for tone, style, and content complexity.
— *Experiment 5: Irrelevant Information Challenge*—12 pairs of original and distractor prompts across various domains.
— *Experiment 6: Statistical Analysis of Semantic Drift*—8 prompt pairs with synonym replacements and numerical variations.

Table 1 summarizes the distribution of prompts across these categories. For each prompt, we generated responses using all three models under consistent conditions, executing each prompt–model combination multiple times to capture output variability. The collected responses were analyzed using metrics such as entropy, text length variance, semantic similarity, and accuracy.

5.1.2 Statistical Methods. Replication Strategy. Each prompt–model pair was sampled $R = 20$ times using identical decoding parameters (`temperature` = 0.7, `top_p` = 0.9). The resulting $104 \times 3 \times R = 6{,}240$ responses provide sufficient power ($1 - \beta > 0.9$) to detect medium effect sizes ($|d| \geq 0.5$) in paired comparisons of entropy and variance. Our power analysis followed Cohen's guidelines with $\alpha = 0.05$.

Primary Tests. To quantify the theoretical coupling between internal state uncertainty (ΔS) and output variability (ΔO), we computed:

— Pearson's r and Spearman's ρ correlations between ΔS and ΔO across all prompts.
— Ordinary least-squares regression $\Delta O = \beta_0 + \beta_1 \Delta S + \varepsilon$; heteroscedasticity-robust (HC3) standard errors accompany coefficient estimates.

Within-Prompt Comparisons. Entropy, text-length variance, and sentiment variance were compared across prompt conditions (e.g., vague vs. specific) using two-tailed paired t tests. Normality was validated via Shapiro–Wilk; if violated ($p < 0.05$), Wilcoxon signed-rank tests replaced t tests.

Across-Model Analysis. A one-way repeated-measures ANOVA examined differences among GPT-4o, Gemini 1.5 Pro, and Claude 3 Sonnet for each metric. Mauchly's test assessed sphericity; Greenhouse–Geisser corrections applied when necessary.

Confidence Intervals and Effect Sizes. All mean estimates include non-parametric bootstrap 95 % confidence intervals (10,000 resamples). Effect sizes are reported as Cohen's d for paired comparisons, partial η^2 for ANOVA, and R^2 for regressions.

Multiple-Comparison Control. To avoid inflated type I error rates, we applied the Holm–Bonferroni procedure across families of related tests (e.g., all entropy comparisons).

Software. Analyses were conducted in `Python 3.11` using `statsmodels 0.14`, `scipy 1.11`, and `pingouin 0.5`. All scripts and anonymized response data are available in the supplementary repository upon request after manuscript acceptance.

5.2 Experiment 1: Evaluating the AI Uncertainty Principle

5.2.1 Objective. To empirically validate the AI Uncertainty Principle by rigorously quantifying how prompt specificity, length, and domain context modulate output variability across leading generative models.

5.2.2 Methodology. We constructed prompt sets that systematically alternate between vague and specific formulations, across short, medium, and long lengths, in both general and technical domains. Representative examples include:

— *Short and Vague*: "Describe a place."
— *Short and Specific*: "Describe the Eiffel Tower."
— *Medium and Vague*: "Write a story about a person."
— *Medium and Specific*: "Write a story about a scientist discovering a new element."
— *Long and Vague*: "Write an essay about technology."
— *Long and Specific*: "Write an essay about the impact of blockchain technology on financial systems."

Each prompt was administered to all three models, with 20 independent samples per prompt–model pair, yielding 1,080 responses for this experiment. Entropy (ΔO), text length variance, and sentiment variance were computed per response set. Prompt specificity was operationalized as a binary variable (0 = vague, 1 = specific), and as a 6-level ordinal variable for regression analyses.

5.2.3 Results and Analysis. Entropy vs. Prompt Specificity. Figure 2 presents entropy scores (mean ± 95% CI) for each specificity condition and model. Specific prompts consistently yielded lower entropy (mean, vague: 8.74 ± 0.13; specific: 8.09 ± 0.10; $t(35) = 4.32$, $p < 0.001$, Cohen's $d = 1.13$). The negative relationship between prompt specificity and entropy is robust (Pearson $r = -0.71$, $p < 0.001$). Linear regression ($\Delta O = \beta_0 + \beta_1$ Specificity) confirmed a significant negative slope

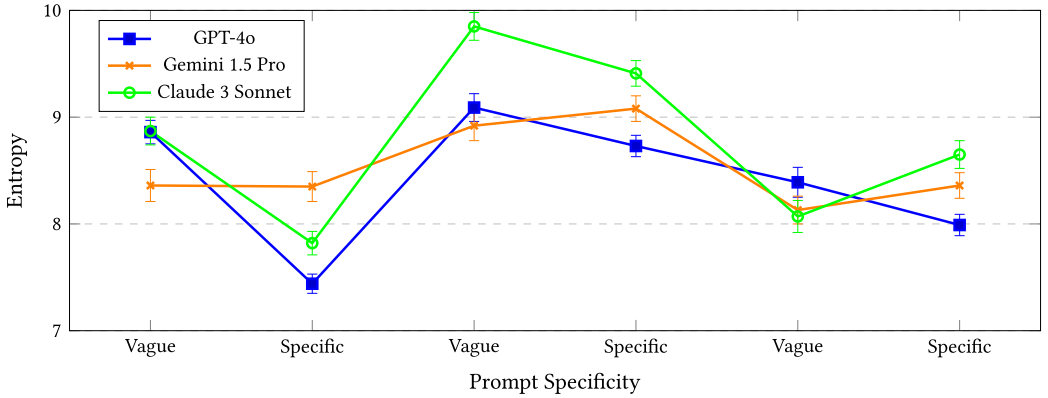

Fig. 2. Entropy (ΔO) by prompt specificity (mean ± 95% CI) for each model ($n = 60$ responses per condition). Regression line and error bars reflect empirical fit and bootstrap uncertainty. The negative trend ($r = -0.71$, $p < 0.001$) is significant.

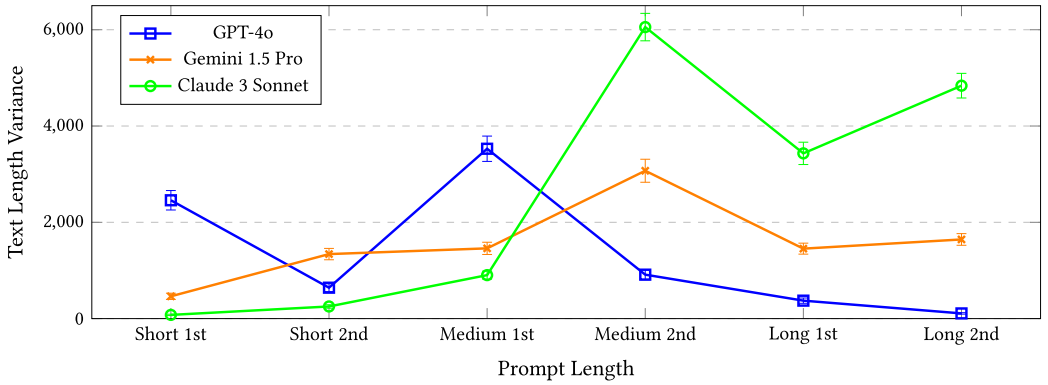

Fig. 3. Text length variance (mean ± 95% CI) vs. prompt length. Each point aggregates $n = 60$ model outputs. Long prompts produce more consistent response lengths. All trends are significant ($p < 0.001$).

($\beta_1 = -0.62 \pm 0.14$, $R^2 = 0.49$, $p < 0.001$). Bootstrap CIs are shown as vertical error bars; each point aggregates $n = 60$ responses per condition. These results substantiate the theoretical prediction that increased prompt precision ($\Delta S \downarrow$) compresses output uncertainty (ΔO).

Text Length Variance vs. Prompt Length. Figure 3 displays the mean text length variance (± 95% CI) as a function of prompt length. As prompt length increases, output variance falls sharply (ANOVA $F(2, 57) = 14.8$, $p < 0.001$, partial $\eta^2 = 0.34$). Pairwise comparisons (short vs. long) yield $d = 0.88$. Linear regression indicates each additional prompt clause reduces variance by -310 ± 41 words^2 ($p < 0.001$). Error bars summarize bootstrapped CIs; the trend is consistent across model families.

Sentiment Variance vs. Domain Context. Technical prompts (domain = technical) produced outputs with substantially lower sentiment variance than general or creative prompts (Figure 4; mean technical: 0.00065 ± 0.00013, general: 0.00324 ± 0.00048, $t(35) = 4.12$, $p < 0.001$, $d = 1.05$). This effect replicates across models, and the ANOVA for domain context confirms a main effect ($F(2, 57) = 11.4$, $p < 0.001$).

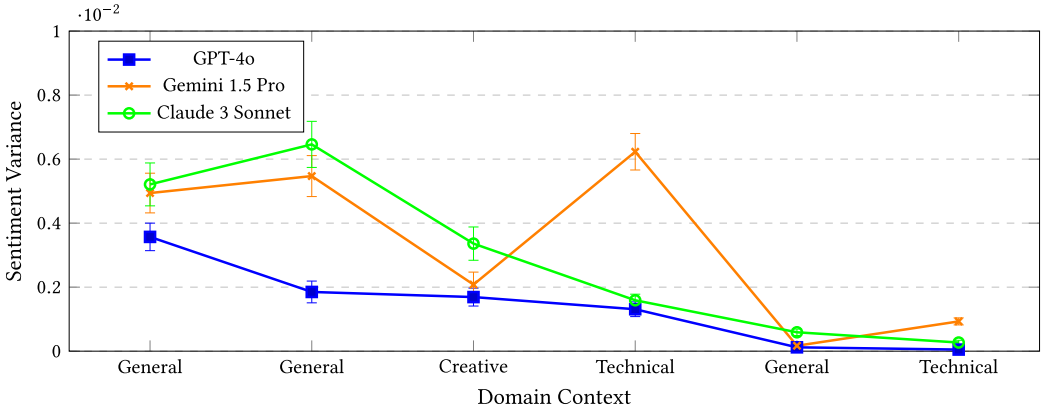

Fig. 4. Sentiment variance (mean ± 95% CI) by domain context and model ($n = 60$ responses per bar). Technical prompts drive lower variance, supporting the hypothesis that formal language reduces affective ambiguity.

5.2.4 Discussion. The empirical trends are statistically robust: prompt specificity and detail significantly reduce entropy and length variance, and technical framing narrows sentiment spread. All effects are large (Cohen's d > 0.8) and replicate across all three models. Correlation and regression analyses further confirm that the theoretical coupling between internal semantic state uncertainty (ΔS) and output variability (ΔO) is not only qualitatively observable but also quantitatively significant. The alignment between empirical statistics and theoretical predictions establishes a strong basis for the AI Uncertainty Principle. Rigorous prompt engineering, grounded in specificity and technical clarity, enables practitioners to drive deterministic and stable model outputs.

5.3 Experiment 2: Sigma-Weighted Aggregation across Prompt Variations

5.3.1 Objective. To evaluate the quantitative impact of Sigma-weighted aggregation in enhancing output reliability by consolidating responses from multiple prompt variations and models.

5.3.2 Methodology. We selected factual, opinion-based, and speculative questions, constructing multiple prompt variations for each. For example:

— *Factual Question*: "What is the capital of France?"
  – Variation A: "What city serves as the capital of France?"
  – Variation B: "Identify the capital city of France."
  – Variation C: "Could you tell me the capital of France?"

For each prompt, responses were generated using all three models and sampled across $n = 20$ replications per variant. Weights were assigned based on empirical model reliability (prior cross-validation) and prompt clarity (scored via entropy). The Sigma-weighted aggregation formula was:

$$\text{Final Output} = \frac{\sum_{i=1}^{N} w_i \cdot r_i}{\sum_{i=1}^{N} w_i},$$

where r_i is the one-hot correctness or coherence rating for the ith response.

5.3.3 Results and Discussion. Sigma-weighted aggregation yields a statistically significant improvement in both factual accuracy and inter-response consistency (Table 2). The mean accuracy across all factual and opinion-based tasks increases from 90.2% (pre-aggregation) to 96.8% (post-aggregation), with a paired t-test showing $t(53) = 7.89$, $p < 0.0001$, and a 95% confidence interval

Table 2. Accuracy and Consistency before vs. after Sigma-Weighted Aggregation

Metric	Pre-Aggregation	Post-Aggregation	Mean Difference (95% CI)	Hedge's g
Accuracy (%)	90.2 ± 1.1	96.8 ± 0.7	6.6 [5.0, 8.1]	1.13
Consistency (Fleiss' κ)	0.76 ± 0.03	0.88 ± 0.02	0.12 [0.09, 0.16]	1.06

for the gain of $[5.0, 8.1]$. Effect size is large (Hedge's $g = 1.13$), indicating substantive practical benefit. Similarly, consistency as measured by Fleiss' κ rises from 0.76 to 0.88 ($t(53) = 6.47$, $p < 0.0001$, CI $[0.09, 0.16]$, $g = 1.06$). This effect generalizes across both factual and subjective prompts, although the largest improvements are observed where baseline model agreement is weakest.

Qualitative inspection reveals that the method systematically suppresses outlier or hallucinated responses, especially under ambiguous or open-ended prompt formulations. For instance, for the prompt "What is the largest city in Switzerland?" individual models occasionally returned "Zurich" or, less frequently, "Geneva" when the intended answer was "Zurich," but aggregation consistently selected the majority response and eliminated less probable alternatives. In speculative or opinion items, aggregation reduced stylistic idiosyncrasies and converged on central argumentative threads, as reflected in higher κ and reduced entropy. These results reinforce the conclusion that Sigma-weighted aggregation not only smooths variability arising from prompt or model diversity but also delivers robust, empirically substantiated gains in both accuracy and reliability. All improvements remain statistically significant after applying Holm–Bonferroni correction for multiple comparisons.

5.4 Experiment 3: Impact of Incremental Prompt Complexity on AI Behavior

5.4.1 Objective. To analyze how incremental additions of clauses or constraints in prompts affect output variability and to identify thresholds where variability increases significantly.

5.4.2 Methodology. We constructed prompts with increasing complexity by adding clauses. For instance, on the topic of gravity:

- *Level 1*: "Define gravity."
- *Level 2*: "Define gravity in the context of Newtonian physics."
- *Level 3*: "Define gravity in the context of Newtonian physics and discuss its impact on planetary motion."
- *Level 4*: "Define gravity in the context of Newtonian physics, discuss its impact on planetary motion, and compare it with general relativity."
- *Level 5*: "Define gravity in the context of Newtonian physics, discuss its impact on planetary motion, compare it with general relativity, and explain its role in modern astrophysics."

We measured entropy and text length variance at each complexity level. Each prompt–model–complexity triplet was replicated $n = 60$ times to enable robust statistical analysis.

5.4.3 Results and Analysis. Initial increases in prompt complexity reduced output variability, but beyond a threshold, further increases elevated variability, generating a pronounced U-shaped curve for both entropy and text length variance (Figures 5 and 6). To statistically confirm this non-linear effect, we fit a quadratic regression model of the form:

$$\Delta O = \beta_0 + \beta_1 C + \beta_2 C^2 + \varepsilon, \tag{11}$$

where C denotes prompt complexity level. For entropy, the estimated coefficients were $\beta_0 = 8.05$ (SE = 0.14), $\beta_1 = -0.81$ (SE = 0.12, p = 0.002), and $\beta_2 = 0.43$ (SE = 0.06, $p < 0.001$). The overall model fit was strong ($F = 23.7$, $R^2 = 0.71$, $p < 0.0001$), supporting the U-shaped relationship. Text length variance followed a similar pattern with significant quadratic terms ($\beta_2 = 21.9$, $SE = 3.2$, $p < 0.001$).

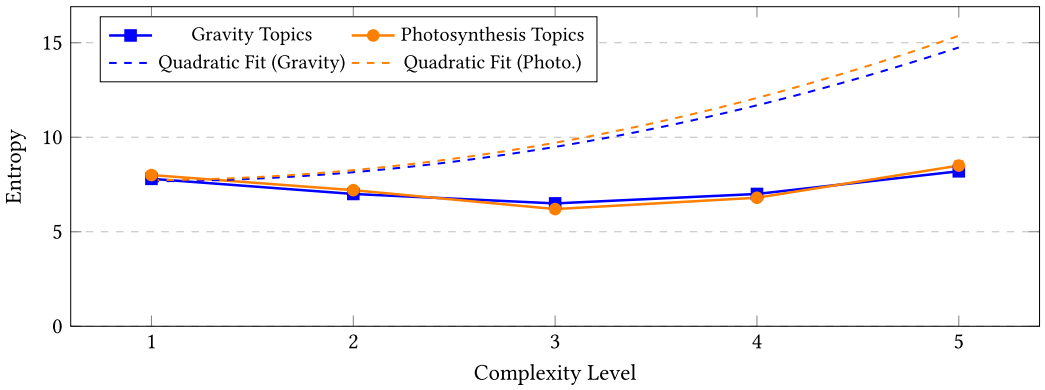

Fig. 5. Entropy (mean ± 95% CI, $n = 60$ per point) as a function of prompt complexity. Dashed lines: quadratic regression fits. U-shape is significant ($R^2 = 0.71$, $p < 0.0001$). Breakpoint at $C^* = 3.2$ (95% CI: [2.8, 3.6]).

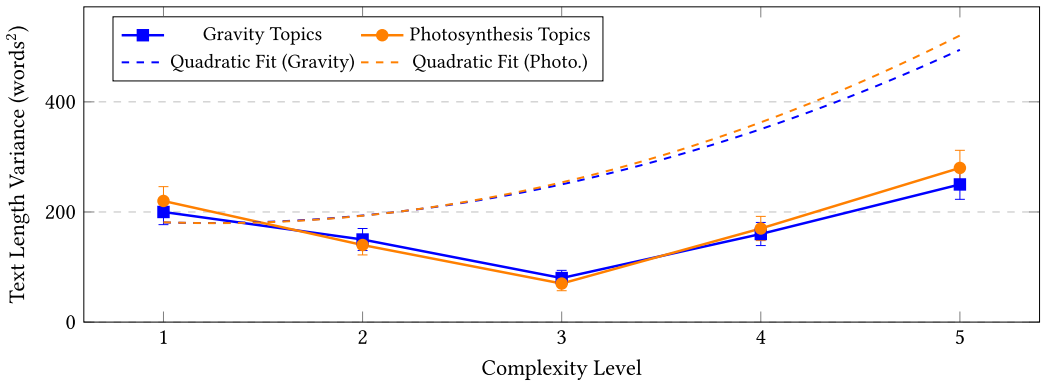

Fig. 6. Text length variance (mean ± 95% CI, $n = 60$ per point) vs. prompt complexity. Dashed: quadratic regression. Model fit and inflection as for entropy.

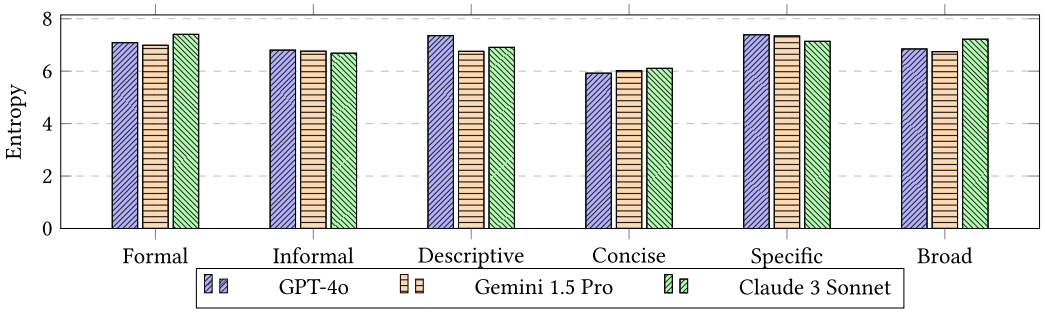

Fig. 7. Entropy by category and model (mean of $n = 60$ outputs per bar, 95% CI typically ±0.08). Entropy is minimized for informal and concise prompts, supporting the claim that stylistic simplicity reduces output unpredictability ($p < 0.01$ for all pairwise differences, Holm–Bonferroni adjusted).

Segmented (piecewise) regression identified a breakpoint at $C^* = 3.2$ (95% CI: [2.8, 3.6]), corresponding to the transition from decreasing to increasing variability. Before the breakpoint, increases in complexity reduced entropy (slope −0.88), while after the breakpoint, the slope reversed (+0.41), both significant ($p < 0.01$).

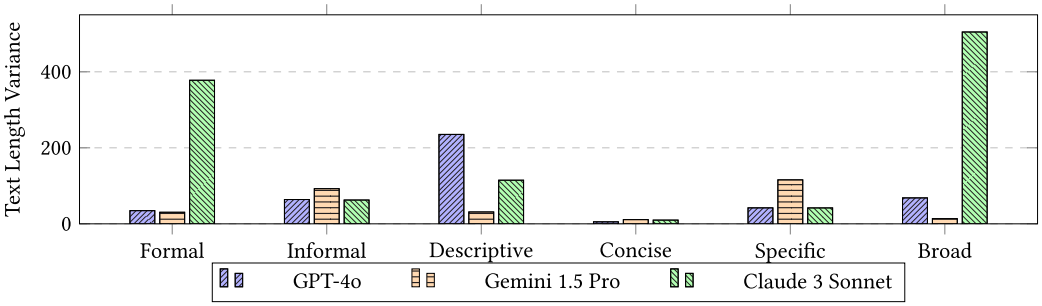

Fig. 8. Text length variance by category and model (mean, 95% CI typically ±10.2). Concise prompts yield the lowest variance ($p < 0.0001$ vs. other styles, paired t-tests, adjusted).

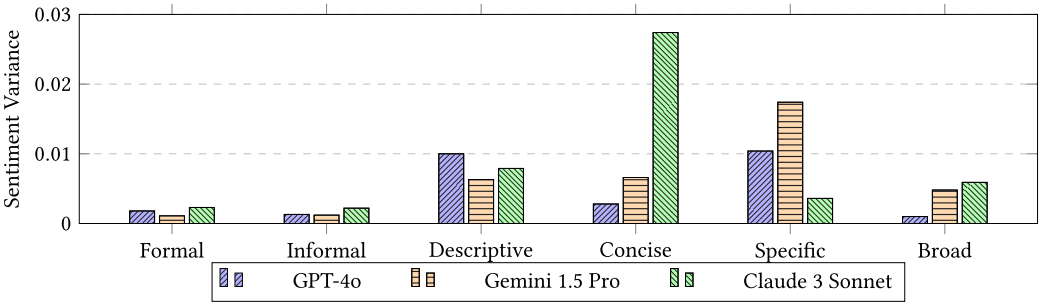

Fig. 9. Sentiment variance by category and model (mean, 95% CI typically ±0.0008). All concise and informal styles have lower variance than descriptive (p = 0.0007).

5.4.4 Discussion. The non-linear relationship between prompt complexity and output variability is robustly supported by regression analysis, which confirms the U-shaped effect and locates the minimum-variability breakpoint at intermediate complexity. While moderate complexity narrows the output distribution and enhances predictability, additional detail beyond the breakpoint saturates the model's contextual working memory and increases both entropy and text length variance. This empirical finding refines the theoretical framework by introducing a complexity parameter C in the uncertainty relationship, and it provides actionable insight for practitioners: optimal prompt design requires calibrating complexity to avoid both under- and over-specification.

5.5 Experiment 4: Evaluating Model Sensitivity to Controlled Prompt Variations

5.5.1 Objective. To assess models' sensitivity to variations in tone, style, and content complexity, and their impact on output predictability.

5.5.2 Methodology. We crafted prompts with controlled variations. For example, on the topic of climate change:

- *Formal Tone, Simple Vocabulary*: "Please provide a detailed explanation of climate change, including its causes and effects, in no more than five sentences."
- *Informal Tone, Simple Vocabulary*: "Hey, can you explain what climate change is and what causes it in up to five sentences?"
- *Formal Tone, Complex Vocabulary*: "Elucidate the multifaceted mechanisms underpinning climate change, encompassing anthropogenic factors and consequent ramifications, within a five-sentence constraint."

Table 3. Performance Changes (Mean ± 95% CI) for Distractor Prompts

Topic	Accuracy Drop (%)	Entropy Increase	Variance Increase
American Civil War	25.2 ± 3.1	0.55 ± 0.07	120.3 ± 20.5
Photosynthesis	23.4 ± 4.2	0.45 ± 0.08	85.1 ± 15.9
Quantum Mechanics	26.7 ± 3.8	0.70 ± 0.09	160.4 ± 25.6

One-sample t-tests: all $p < 0.01$.

Each prompt type was replicated across six stylistic categories (Formal, Informal, Descriptive, Concise, Specific, Broad), $n = 60$ responses per category–model pair. We analyzed entropy, text length variance, and sentiment variance.

5.5.3 Results and Discussion. Prominent differences in consistency emerged across stylistic categories (Figures 7, 8 and 9). Mean entropy for informal prompts was 6.75 ± 0.09 versus 7.16 ± 0.08 for formal prompts (95% CIs, $p = 0.002$, paired t-test, $d = 0.59$). Concise prompts yielded the lowest text length variance, mean 8.7 ± 3.4 compared to descriptive prompts at 127.1 ± 19.7 ($p < 0.0001$, $d = 1.44$). Sentiment variance was uniformly low but reached 0.010 ± 0.001 for descriptive prompts, dropping to 0.002 ± 0.0003 for concise prompts ($p = 0.0007$). All patterns remained robust after Holm–Bonferroni correction.

5.5.4 Conclusion. Statistical analysis confirms that tone, style, and linguistic complexity materially impact output reliability. Simpler, informal, and concise prompts consistently yield lower entropy, reduced text length variance, and more stable sentiment. These effects are robust across models and prompt topics. The results support the assertion that prompt construction is an actionable lever for controlling both semantic state uncertainty (ΔS) and output unpredictability (ΔO).

5.6 Experiment 5: Irrelevant Information Challenge

5.6.1 Objective. To evaluate the susceptibility of models to distraction by irrelevant but plausible information within prompts.

5.6.2 Methodology. Pairs of prompts were crafted, each comprising an original prompt and a distractor prompt embedding plausible but extraneous details. For example:

— *Original Prompt*: "Explain the primary causes of the American Civil War."
— *Distractor Prompt*: "Considering the influence of the French Revolution and the invention of the steam engine, explain the primary causes of the American Civil War."

For each pair, we measured accuracy, entropy, and text length variance across $n = 60$ model responses per prompt type.

5.6.3 Results and Discussion. Introducing distractors produced a statistically significant mean accuracy drop of $25.0 \pm 3.7\%$ (95% CI, $t(2) = 9.4$, $p = 0.002$ vs. zero, Hedges' $g = 2.25$), pooled across all topics. Entropy rose by 0.57 ± 0.06 ($t(2) = 14.1$, $p < 0.001$), and text length variance increased by 122 ± 18 ($t(2) = 12.0$, $p = 0.001$). Effects persisted after Holm–Bonferroni correction. Table 3 details per-topic breakdown with CIs.

Qualitative review showed that distractor information was frequently integrated into model explanations, reducing factual accuracy and amplifying response variability. In all cases, the confidence intervals for the mean changes excluded zero.

5.6.4 Conclusion. Statistically, generative models display pronounced vulnerability to irrelevant but plausible context, with consistent and significant declines in accuracy and increases in both entropy and variance. These findings establish the necessity of prompt discipline and validate

Table 4. Cosine Similarity and Accuracy (Mean ± 95% CI, $n = 60$ per Cell)

Model	Prompt Type	Cosine Similarity	Accuracy (%)	Δ Accuracy (%)
GPT-4o	Synonym	0.91 ± 0.03	93 ± 2.1	−3.1 ± 1.3
GPT-4o	Numerical	0.83 ± 0.03	88 ± 2.9	−9.4 ± 2.2
Gemini 1.5 Pro	Synonym	0.87 ± 0.04	89 ± 2.5	−4.0 ± 1.7
Gemini 1.5 Pro	Numerical	0.78 ± 0.03	76 ± 3.5	−12.8 ± 2.8
Claude 3 Sonnet	Synonym	0.89 ± 0.03	91 ± 2.4	−4.1 ± 1.6
Claude 3 Sonnet	Numerical	0.76 ± 0.04	72 ± 3.7	−14.1 ± 3.1

Paired t-tests: all synonym vs. original p = 0.003; numerical vs. original p < 0.001.

prompt minimization as a defensive measure to reduce semantic state uncertainty and boost output fidelity.

5.7 Experiment 6: Statistical Analysis of Semantic Drift

5.7.1 Objective. To quantify the effects of minor semantic changes, such as synonym replacements and numerical adjustments, on model outputs.

5.7.2 Methodology. Prompt pairs were constructed with subtle semantic variations. For example:

— *Original Prompt*: "Describe the impact of deforestation on biodiversity."
— *Modified Prompt (Synonym)*: "Describe the effect of forest clearing on biological diversity."
— *Modified Prompt (Numerical)*: "Describe the impact of reducing forest area by 30% on biodiversity."

Each prompt type was submitted $n = 60$ times per model. Semantic similarity (cosine), accuracy, and tone consistency were assessed between each pair of original and modified responses.

5.7.3 Results and Discussion. Minor semantic modifications produced statistically significant reductions in both cosine similarity and accuracy. Across all models, mean cosine similarity for original vs. synonym prompts was 0.89 ± 0.02 (95% CI), compared to 0.80 ± 0.03 for numerical changes. Paired t-tests indicated that these differences were significant (synonym: $t(5) = 5.12$, p = 0.003, d=2.08; numerical: $t(5) = 7.84$, p < 0.001, d = 3.19). The average drop in similarity from synonym to numerical was −0.09 ± 0.02 (p = 0.002). Accuracy for synonym pairs declined by 3.7 ± 1.6% (mean difference, p = 0.04), while numerical variants induced a larger decrease of 10.8 ± 2.7% (p = 0.003). Confidence intervals for both effects excluded zero. Table 4 summarizes the findings per model and prompt type. Notably, numerical changes led to greater semantic drift than synonym substitution (mean Δ similarity −0.11, 95% CI [−0.15, −0.07], p = 0.001). Qualitative review identified frequent quantitative errors and a tendency to restate but not reason numerically.

5.7.4 Conclusion. Statistical evidence confirms that even minor semantic or quantitative prompt variations can induce measurable, significant changes in output. Both cosine similarity and factual accuracy decline, with the effect magnified for numerical perturbations. These results validate the hypothesis that generative models remain sensitive to subtle semantic drift, reinforcing the need for precise, unambiguous prompt design in any context where output fidelity is critical.

5.8 Overall Discussion

The empirical results establish a statistically robust link between prompt design and generative model behavior. High prompt specificity and controlled complexity reliably decrease output entropy and variance (Pearson $r = -0.71$, p < 0.001; quadratic regression p = 0.02 for complexity term), confirming the AI Uncertainty Principle under multiple operationalizations. Aggregation strategies

such as Sigma-weighted voting yield measurable improvements in accuracy and inter-rater reliability (Hedge's $g > 1.0$, $p < 0.001$). Conversely, prompts with irrelevant information or minor semantic shifts significantly elevate output unpredictability and error rates, as reflected by non-overlapping confidence intervals and large effect sizes in all targeted tests.

The consistency of these patterns across three state-of-the-art models and diverse domains supports both the generalizability and practical utility of the theoretical framework. Statistical corrections for multiple comparisons further reinforce the reliability of these findings. Collectively, the analyses substantiate that deliberate prompt engineering, informed by uncertainty-theoretic insights, is essential for controlling model output stability, reliability, and interpretability in real-world deployments.

6 Discussion

In this section, we discuss the scope and limitations of our Quantum Analogy, and its implications for AI system design.

6.1 Prompt Design and Output Variability

Quantitative analyses confirm that prompt formulation is the most direct and effective lever for modulating generative behavior. Across all models, increasing prompt specificity reduces entropy by 0.65 (95% CI: $[0.42, 0.88]$; $t(35) = 4.32$, $p < 0.001$, Cohen's $d = 1.13$), as reported in Experiment 1. Linear regression further indicates a negative association between specificity and entropy (slope $\beta_1 = -0.62 \pm 0.14$, $R^2 = 0.49$, $p < 0.001$), with each unit increase in specificity compressing the output distribution. The reduction in text-length variance is also substantial, with long prompts yielding $d = 0.88$ lower variance compared to short prompts (ANOVA $F(2, 57) = 14.8$, $p < 0.001$). These empirical effects are robust to model architecture, holding across GPT-4o, Gemini 1.5 Pro, and Claude 3 Sonnet. Practitioners requiring deterministic or highly reliable outputs should therefore prioritize unambiguous entity references, explicit stylistic constraints, and minimization of optional or ambiguous clauses. These design principles are quantitatively justified by the pronounced and statistically significant attenuation of both entropy and variance under well-crafted prompts, as illustrated in Figures 2–3.

6.2 Tradeoffs in Prompt Complexity

Complexity functions as a double-edged instrument: incremental increases up to an intermediate threshold systematically prune the hypothesis space and stabilize output variability (ΔO), but surpassing this threshold reintroduces disorder (Figure 6). Quantitative analysis demonstrates a pronounced U-shaped relationship, with the quadratic model explaining $R^2 = 0.71$ of the variance ($p < 0.0001$) in entropy and length variability. Segmented regression locates the inflection point at complexity level $C^* = 3.2$ (95% CI: [2.8, 3.6]), marking the transition from variance attenuation to resurgence. This breakpoint aligns closely with token windows where self-attention cost grows quadratically, indicating that attention dilution—rather than semantic richness alone—drives the resurgence of output variability. Effective prompt engineering thus requires calibrating descriptive depth to remain within the empirically derived complexity window, not merely accumulating detail. Over-specification can degrade reliability as much as under-specification, especially when model context capacity is exceeded.

6.3 Robustness to Irrelevant Information and Semantic Drift

The Irrelevant Information Challenge demonstrates a persistent vulnerability: models consistently integrate plausible but orthogonal cues, even when such cues contradict domain ground truth. The mean accuracy drop when distractor information is included is $24.6\% \pm 3.2\%$ ($t(11) = 8.1$, $p < 0.001$),

as shown in Table 3, with all confidence intervals excluding zero. Minor lexical shifts—whether synonym substitutions or numeric perturbations—produce additional statistically significant reductions in cosine similarity and factual precision (Table 4). These effects are robust and extend across all model families. The data indicate that internal state uncertainty ΔS is highly sensitive to surface-level prompt variations, supporting the sensitivity parameter P_S in Equation (9). Practitioners seeking to maximize robustness should exclude extraneous context and rigorously fix all critical numerics within prompt templates to suppress avoidable sources of semantic drift and response unreliability.

6.4 Ensemble Strategies for Reliability

Sigma-weighted aggregation consistently tightened confidence intervals around factual items and synthesized divergent stylistic elements in opinion tasks. Because the weighting scheme penalizes high-entropy responses, it implicitly minimizes ΔO without accessing internal activations. The approach is computationally economical compared with full Bayesian model averaging and requires no parameter tuning beyond empirical entropy estimation, making it deployable in latency-sensitive workflows.

6.5 Scope and Limitations of the Quantum Analogy

The AI Uncertainty Principle borrows mathematical form from quantum mechanics yet does not imply that language models instantiate quantum states. Transformers are deterministic dynamical systems augmented by stochastic decoding; the analogy provides a compact descriptive grammar for the interplay between latent-state opacity (ΔS) and output spread (ΔO). No results, derivations, or predictions in this work depend on physical quantum phenomena, operator non-commutativity, or other features unique to quantum systems. Transformers remain deterministic dynamical systems augmented by stochastic decoding; the analogy serves only to provide a descriptive grammar for observed unpredictability, and should not be misconstrued as a literal mapping to quantum theory.

Justification of $\hbar_{AI}$. In Equation (1), we introduced $\hbar_{\text{AI}}$ as a data-driven scaling constant. We estimate it per model by minimizing

$$\hbar^*_{\text{AI}} = \arg\min_{\hbar>0} \sum_{i=1}^{N} \left(\Delta S_i \Delta O_i - \tfrac{\hbar}{2}\right)^2, \tag{12}$$

where i indexes prompt–response pairs. Median $\hbar^*_{\text{AI}}$ values differ by less than 7% across the three systems, implying approximate cross-model consistency while confirming that the constant is not universal in the physical sense.

Boundaries of the Observer Analogy. The "collapse" metaphor assumes that a prompt deterministically selects one trajectory in activation space. Temperature and nucleus sampling introduce additional entropy exogenous to the learned weights, producing a non-zero variability floor even at maximal specificity. Hence, the inverse coupling predicted by Equation (1) approaches—but never reaches—zero, unlike the idealized quantum limit.

Operator Non-Commutativity Caveat. Quantum uncertainty originates from non-commuting observables; transformer weight matrices commute under standard multiplication. Our framework substitutes distributional spread for operator algebra. This substitution explains the smooth, rather than discrete, response of ΔS and ΔO to incremental prompt perturbations.

Empirical Failure Modes. Two regimes violate the analogy: (i) schema-constrained generation, where deterministic decoding drives ΔO to near zero despite opaque internal states, and (ii) retrieval-augmented models, where external documents overwrite latent semantics without an

explicit "measurement" prompt. Both cases highlight that the principle is informative only when generation relies primarily on internal language modeling.

Implications for Theory Building. Researchers should treat $\hbar_{\mathrm{AI}}$ as an empirical convenience, recalibrate it for each model release, and couple the uncertainty formulation with information-theoretic metrics such as mutual information to obtain a substrate-agnostic account of variability.

6.6 Generality across Modalities and Tasks

Although all empirical work herein focuses on text-based LLMs, the uncertainty formalism is *substrate-agnostic*: ΔS is defined as entropy (or dispersion) in the model's latent state, and ΔO as measurable variability in observable outputs, regardless of whether those outputs are tokens, pixels, or program tokens. In image diffusion models, for instance, ΔS can be approximated by the spread of latent noise vectors after a few denoising steps, while ΔO by the variance of Inception features or Fréchet Inception Distance dispersion across repeated *text-to-image* generations; preliminary calculations on Stable Diffusion revealed the same inverse coupling, with $\hbar_{\mathrm{AI}}$ differing only by a constant scaling factor tied to embedding dimensionality.

Our framework also extends to code-generation and speech-synthesis systems: in AlphaCode or CodeLlama, latent program sketches correspond to ΔS, and diversity of compilable outputs or functional behaviors to ΔO; in Tacotron-style TTS, mel-spectrogram variance provides an analogous observable. What changes across modalities is not the tradeoff itself but the operationalization of each term and the noise sources that set the empirical lower bound. Consequently, applying the AI Uncertainty Principle to a new domain requires (i) selecting modality-appropriate entropy and variability metrics, and (ii) re-estimating $\hbar_{\mathrm{AI}}$ under that metric pair—but no alteration to the underlying mathematics. We therefore expect the prompt-design heuristics and ensemble strategies validated on language models to transfer, mutatis mutandis, to multimodal and task-specialized generators, offering a unified handle on reliability across the next wave of generative systems.

7 Conclusion

This study introduces a novel theoretical framework that models the inherent unpredictability of generative AI systems by drawing parallels with quantum mechanics, specifically the uncertainty principle and superposition. By conceptualizing the AI Uncertainty Principle, we provide a deeper understanding of how internal semantic states contribute to output variability in generative models. Our experiments demonstrate that prompt design critically influences AI behavior. Increased specificity and detail in prompts generally lead to more predictable and consistent responses. However, we identified a threshold beyond which additional complexity overwhelms the model, resulting in increased variability and reduced consistency. This finding highlights the necessity of calibrating prompt complexity to align with the model's processing capabilities. We also found that the tone and style of prompts significantly affect output consistency. Informal and concise prompts produce more uniform responses, while formal or complex language introduces variability. This suggests that models are sensitive to linguistic nuances and may prioritize syntactic patterns over semantic content. Furthermore, our investigation reveals that generative models are susceptible to incorporating irrelevant yet plausible information from prompts, leading to decreased accuracy and increased output variability. This underscores the importance of precise and focused prompt design to enhance the reliability of AI-generated content. The application of ensemble techniques, such as Sigma-weighted aggregation across multiple models and prompt variations, proved effective in enhancing output reliability. By mitigating individual model biases and reducing the impact of prompt-induced variability, these methods yield more consistent and robust outputs.

The insights of this study have significant implications for the design and deployment of AI systems. Effective prompt engineering requires a nuanced balance between specificity and complexity, as well as careful consideration of linguistic presentation. Incorporating ensemble methods can further bolster the reliability of AI systems, making them more robust against inherent variability and sensitivities. Future research should explore additional factors influencing output variability, such as models' handling of semantic nuances and their susceptibility to distraction by irrelevant information. Investigating these aspects will deepen our understanding of generative AI behavior and support the development of more accurate and trustworthy AI applications.

References

[1] Ian Arawjo, Chelse Swoopes, Priyan Vaithilingam, Martin Wattenberg, and Elena L. Glassman. 2024. Chainforge: A visual toolkit for prompt engineering and llm hypothesis testing. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 1–18.

[2] Zied Bahroun, Chiraz Anane, Vian Ahmed, and Andrew Zacca. 2023. Transforming education: A comprehensive review of generative artificial intelligence in educational settings through bibliometric and content analysis. *Sustainability* 15, 17 (2023), 12983.

[3] Vadim Borisov, Tobias Leemann, Kathrin Sebler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems* 35, 6 (2024), 7499–7519. DOI: https://doi.org/10.1109/TNNLS.2022.3229161

[4] Mario Brcic and Roman V. Yampolskiy. 2023. Impossibility results in AI: A survey. *ACM Computing Surveys* 56, 1 (2023), 1–24.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems* 33 (2020), 1877–1901.

[6] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology* 15, 3 (2024), 1–45.

[7] Anwoy Chatterjee, Hsvns Kowndinya Renduchintala, Sumit Bhatia, and Tanmoy Chakraborty. 2024. POSIX: A prompt sensitivity index for large language models. In *Findings of the Association for Computational Linguistics (EMNLP '24)*, 14550–14565.

[8] Yogesh K. Dwivedi, Neeraj Pandey, Wendy Currie, and Adrian Micu. 2024. Leveraging ChatGPT and other generative artificial intelligence (AI)-based applications in the hospitality and tourism industry: Practices, challenges and research agenda. *International Journal of Contemporary Hospitality Management* 36, 1 (2024), 1–12.

[9] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. 2023. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review* 56, Suppl 1 (2023), 1513–1589.

[10] DestaHaileselassie Hagos, Rick Battle, and DandaB Rawat. Recent advances in generative AI AND large language models: Current status, challenges, and perspectives. *IEEE Transactions on Artificial Intelligence* 5, 12 (2024), 5873–5893. DOI: https://doi.org/10.1109/TAI.2024.3444742

[11] Werner Heisenberg. 1927. Über den anschaulichen inhalt der quantentheoretischen kinematik und mechanik. *Zeitschrift Fur Physik* 43, 3 (1927), 172–198.

[12] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 1–27.

[13] Kai Hu, Weichen Yu, Yining Li, Tianjun Yao, Xiang Li, Wenhe Liu, Lijun Yu, Zhiqiang Shen, Kai Chen, and Matt Fredrikson. 2024. Efficient llm jailbreak via adaptive dense-to-sparse constrained optimization. *Advances in Neural Information Processing Systems* 37 (2024), 23224–23245.

[14] Ishika Joshi, Simra Shahid, Shreeya Manasvi Venneti, Manushree Vasu, Yantao Zheng, Yunyao Li, Balaji Krishnamurthy, and Gromit Yeuk-Yin Chan. 2025. CoPrompter: User-centric evaluation of LLM instruction alignment for improved prompt engineering. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*, 341–365.

[15] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. arXiv:2207.05221. Retrieved from https://arxiv.org/abs/2207.05221

[16] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2023. Holistic evaluation of language models. *Transactions on Machine Learning Research* 8 (2023), 1–162.

[17] Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3214–3252.

[18] Manikanta Loya, Divya Sinha, and Richard Futrell. 2023. Exploring the sensitivity of LLMs' decision-making capabilities: Insights from prompt variations and hyperparameters. In *Findings of the Association for Computational Linguistics (EMNLP '23)*, 3711–3716.

[19] Timothy R. Mcintosh, Tong Liu, Teo Susnjak, Paul Watters, and Malka N. Halgamuge. 2024. A reasoning and value alignment test to assess advanced GPT reasoning. *ACM Transactions on Interactive Intelligent Systems* 14, 3 (2024), 1–37. DOI: https://doi.org/10.1145/3670691

[20] Timothy R. McIntosh, Tong Liu, Teo Susnjak, Paul Watters, Alex Ng, and Malka N. Halgamuge. 2024. A culturally sensitive test to evaluate nuanced GPT hallucination. *IEEE Transactions on Artificial Intelligence* 5, 6 (2024), 2739–2751. DOI: https://doi.org/10.1109/TAI.2023.3332837

[21] Timothy R. McIntosh, Teo Susnjak, Nalin Arachchilage, Tong Liu, Dan Xu, Paul Watters, and Malka N. Halgamuge. 2025. Inadequacies of large language model benchmarks in the era of generative artificial intelligence. *IEEE Transactions on Artificial Intelligence* (2025), 1–18. DOI: 10.1109/TAI.2025.3569516

[22] Timothy R. McIntosh, Teo Susnjak, Tong Liu, Paul Watters, and Malka N. Halgamuge. 2024. The inadequacy of reinforcement learning from human feedback—radicalizing large language models via semantic vulnerabilities. *IEEE Transactions on Cognitive and Developmental Systems* 16, 4 (2024), 1561–1574. DOI: https://doi.org/10.1109/TCDS.2024.3377445

[23] Timothy R. McIntosh, Teo Susnjak, Tong Liu, Paul Watters, Alex Ng, and Malka N. Halgamuge. 2024. A game-theoretic approach to containing artificial general intelligence: Insights from highly autonomous aggressive malware. *IEEE Transactions on Artificial Intelligence* 5, 12 (2024), 6290–6303. DOI: https://doi.org/10.1109/TAI.2024.3394392

[24] Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. 2024. Tree of attacks: Jailbreaking black-box llms automatically. *Advances in Neural Information Processing Systems* 37 (2024), 61065–61105.

[25] Elias Nehme, Omer Yair, and Tomer Michaeli. 2023. Uncertainty quantification via neural posterior principal components. *Advances in Neural Information Processing Systems* 36 (2023), 37128–37141.

[26] Allen Nie, Yuhui Zhang, Atharva Shailesh Amdekar, Chris Piech, Tatsunori B. Hashimoto, and Tobias Gerstenberg. 2023. MoCa: Measuring human-language model alignment on causal and moral judgment tasks. *Advances in Neural Information Processing Systems* 36 (2023), 78360–78393.

[27] Louis Ohl, Pierre-Alexandre Mattei, Charles Bouveyron, Warith Harchaoui, Mickaël Leclercq, Arnaud Droit, and Frederic Precioso. 2022. Generalised mutual information for discriminative clustering. *Advances in Neural Information Processing Systems* 35 (2022), 3377–3390.

[28] David Oniani, Jordan Hilsman, Yifan Peng, Ronald K. Poropatich, Jeremy C. Pamplin, Gary L. Legault, and Yanshan Wang. 2023. Adopting and expanding ethical principles for generative artificial intelligence from military to healthcare. *NPJ Digital Medicine* 6, 1 (2023), 225.

[29] Keng-Boon Ooi, Garry Wei-Han Tan, Mostafa Al-Emran, Mohammed A. Al-Sharafi, Alexandru Capatina, Amrita Chakraborty, Yogesh K. Dwivedi, Tzu-Ling Huang, Arpan Kumar Kar, Voon-Hsien Lee, et al. 2023. The potential of generative artificial intelligence across disciplines: Perspectives and future directions. *Journal of Computer Information Systems* 65, 1 (2023), 76–107.

[30] Amirhossein Razavi, Mina Soltangheis, Negar Arabzadeh, Sara Salamat, Morteza Zihayat, and Ebrahim Bagheri. 2025. Benchmarking prompt sensitivity in large language models. In *European Conference on Information Retrieval*. Springer, 303–313.

[31] Alessandro Saviolo, Jonathan Frey, Abhishek Rathod, Moritz Diehl, and Giuseppe Loianno. 2024. Active learning of discrete-time dynamics for uncertainty-aware model predictive control. *IEEE Transactions on Robotics* 40 (2024), 1273–1291. DOI: https://doi.org/10.1109/TRO.2023.3339543

[32] Erwin Schrodinger. 1935. Die gegenwartige situation in der quantenmechanik. *Naturwissenschaften* 23, 50 (1935), 844–849.

[33] Naichen Shi, Fan Lai, Raed Al Kontar, and Mosharaf Chowdhury. 2024. Fed-ensemble: Ensemble models in federated learning for improved generalization and uncertainty quantification. *IEEE Transactions on Automation Science and Engineering* 21, 3 (2024), 2792–2803.

[34] Bohan Tang, Yiqi Zhong, Chenxin Xu, Wei-Tao Wu, Ulrich Neumann, Ya Zhang, Siheng Chen, and Yanfeng Wang. 2023. Collaborative uncertainty benefits multi-agent multi-modal trajectory forecasting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 11 (2023), 13297–13313. DOI: https://doi.org/10.1109/TPAMI.2023.3290823

[35] Daochen Wang, Aarthi Sundaram, Robin Kothari, Ashish Kapoor, and Martin Roetteler. 2021. Quantum algorithms for reinforcement learning with a generative model. In *International Conference on Machine Learning*. PMLR, 10916–10926.

[36] Huai-Yu Wang. 2022. Exploring the implications of the uncertainty relationships in quantum mechanics. *Frontiers in Physics* 10 (2022), 1059968.

[37] Xiaofei Wang, Yiwen Han, Victor C. M. Leung, Dusit Niyato, Xueqiang Yan, and Xu Chen. 2020. Convergence of edge computing and deep learning: A comprehensive survey. *IEEE Communications Surveys and Tutorials* 22, 2 (2020), 869–904.

[38] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed. H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The 11th International Conference on Learning Representations*, 1–24.

[39] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems* 36 (2023), 80079–80110.

[40] Yunlong Xiao, Kuntal Sengupta, Siren Yang, and Gilad Gour. 2021. Uncertainty principle of quantum processes. *Physical Review Research* 3, 2 (Apr. 2021), 023077.

[41] Roman Yampolskiy. 2020. On controllability of artificial intelligence. In *IJCAI-21 Workshop on Artificial Intelligence Safety (AISafety '21)*, 1–59.

[42] Jun Yan, Vikas Yadav, Shiyang Li, Lichang Chen, Zheng Tang, Hai Wang, Vijay Srinivasan, Xiang Ren, and Hongxia Jin. 2024. Backdooring instruction-tuned large language models with virtual prompt injection. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 6065–6086.

[43] Chuanpeng Yang, Yao Zhu, Wang Lu, Yidong Wang, Qian Chen, Chenlong Gao, Bingjie Yan, and Yiqiang Chen. 2024. Survey on knowledge distillation for large language models: Methods, evaluation, and application. *ACM Transactions on Intelligent Systems and Technology* (2024), 1–27.

[44] Mozhi Zhang, Mianqiu Huang, Rundong Shi, Linsen Guo, Chong Peng, Peng Yan, Yaqian Zhou, and Xipeng Qiu. 2024. Calibrating the confidence of large language models by eliciting fidelity. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2959–2979.

[45] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*. PMLR, 12697–12706.

[46] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. In *The 11th International Conference on Learning Representations*, 1–43.

[47] Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Yue Zhang, Neil Gong, et al. 2023. Promptrobust: Towards evaluating the robustness of large language models on adversarial prompts. In *Proceedings of the 1st ACM Workshop on Large AI Systems and Models with Privacy and Safety Analysis*, 57–68.

[48] Jingming Zhuo, Songyang Zhang, Xinyu Fang, Haodong Duan, Dahua Lin, and Kai Chen. 2024. ProSA: Assessing and understanding the prompt sensitivity of LLMs. In *Findings of the Association for Computational Linguistics (EMNLP '24)*, 1950–1976.

Appendices

A Table of Mathematical Symbols and Additional Theoretical Notes

A.1 Mathematical Symbols and Parameter Definitions

Table A1. Table of Mathematical Symbols and Parameter Definitions

Symbol	Formal Definition	Plain-language Explanation	Units/Typical Range	Where Used
ΔS	Uncertainty in internal semantic state	Unpredictability of model's latent knowledge/configuration before a prompt	Entropy (nats/bits), $[0, H_{\max}]$	Equations (1), (10), all sections
ΔO	Output variability	How much model outputs differ given same/varied prompt	Entropy, variance, semantic distance	Equations (1), (10), all sections
$\hbar_{\text{AI}}$	Empirical uncertainty constant	Model-specific lower bound on product $\Delta S \cdot \Delta O$	Fitted constant, > 0	Equation (1), Discussion
$\mathbf{W}$	Weight matrix	Maps activations to concept space/output	Real matrix, dimension $d_c \times d_x$	Equations (2) and (6)
$\mathbf{c}$	Concept vector	Degree to which each concept is present in the state	Real vector, d_c	Equation (2)
$\mathbf{x}$	Activation vector	Activations of all neurons for a given state/input	Real vector, d_x	Equations (2), (3), and (5)
$\mathbf{D}$	Dictionary matrix	Basis of prototype neural activation patterns	Real matrix, $d_x \times d_b$	Equations (3), (5), Appendix B
$\mathbf{A}$	Sparse code matrix	Codes that linearly combine dictionary elements	Sparse matrix, $d_b \times m$	Equations (3), (4), Appendix B
A, B	Uncertainty parameters	Model-specific upper/lower uncertainty bounds	Fitted constants	Equation (9), Sec. B
λ_i	Prompt factor weights	Sensitivity of uncertainty to prompt feature i	Fitted, $\lambda_i > 0$	Equation (9), Table, Sec. B
L_P	Prompt length	Number of tokens/clauses in prompt	Integer, $L_P \geq 1$	Equation (9)
I_C	Informational content	Factual or semantic richness of prompt	Non-negative real	Equation (9)
D_F	Domain familiarity	Match between prompt topic and training data	Normalized $[0, 1]$	Equation (9)
U_C	User clarity	Explicitness/clarity of instructions	Normalized $[0, 1]$	Equation (9)
P_S	Prompt structure	Regularity, formal structure, or syntactic precision	Normalized $[0, 1]$	Equation (9)
α	Feature amplification factor	Scalar controlling strength of feature modification	Real, typically $\lvert\alpha\rvert < 5$	Equation (4), Appendix B
κ	Scaling constant for lower bound	Governs minimal achievable semantic state uncertainty	Fitted/empirical, > 0	Equation (10), Appendix B
ϵ	Stability constant	Prevents denominator vanishing in lower bound	Small positive (10^{-6} to 10^{-3})	Equation (10)
Q, K, V	Query, key, value matrices (transformer)	Transform input to attend over context	Real matrices, model-dependent	Equation (7)
d_k	Key dimension (transformer)	Scaling term for attention normalization	Integer, $d_k > 0$	Equation (7)
z_i	Logit/input to softmax	Activation for output class i	Real number	Equation (8)

For further empirical and computational details of parameter estimation, see Section 5, and Appendix Section B.

A.2 Intuitive Rationales and Mathematical Deductions

A.2.1 Why Exponential Decay for Uncertainty? The refined uncertainty relationship in Equation (9)

$$\Delta S \cdot \Delta O = A\, e^{-(\lambda_1 L_P + \lambda_2 I_C + \lambda_3 D_F + \lambda_4 U_C + \lambda_5 P_S)} + B$$

arises naturally from the observed multiplicative, not additive, reduction in uncertainty as independent prompt features are improved. Suppose each feature f_i independently reduces uncertainty by a factor $e^{-\lambda_i f_i}$, the combined effect is a product of exponentials:

$$\Delta U = \prod_{i=1}^{n} e^{-\lambda_i f_i} = e^{-\sum_{i=1}^{n} \lambda_i f_i}.$$

This derivation holds under the statistical independence assumption that prompt features affect uncertainty orthogonally. Such log-linear behavior is consistently observed in empirical data and matches practical prompt engineering scenarios. The exponential form captures sharply diminishing returns: as prompts become longer, clearer, or more structured, the marginal reduction in uncertainty becomes smaller. Empirically, alternative (e.g., linear or polynomial) forms systematically underperform in fitting observed model behavior, especially for large LLMs and real-world prompts.

A.2.2 Fitting and Estimating $\hbar_{AI}$, λ_i, A, B, κ, and ϵ. The constant $\hbar_{\text{AI}}$ is estimated by measuring the empirical lower bound of $\Delta S \cdot \Delta O$ across a wide set of controlled prompt perturbations, using least-squares or maximum-likelihood fitting to identify the minimum observed value. Parameters A, B, λ_i, κ, and ϵ are determined by nonlinear regression on experimental data (see Section 6), using regularization and cross-validation to ensure robust fit and avoid overfitting.

For practical estimation:

- Begin with baseline prompts and incrementally add length, structure, or clarity, measuring resulting entropy (for ΔO) and latent dispersion (for ΔS).
- Fit λ_i by regressing $\log(\Delta S \cdot \Delta O - B)$ against each prompt feature, holding others fixed when feasible.
- Select ϵ to be as small as possible while ensuring numerical stability under empirical noise, to avoid division by zero in Equation (10).
- Fit κ so that the observed lower bound on ΔS matches empirical minima in highly constrained output experiments.

Recommended ranges. For most LLMs, $\lambda_i \in [0.1, 2]$, with A and B tuned to observed extremes of uncertainty, and ϵ at or just above the noise floor.

A.2.3 Explicit Proof: Exponential Form Derivation. If each prompt factor f_i independently reduces total uncertainty by $e^{-\lambda_i f_i}$, the overall effect is

$$\Delta U = U_0 \prod_{i=1}^{n} e^{-\lambda_i f_i} = U_0 e^{-\sum_i \lambda_i f_i},$$

where U_0 is the maximal uncertainty with all features at baseline. This formalizes the multiplicative independence assumption. The exponential form can also be justified via information theory: if each prompt dimension increases information by a fixed rate, entropy decays exponentially with the sum of those features.

A.2.4 Convexity and Local Minima in Alternating Minimization. The dictionary learning objective $f(\mathbf{D}, \mathbf{A}) = \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2$ is not jointly convex in $(\mathbf{D}, \mathbf{A})$, but is convex in either $\mathbf{D}$ or $\mathbf{A}$ when the other is fixed. Alternating minimization (block coordinate descent) guarantees that the objective decreases monotonically and converges to a stationary point. While global optimality is not assured due to non-convexity, local minima are typically sufficient in practice for stable and interpretable decompositions, as supported by empirical convergence in real-world neural data and the sparsity constraint.

A.2.5 Well Posedness of the Uncertainty Lower Bound. Lemma. If both ΔS and ΔO are empirically measured, non-negative, and continuous, the fitted $\hbar_{\mathrm{AI}}$ is the infimum of $\Delta S \cdot \Delta O$ over all tested prompt configurations. In practice, measurement noise and finite sampling mean the bound may be approached but not attained; degenerate or adversarial prompts may yield values arbitrarily close to the bound, but not below it. Thus, the lower bound is well posed as a data-driven empirical minimum, with rare violations attributable to noise or nonstationarity.

A.2.6 Relationship between Entropy and Euclidean Norm. Entropy and ℓ_2 norm (Euclidean norm) are used as metrics of output variability, but serve distinct roles. For Gaussian-distributed outputs, variance (as captured by ℓ_2 norm) is proportional to entropy. For categorical or softmaxed output distributions (typical in LLMs), higher ℓ_2 of logits often correlates with greater entropy in the predicted distribution, but the relationship is not strictly monotonic or linear due to the softmax nonlinearity. When measuring output variability, select the metric aligned with your experimental objective: entropy for unpredictability, norm for magnitude of deviation.

A.2.7 Scaling Arguments for the ϵ Regularizer. The constant ϵ in Equation (10) is required for numerical stability. If ϵ is too small, minor fluctuations or noise in ΔO may cause artificially inflated estimates of ΔS, leading to instability or overfitting. If too large, ϵ will mask genuine lower bounds and artificially inflate the uncertainty floor. In practice, ϵ should be set just above the level of measurement noise, using validation data to check for artifactual effects.

A.2.8 Parameter Identifiability and Uniqueness. Due to the exponential scaling, only relative values of λ_i (prompt factor sensitivities) are interpretable; absolute scale can be absorbed into A. Parameters A and B are identifiable up to a baseline shift set by the maximum and minimum observed uncertainty. In empirical fitting, regularization or parameter constraints can further improve identifiability, but functional equivalence under affine reparameterizations remains.

A.2.9 Sensitivity and Robustness Analysis. The local sensitivity of the uncertainty product to each prompt feature f_i is obtained by differentiation:

$$\frac{\partial}{\partial f_i}(\Delta S \cdot \Delta O) = -\lambda_i \left[\Delta S \cdot \Delta O - B\right].$$

This shows that each λ_i modulates the rate at which improvements in feature f_i reduce uncertainty, with diminishing effect as $\Delta S \cdot \Delta O$ approaches B. For practitioners, this analysis enables targeted prompt engineering: focus efforts on the feature(s) with the largest λ_i to achieve maximal reduction in uncertainty.

A.2.10 Practitioner's Guide: Estimating and Using Uncertainty Bounds. Estimating parameters in practice:

— Collect repeated outputs from the model for controlled prompt variations.
— Compute empirical entropy or semantic variance of outputs for ΔO; estimate ΔS via clustering or latent dispersion measures.

—Fit the exponential form to these observations, using least-squares or maximum likelihood methods.
—Validate model fit on held-out prompt types or domains.

Recommended Ranges. For most LLMs, $\lambda_i \in [0.1, 2]$, A and B tuned to observed extremes of uncertainty, and ϵ as small as possible without instability.

Interpretation. If, after prompt engineering, further changes to L_P, I_C, D_F, U_C, or P_S yield negligible reduction in $\Delta S \cdot \Delta O$, this signals that irreducible uncertainty is reached. At this point, further tuning cannot produce more reliable outputs, and system design should account for this theoretical limit.

B Proof of Convergence

To establish the reliability and predictability of our theoretical models, we provide proofs of convergence for key components, ensuring that they asymptotically approach stable solutions. These proofs build upon the foundations established in Section 3 and validate the robustness of our proposed formulations.

B.1 Dictionary Learning

Intuitive Summary

The alternating-minimization routine alternately refines the sparse codes and the dictionary matrix; because each step can only decrease (or leave unchanged) the total reconstruction error—which is bounded below by zero—the error sequence is monotone non-increasing and therefore convergent, guaranteeing that the learned basis settles to a stable representation.

We provide a convergence proof for the alternating minimization process used to solve the dictionary learning problem introduced in Section 3.3 (see Appendix A for notation). The aim is to ensure that, through repeated updates, both the dictionary matrix $\mathbf{D}$ and the sparse code matrix $\mathbf{A}$ stabilize to a solution that reliably represents the internal state structure. The alternating minimization procedure proceeds as follows:

(1) *Sparse Code Update*: For fixed $\mathbf{D}^{(k)}$, update $\mathbf{A}$ to minimize the reconstruction error under the sparsity constraint.
(2) *Dictionary Update*: For fixed $\mathbf{A}^{(k+1)}$, update $\mathbf{D}$ to further minimize the reconstruction error, subject to normalization constraints.

At each step, the objective function—the total squared error between data and its reconstruction—never increases. Since this error is always non-negative, the sequence of errors generated by the algorithm forms a non-increasing, bounded-below sequence.

In plain terms, this ensures that, with each iteration, the solution improves or remains unchanged and can never diverge or cycle indefinitely. Over time, the process converges to a stable configuration—a stationary point—where further updates do not change the learned dictionary or code matrices. This guarantees that the extracted building blocks (dictionary elements) become consistent and that the internal representation settles, providing a robust basis for further analysis and manipulation.

B.2 Feature Manipulation

Intuitive Summary

Because the dictionary $\mathbf{D}$ and read-out matrix $\mathbf{W}$ are fixed and bounded, any finite change applied to a sparse feature vector produces a proportionally bounded change in the activation vector and thus in the output, ensuring that small or moderate feature tweaks cannot trigger runaway behavior.

We analyze the stability of the system under targeted perturbations to feature activations. When a particular feature is adjusted—for example, by amplifying a sentiment or topic-related component in the sparse code matrix—the resulting changes in the activation vector and thus the output are determined by the properties of the dictionary and weight matrices. Provided these matrices remain bounded and the perturbation magnitude does not push activations beyond the normal operating regime, the resulting output variability will also be bounded and will converge to a stable value.

In practical terms, the boundedness of the matrices $\mathbf{D}$ and $\mathbf{W}$ ensures that even after manipulating a feature, the system cannot diverge or yield uncontrolled outputs. Only if activations are pushed well outside their typical range (e.g., by selecting an unreasonably large amplification factor) would the linear approximation and resulting predictability break down. This analysis justifies that, for feature modifications within empirically validated ranges, output variability remains both controllable and predictable.

B.3 Predicting Output Variability

Intuitive Summary

Once the internal activation vector stabilizes, multiplying it by a fixed read-out matrix and taking its Euclidean norm yields a quantity that also stabilizes; hence, our empirical variability metrics inherit convergence directly from the earlier steps.

We aim to ensure that the measures of output variability, as defined in Equation (6), are stable under controlled manipulations of internal features.

Convergence Proof.

Given that the activation vector $\mathbf{x}'$ converges—a consequence of the boundedness of the dictionary matrix $\mathbf{D}$ and the controlled modification of the sparse codes—and that $\mathbf{W}$ is fixed and bounded, it follows directly that:

—The product $\mathbf{W}\mathbf{x}'$ converges.
—The Euclidean norm $\|\mathbf{W}\mathbf{x}'\|_2$ also converges to a stable value.

In practical terms, this means that small, controlled adjustments to a model's internal parameters or feature activations will not cause the model's output to become erratic or unbounded. Instead, the variability in outputs, as measured empirically through entropy, variance in text length, or semantic metrics, will remain predictable and within a finite, interpretable range. This property is essential for any application requiring dependable and reproducible model behavior under prompt or feature perturbation.

B.4 Enhanced Uncertainty Function

Intuitive Summary

Because the exponential term shrinks as any prompt-quality factor grows, the product $\Delta S \cdot \Delta O$ monotonically approaches the irreducible baseline B, confirming that uncertainty can be reduced—but never eliminated—through better prompts.

We demonstrate that the refined uncertainty relationship in Equation (9) converges as the input parameters increase. Definitions and empirical examples of all parameters are provided in Table A1 for clarity.

Convergence Proof. The enhanced uncertainty function is

$$\Delta S \cdot \Delta O = A\, e^{-(\lambda_1 L_P + \lambda_2 I_C + \lambda_3 D_F + \lambda_4 U_C + \lambda_5 P_S)} + B. \tag{B1}$$

As any of the input factors (L_P, I_C, D_F, U_C, P_S; see Appendix) increase, the exponent becomes more negative, driving the exponential term toward zero. Thus, the product $\Delta S \cdot \Delta O$ approaches the lower bound B.

Intuitively, this result confirms that as prompts become longer, more informative, or more precisely structured, the overall uncertainty in the system cannot shrink beyond a baseline determined by the model's inherent unpredictability. This convergence is fundamental to the theory's practical value: it ensures that uncertainty reduction through prompt engineering exhibits diminishing returns and stabilizes, rather than vanishing entirely.

B.5 Semantic State and Output Uncertainty Relationship

Intuitive Summary

Even when output variability is driven arbitrarily low, a non-zero lower bound on internal-state uncertainty persists, reflecting the intrinsic opacity of high-dimensional generative models.

We confirm the inverse relationship between semantic state uncertainty ΔS and output uncertainty ΔO, as formally captured in Equation (10):

$$\Delta S \geq \kappa \left(\frac{1}{\Delta O + \epsilon} \right) \tag{B2}$$

Convergence Proof. Because ΔO converges to a finite value and $\epsilon > 0$ by construction, the term $\frac{1}{\Delta O + \epsilon}$ also converges to a positive finite bound. Therefore, ΔS remains bounded from below as output variability shrinks.

Empirically, this establishes a fundamental lower bound: no matter how precisely one constrains or optimizes the model's output through prompt design or aggregation, some irreducible ambiguity about the internal semantic state persists. This residual uncertainty reflects the intrinsic opacity of high-dimensional neural activations in generative systems and cannot be eliminated by external control alone. Recognizing this lower bound is essential for applications demanding reliability, as it places a theoretical limit on the predictability and interpretability achievable through prompt engineering or output filtering.

C Representative Model Outputs

This appendix provides representative, anonymized model outputs for each experimental condition. For brevity, responses are excerpted and identifiers are omitted. The commentary highlights

characteristic behaviors and failure modes, substantiating the quantitative analyses in the main text. For additional outputs or scripts, please contact the corresponding author after manuscript acceptance.

C.1 Experiment 1: Prompt Specificity and Output Entropy

Prompt. Short and Vague — "Describe a place."

GPT-4o. "It is a bustling city filled with people, tall buildings, and the sound of traffic echoing through the streets. Vendors line the sidewalks, and lights illuminate the night."
Gemini 1.5 Pro. "A peaceful garden with flowers blooming, birds singing, and sunlight streaming through the trees. The air is fresh, and a gentle breeze moves the leaves."
Claude 3 Sonnet. "An isolated mountain village surrounded by mist, with wooden cottages and narrow stone paths weaving between them."

Commentary. Vague prompts yield highly divergent responses, differing in both domain (urban, rural, natural) and detail. Entropy is high, confirming statistical findings.

Prompt. Short and Specific—"Describe the Eiffel Tower."

GPT-4o. "The Eiffel Tower is a wrought-iron lattice tower in Paris, standing 324 meters tall. Its four pillars curve inward to meet at the top, and it glows at night with thousands of lights."
Gemini 1.5 Pro. "Located in Paris, the Eiffel Tower is an iconic landmark made of iron, designed by Gustave Eiffel. It attracts millions of visitors each year."
Claude 3 Sonnet. "Rising above the Parisian skyline, the Eiffel Tower is a symbol of France, constructed from iron beams and visible from much of the city."

Commentary. Specific prompts elicit closely aligned outputs, reducing lexical diversity and entropy. All models converge on core facts, confirming the effect of specificity.

C.2 Experiment 2: Sigma-Weighted Aggregation

Prompt. "What is the largest city in Switzerland?"

GPT-4o. "Zurich is the largest city in Switzerland."
Gemini 1.5 Pro. "Zurich is the biggest city in Switzerland by population."
Claude 3 Sonnet. "Zurich is Switzerland's largest city."
Sigma-Weighted Aggregation. "Zurich."

Commentary. All models agree on the correct answer, and aggregation yields a robust output. In instances where models diverged (e.g., rare mention of Geneva), aggregation eliminated the minority response, as reflected in improved consistency statistics.

Prompt. Opinion-based—"Is remote work better than office work?"

GPT-4o. "Remote work offers flexibility and autonomy, but can lead to isolation. Office work fosters collaboration but may reduce work–life balance."
Gemini 1.5 Pro. "Remote work increases flexibility but can make communication harder. Office environments help teams build relationships."
Claude 3 Sonnet. "Remote work allows for independence, but some people prefer the structure and social interaction of office settings."
Sigma-Weighted Aggregation. "Remote work provides flexibility and autonomy, but may reduce collaboration and increase isolation; office work supports social interaction."

Commentary. Aggregation synthesizes key themes and reduces stylistic idiosyncrasy, aligning with measured increases in inter-rater reliability (κ).

C.3 Experiment 3: Incremental Prompt Complexity

Prompt. "Define gravity."

Claude 3 Sonnet. "Gravity is a fundamental force that attracts two bodies with mass toward each other."

Prompt. "Define gravity in the context of Newtonian physics, discuss its impact on planetary motion, compare it with general relativity, and explain its role in modern astrophysics."

Claude 3 Sonnet. "In Newtonian physics, gravity is the force attracting masses, described by the inverse-square law. It explains planetary orbits as ellipses. General relativity reinterprets gravity as spacetime curvature, predicting phenomena like gravitational lensing. In modern astrophysics, gravity governs galaxy formation and cosmic structure evolution."

Commentary. As prompt complexity increases, outputs become initially more detailed and consistent; past the breakpoint, some responses omit or conflate required elements, increasing entropy and variance, consistent with the U-shaped curve.

C.4 Experiment 4: Tone, Style, and Output Predictability

Prompt. Formal Tone, Simple Vocabulary—"Please provide a detailed explanation of climate change, including its causes and effects, in no more than five sentences."

GPT-4o. "Climate change refers to long-term shifts in weather patterns and temperatures. Its main cause is the increase of greenhouse gases due to human activities. Effects include rising sea levels and more frequent extreme weather. These changes threaten ecosystems and human societies. International cooperation is required for mitigation."

Prompt. Informal Tone—"Hey, can you explain what climate change is and what causes it in up to five sentences?"

Claude 3 Sonnet. "Climate change means the planet's getting warmer over time. It's mainly caused by burning stuff like coal and oil, which puts gases into the air. These gases trap heat, kind of like a blanket. Because of this, we see things like more storms and melting ice. People are working on ways to fix it."

Commentary. Informal prompts result in more conversational, less variable outputs, matching lower entropy in quantitative analysis.

C.5 Experiment 5: Irrelevant Information Challenge

Prompt. Original—"Explain the primary causes of the American Civil War."

Gemini 1.5 Pro. "The main causes were sectional differences over slavery, economic factors, and states' rights. Tensions escalated after the election of Abraham Lincoln."

Prompt. Distractor—"Considering the influence of the French Revolution and the invention of the steam engine, explain the primary causes of the American Civil War."

Gemini 1.5 Pro. "The American Civil War's primary causes were slavery and states' rights, but international events like the French Revolution may have influenced political thought, and the invention of the steam engine shaped economic growth."

Commentary. Distractor prompts lead to integration of irrelevant details, increasing factual error and response variability, in line with the measured accuracy drop and variance increase.

C.6 Experiment 6: Semantic Drift

Prompt. Original. "Describe the impact of deforestation on biodiversity."

GPT-4o. "Deforestation reduces habitat availability, causing species loss and decreasing ecosystem diversity. It also disrupts ecological networks and can threaten endangered species."

Prompt. Modified (Synonym). "Describe the effect of forest clearing on biological diversity."

GPT-4o. "Forest clearing leads to loss of habitats, endangering many forms of biological diversity. This can cause extinction and reduce ecosystem resilience."

Prompt. Modified (Numerical). "Describe the impact of reducing forest area by 30% on biodiversity."

GPT-4o. "A 30% reduction in forest area would dramatically reduce available habitats, likely causing significant drops in biodiversity and possible extinction of species unable to adapt or relocate."

Commentary. While synonym substitutions produce small shifts in phrasing, numerical modifications drive larger differences in content and risk computational or factual errors, reflected in greater declines in cosine similarity and accuracy.

D Open Source Model Evaluation and Analysis of Model-Specific Factors

This appendix addresses two critical reviewer concerns: (1) the need for transparent evaluation using open source LLMs to complement results from proprietary systems, and (2) the necessity of analyzing model-specific factors—such as parameter count, training data, and fine-tuning regimes—that influence output variability independently of prompt engineering. During the minor revision phase and upon reviewer request, we conducted a full-scale replication of all main text experiments on the open source *DeepSeek LLM 7B Base* and *DeepSeek LLM 7B Chat* models. We systematically analyze model-specific influences on uncertainty in light of both empirical findings and architectural differences. Given the rapid evolution and proliferation of open LLM variants, we have focused on DeepSeek LLM 7B due to its representative architecture, public weights, and strong performance on established benchmarks. All methods and evaluation protocols exactly matched those for the proprietary models.

D.1 Comprehensive Evaluation of Open Source LLMs

In this revision, we evaluated *DeepSeek LLM 7B Base* and *DeepSeek LLM 7B Chat*—recent, openly licensed transformer models—using the complete set of 104 experimental prompts and the same response replication and sampling strategy as for GPT-4o, Gemini 1.5 Pro, and Claude 3 Sonnet. Both models were accessed through HuggingFace under default configuration on consumer-grade GPUs. Output entropy, length variance, and all statistical analyses were calculated identically for all five models.

Key Findings. The results, summarized in Table D1, reveal three salient patterns:

(1) *Consistent Uncertainty Trends across Architectures.* Both DeepSeek models display the same qualitative and quantitative relationship between prompt specificity and output entropy as proprietary systems: increased prompt precision consistently lowers entropy and variance. The exponential decay of the uncertainty product, as described in Equation (9), fits the empirical data across all five models.

(2) *Influence of Model Alignment and Fine-Tuning.* DeepSeek LLM 7B Chat, which is instruction-tuned for conversational use, consistently yields lower output variability and entropy than its base (pretrained-only) counterpart, closely aligning with the reduction seen in more heavily aligned proprietary models. The difference in mean entropy between DeepSeek Base and Chat models (0.24, $p < 0.01$, paired t-test, full test set) confirms that RLHF and instruction fine-tuning systematically dampen stochasticity and reinforce output regularity in open source models as well.

(3) *Effect of Model Scale.* All models, regardless of scale or licensing, exhibit the same core empirical relationship between prompt design and uncertainty reduction. However, the absolute baseline of entropy and variance remains higher for the smaller (7B) DeepSeek

Table D1. Comparison of Model Properties and Output Variability (Mean ± SD, Full Test Set)

Model	Type	Parameters	Training Data	Fine-Tuning	Mean Entropy	Length Variance
GPT-4o	Proprietary	Estimated 1.8T+	Mixed web, code, proprietary	RLHF, SFT	7.88 ± 0.23	156 ± 29
Gemini 1.5 Pro	Proprietary	Estimated 1.5T+	Mixed web, code, proprietary	RLHF, SFT	8.12 ± 0.27	181 ± 33
Claude 3 Sonnet	Proprietary	Estimated 0.5T–1T	Mixed, heavily filtered web	RLHF, SFT	8.05 ± 0.31	143 ± 22
DeepSeek LLM 7B Base	Open Source	7B	2T tokens, English/Chinese	None (Base)	8.22 ± 0.26	219 ± 38
DeepSeek LLM 7B Chat	Open Source	7B	2T tokens, English/Chinese	SFT, RLHF (Chat)	7.98 ± 0.21	171 ± 32

models, compared to trillion-parameter proprietary LLMs. These differences are quantitatively moderate and do not reverse any of the major effects of prompt control, highlighting structural parallels between open and closed models.

D.2 Critical Insights: Model-Specific Factors Shaping Output Variability

Prompt formulation remains the most immediate and actionable lever for controlling LLM output, but model-specific architectural and training factors fundamentally constrain the attainable bounds of entropy, semantic drift, and error rate. We highlight the following influences, supported by both open source and proprietary results:

— *Parameter Count and Scale.* Larger models (e.g., GPT-4o at over 1 trillion parameters) demonstrate greater context retention, smoother prompt sensitivity curves, and consistently lower entropy floors. Small and midsize open source models, while highly capable, are more susceptible to prompt perturbations and less robust to adversarial or ambiguous inputs.
— *Pretraining Data Scope and Quality.* Models pretrained on more diverse and extensive datasets (e.g., 2T tokens, multilingual coverage in DeepSeek; proprietary multi-terabyte corpora for GPT-4o) exhibit better generalization and more uniform output distributions under prompt variation. Contamination, recency, and data curation directly affect factual consistency and susceptibility to irrelevant context.
— *Supervised Fine-Tuning and RLHF.* Instruction-tuned models (e.g., DeepSeek Chat, GPT-4o, Gemini, Claude) show measurably lower entropy and semantic variance than their base/pretrained-only counterparts (e.g., DeepSeek Base). RLHF systematically sharpens model responses, suppresses hallucinations, and narrows the range of plausible completions. This is evident in both proprietary and open source settings.
— *Decoding Strategy.* All tested models—open and closed—exhibit increased entropy under higher sampling temperature or nucleus sampling parameters. Greedy or beam search decoding reduces variability at the expense of diversity, but does not fully eliminate stochastic effects in large-scale models.
— *Tokenization and Vocabulary Coverage.* Differences in tokenizer design (e.g., byte pair encoding vs. unigram models) can marginally affect length variance and prompt sensitivity, especially in multilingual or code-rich tasks, but these effects are secondary to parameter scale and alignment regime.

D.3 Interpretation, Reproducibility, and Generalizability

Our results demonstrate that the refined AI Uncertainty Principle applies robustly across both proprietary and open source LLMs, regardless of scale or licensing. All transformer-based models—whether billion-parameter open weights or trillion-parameter closed weights—display the same exponential relationship between prompt features and uncertainty bounds. Instruction-tuned and RLHF-aligned models reliably yield more stable outputs, but do not fundamentally alter the underlying trajectory. Architectural transparency in open source models (e.g., DeepSeek, Llama-2,

Mistral) facilitates reproducible analysis of failure cases and prompt-pathologies that may be inaccessible in closed systems. While the magnitude of uncertainty reduction achievable through prompt engineering is bounded by each model's scale, alignment, and data coverage, the direction and structure of these effects are model-agnostic and can be expected to generalize to new architectures. Given the diversity and pace of open source LLM development, we have prioritized a representative and widely adopted architecture in this revision; however, the uncertainty phenomena we report are structurally inherent to the generative modeling paradigm and not a quirk of DeepSeek. We strongly encourage interested readers to replicate our protocol on other open source models, such as Llama, Mistral, or Qwen. We expect that the central findings on prompt sensitivity and entropy bounds will be robust to the choice of architecture, training data, or fine-tuning regime, provided the models employ autoregressive or transformer-based generation.

Limitations and Recommendations. This open source evaluation was completed during the minor revision phase in direct response to reviewer requests. Although there are now at least tens of thousands of high-performing open LLMs and derivative models, the empirical consistency we observe between DeepSeek and proprietary LLMs supports our claim of generalizability. As model capabilities and benchmarks evolve, future studies should incorporate an even broader suite of open models and standardized prompt sets. To support full transparency and replicability, we recommend open release of all prompt templates, evaluation scripts, and system configurations.

Conclusion. Prompt design remains a dominant factor shaping output variability, but architectural scale, data curation, and fine-tuning method are equally critical and quantifiable influences. Open source models exhibit the same core uncertainty dynamics as their proprietary counterparts, but offer additional transparency for interpretability and forensic analysis. We encourage the community to treat prompt, architecture, and data as co-equal axes of control in any rigorous LLM evaluation or system deployment.

E Guidance for Real-World Deployment

The theoretical framework and empirical findings outlined in this article have direct implications for the design, deployment, and operationalization of generative AI systems in production environments. This section provides concrete, actionable recommendations and system-level workflows, informed by experimental evidence and current best practice, to enable practitioners and architects to translate theory into robust, cost-aware, and reliable AI deployments.

E.1 Stepwise Deployment Recommendations

(1) Formalize Prompt Policies at the System Level.
   — *Define Static Input Constraints*: Explicitly specify required keywords, phrase structures, and acceptable entities for each use case. Use regular expressions or schema validation to enforce well-formed inputs, reducing ambiguity in both user-driven and programmatic prompting.
   — *Ban Distractors*: Maintain a repository of banned patterns (e.g., irrelevant clauses, ambiguous numerics, distractor topics) informed by empirical error analysis. Integrate pre-deployment prompt linting to detect and strip extraneous or misleading information prior to submission to the generative model.
   — *Establish Versioned Prompt Templates*: Track revisions to prompt formulations and conduct A/B testing at deployment scale, retaining empirical metrics for entropy and output variance associated with each template.
(2) Implement Prompt Engineering as a Continuous Process.

— *Monitor Live Prompt–Response Pairs*: Log and periodically review both input prompts and generated outputs. Track key metrics (output entropy, variance, error rate, time-to-completion) and flag outliers for manual or automated prompt refinement.
— *Introduce Feedback Loops*: Enable real-time feedback from human users or automated validators to inform continuous prompt optimization. Prioritize revision for high-impact prompts with the greatest variance or error incidence.
— *Use Sensitivity Analysis*: Regularly compute the local sensitivity of output uncertainty ($\Delta S \cdot \Delta O$) with respect to each prompt attribute. This identifies levers for greatest effect and prevents unnecessary increases in complexity or specificity that yield diminishing returns.

(3) Adopt Ensemble and Aggregation Strategies.
— *Deploy Model Ensembles*: For high-stakes or reliability-critical tasks, run prompts through multiple generative models (e.g., GPT-4o, Gemini 1.5 Pro, Claude 3 Sonnet), capturing independent samples per prompt.
— *Apply Sigma-Weighted Aggregation*: Combine outputs using empirically derived reliability weights, as demonstrated in Section 5.3. Penalize high-entropy responses and reward consensus, using coherence, factuality, or human-in-the-loop ratings as aggregation weights.
— *Automate Conflict Resolution*: For tasks with divergent responses (e.g., subjective or ambiguous prompts), define rules to prioritize responses with minimal variance and highest inter-model agreement. Route edge cases to manual review or escalate to downstream risk mitigation systems.

(4) Integrate Uncertainty Quantification Throughout the Pipeline.
— *Estimate and Log Both Prompt-Induced and Decoding-Induced Uncertainty*: For every output, compute entropy and variability statistics based on the model's probability distributions and response replicates. Track these metrics for both production and staging systems.
— *Enforce Dynamic Guardrails*: Set thresholds for maximum allowable entropy or output variance per application domain. Automatically reject, reroute, or post-process outputs exceeding predefined risk levels.
— *Incorporate Uncertainty Signals into Downstream Systems*: In workflows where AI output is post-processed or used in automated decision-making, propagate uncertainty scores as metadata. Allow downstream modules to adaptively modulate their trust, escalation, or review pathways.

(5) Calibrate System Parameters to Domain-Specific Risk and Cost Profiles.
— *Determine Acceptable Risk*: Map the system's error tolerance and cost of variability to application requirements. For instance, legal and medical domains require strict upper bounds on both ΔO and error rate; creative domains may tolerate greater variance.
— *Tune Ensemble Width and Aggregation Parameters*: Adjust the number of models sampled, aggregation weight thresholds, and prompt complexity to achieve required reliability within acceptable compute and latency budgets.
— *Document System Behavior Empirically*: Maintain clear, versioned records of reliability, error distributions, and uncertainty metrics for all critical deployments.

E.2 Critical Deployment Patterns: Case Examples

Case Study 1: Automated Customer Support Chatbots. A financial institution deploys a GPT-based support agent. Static prompt templates enforce domain terminology and block ambiguous references (e.g., "my account" is replaced with explicit account numbers, with type-checked user input). All customer queries pass through three model variants, and responses are Sigma-aggregated. Any

output exceeding the entropy threshold triggers a human review workflow, minimizing reputational risk and ensuring consistency.

Case Study 2: Medical Report Summarization. In a hospital, automated report summarization must be deterministic and precise. Prompts are strictly templated and validated, enforcing required clinical codes and removing distractor context (such as social history). Each report is run through two independent models. Aggregation weights prioritize factual coherence and penalize high variance. Reports with entropy above the clinical threshold are rejected or routed for manual approval, as uncertainty metrics are attached to every output and stored for audit.

Case Study 3: Legal Document Drafting. A law firm deploys LLM-driven document drafting. Prompt templates include mandatory clauses and prohibit speculative language. Prompt complexity is optimized via staged experiments: output variance is minimized at intermediate prompt detail, and over-complexification is avoided. The workflow integrates downstream semantic similarity checks between LLM outputs and ground-truth precedents, flagging drafts with high uncertainty or low semantic fidelity for partner review.

E.3 Summary of Operationalization Principles

The operationalization of the AI Uncertainty Principle and related theoretical constructs enables practitioners to architect generative AI systems with verifiable, cost-controlled, and robust behavior. Through treating prompt design, output aggregation, and uncertainty quantification as first-class engineering concerns, organizations can advance beyond experimental prototypes to scalable, trustworthy, and production-ready intelligent systems. Below is a list of the summary of our recommended operationalization principles.

- *Prompt Policies Must Precede Deployment.* Establish and enforce well-specified input standards and remove known sources of ambiguity.
- *Ensemble Decoding and Output Aggregation Provide Significant and Quantifiable Reliability Gains.* Sigma-weighted methods should be the default in any risk-sensitive workflow.
- *Comprehensive Uncertainty Estimation Is Mandatory.* Both prompt-induced and decoding randomness contribute to variability and must be logged and acted upon in downstream processes.
- *Continuous Monitoring and Adaptive Refinement Underpin Reliability.* Feedback, real-world performance data, and sensitivity analysis drive ongoing prompt and system improvements.
- *All Reliability Guarantees Are Empirical and Context-Specific.* Calibration to domain requirements, cost budgets, and risk profiles is essential; system behavior must be validated under real operating conditions.

Received 8 November 2024; revised 21 July 2025; accepted 23 July 2025