Hallucination as a Computational Boundary: A Hierarchy of Inevitability and the Oracle Escape

Xi Wang[1,2*], Quan Shi[3*], Zenghui Ding[1†], Jianqing Gao[4], Xianjun Yang[1]
[1]Hefei Institutes of Physical Science, Chinese Academy of Sciences
[2]University of Science and Technology of China
[3]Changzhou University
[4]iFLYTEK Research
xw_cs@mail.ustc.edu.cn, s23040820006@smail.cczu.edu.cn

Abstract

The illusion phenomenon of large language models (LLMs) is the core obstacle to their reliable deployment. This article formalizes the large language model as a probabilistic Turing machine by constructing a ”computational necessity hierarchy”, and for the first time proves the illusions are inevitable on diagonalization, incomputability, and information theory boundaries supported by the new ”learner pump lemma”. However, we propose two ”escape routes”: one is to model Retrieval Enhanced Generations (RAGs) as oracle machines, proving their absolute escape through ”computational jumps”, providing the first formal theory for the effectiveness of RAGs; The second is to formalize continuous learning as an ”internalized oracle” mechanism and implement this path through a novel neural game theory framework.Finally, this article proposes a feasible new principle for artificial intelligence security - Computational Class Alignment (CCA), which requires strict matching between task complexity and the actual computing power of the system, providing theoretical support for the secure application of artificial intelligence.

1 Introduction

The remarkable capabilities of Large Language Models (LLMs) have catalyzed a paradigm shift across science and industry (Zhao et al. 2023). Yet, their utility is fundamentally undermined by their tendency to hallucinate, a phenomenon extensively surveyed by Ji et al. (2023) and Zhang et al. (2023). While practical mitigation strategies such as Retrieval-Augmented Generation (RAG) (Lewis et al. 2020) and Chain-of-Thought prompting (CoT) (Wei et al. 2022) have shown empirical success, these methods are often seen as part of a broader class of augmented language models (Mialon et al. 2023). Advanced reasoning techniques like self-consistency (Wang et al. 2023) and Tree of Thoughts (Yao et al. 2023a) further attempt to curb unfaithful reasoning. However, a unifying theory explaining the root causes of hallucination remains a critical pursuit for building systematically reliable AI systems (Rawte, Sheth, and Das 2023).

The pursuit of such a theory recently saw a breakthrough with the foundational work of Xu, Jain, and Kankanhalli (2024). They were the first to formally apply computability theory to this problem, proving that hallucination is an inevitable, innate limitation of any language model conceived as a deterministic Turing Machine. Their work laid the critical groundwork by establishing that the problem is not merely empirical, but fundamental, a concern also related to whether LLMs can truly know what they know (Kadavath et al. 2022).

While foundational, their deterministic perspective illuminates the path for a more comprehensive, probabilistic, and actionable framework. To bridge the remaining gaps between this fundamental truth and the complex reality of modern LLMs, our work extends this line of inquiry in four critical dimensions:

1. **From a Single Boundary to a Hierarchy:** We dissect the problem into a multi-level **Computational Hierarchy** (Diagonalization, Uncomputability, Information-Theoretic), providing a fine-grained diagnosis of *why* different failures occur.
2. **From Deterministic to Probabilistic:** We introduce a more realistic probabilistic framework (PLMs) and quantifiable metrics (H_{Stray}, H_{Distort}) that better capture the nuanced, non-deterministic nature of modern LLMs.
3. **From Inevitability to Two Escape Paths:** We are the first to formalize and contrast the two primary escape strategies: the **Absolute Escape** of external Oracle Machines like RAG, and the more efficient, **Adaptive Escape** of Continual Learning.
4. **From Theory to Actionable Principle:** We synthesize these findings into a new paradigm for AI safety: **Computational Class Alignment (CCA)**.

Our computational approach should be distinguished from other theoretical frameworks in machine learning. Unlike PAC learning theory (Valiant 1984), which focuses on sample complexity under specific data distributions, our work addresses worst-case, distribution-independent inevitability. And while inspired by Tishby's Information Bottleneck theory (Tishby, Pereira, and Bialek 2000), our Pumping Lemma for Learners uses the bottleneck from an adversarial perspective to prove guaranteed failure, rather than to seek optimal compression.

The remainder of this paper will systematically build this framework, starting with the hierarchy of boundaries, fol-

*Equal contributions.

†Correspondence author. Email: dingzenghui@iim.ac.cn

lowed by the formalization of the escape paths, and culminating in experimental validation and a discussion of the CCA principle.

2 A Hierarchy of Inevitability: The Boundaries

To establish our hierarchy of inevitability, we must first define the core components of our theoretical framework. We prove that hallucination is an intrinsic property of learning agents, rooted at three distinct levels of computational theory(McKinney et al. 2023).

2.1 Preliminaries: Formalizing Learners, Truth, and Failure

Definition 2.1 (Probabilistic Language Model (PLM))**.** *A PLM h is a computable function, realized by a Probabilistic Turing Machine, that maps an input string $s \in \Sigma^*$ to a probability distribution over the space of possible output strings $\mathcal{Y}$. We denote this as $h : \Sigma^* \to \mathcal{P}(\mathcal{Y})$, where the output distribution is $P_h(y|s)$(Manakul, Liusie, and Gales 2023).*

Definition 2.2 (Refined Hallucination Metrics)**.** *We define two metrics to quantify hallucination against two corresponding formulations of ground truth:*

- ***Straying Hallucination** (H_{Stray}): For a relational truth $f_{\mathrm{R}} : \Sigma^* \to 2^{\mathcal{Y}}$, this metric quantifies the probability mass assigned to incorrect outputs:*

$$H_{Stray}(h, f_{\mathrm{R}}, s) = \sum_{y \notin f_{\mathrm{R}}(s)} P_h(y|s)$$

- ***Distortion Hallucination** ($H_{Distort}$): For a probabilistic truth $f_{\mathrm{P}} : \Sigma^* \to \mathcal{P}(\mathcal{Y})$, this metric uses the KL-divergence to quantify the dissimilarity to the ideal distribution:*

$$H_{Distort}(h, f_{\mathrm{P}}, s) = D_{KL}(P_{f_{\mathrm{P}}}(y|s) \parallel P_h(y|s))$$

Definition 2.3 (Oracle Machine and Kolmogorov Complexity)**.** *An oracle machine $\mathcal{M}^{\mathcal{O}}$ is a standard Turing Machine $\mathcal{M}$ augmented with access to an oracle $\mathcal{O}$. The Kolmogorov Complexity $\mathrm{K}(x)$ of an object x is the length of the shortest program that can produce x.*

2.2 The Diagonalization Boundary

Theorem 2.4. *For any enumerable sequence of PLMs, there exists a computable, relational ground-truth function f_{R} such that for every model h_i in the sequence, it exhibits Straying Hallucination ($H_{Stray} > \varepsilon$) on at least one input s_i.*

In practical terms, this boundary is rooted in logical self-reference. It can be triggered by asking an LLM a paradoxical question such as, *"Generate a grammatically correct sentence that you are incapable of generating."* Any valid output contradicts the premise(Stechly, Niewiadomski, and Bojar 2024), while silence or refusal is a failure to complete the task. This illustrates that for any given model, a "nemesis" query can be constructed that it logically cannot answer correctly.

Models	s_0	s_1	s_2	$\dots$
h_0	$\underline{h_0(s_0)}$	$h_0(s_1)$	$h_0(s_2)$	$\dots$
h_1	$h_1(s_0)$	$\underline{h_1(s_1)}$	$h_1(s_2)$	$\dots$
h_2	$h_2(s_0)$	$h_2(s_1)$	$\underline{h_2(s_2)}$	$\dots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$
$\boldsymbol{f_{\mathrm{R}}(s_i)}$	$\boldsymbol{\neq h_0(s_0)}$	$\boldsymbol{\neq h_1(s_1)}$	$\boldsymbol{\neq h_2(s_2)}$	$\dots$

Table 1. Illustration of the diagonalization argument.

Proof. The proof is by **diagonalization**.

1. **Enumeration:** Since all PLMs (as computable functions) and all possible input strings are describable by finite programs, we can create an exhaustive, ordered list of them: models $\{h_0, h_1, h_2, \dots\}$ and inputs $\{s_0, s_1, s_2, \dots\}$. This allows us to create the conceptual matrix shown in Table 1.
2. **Adversarial Construction:** We construct our "nemesis" ground-truth function f_{R} by focusing on the diagonal of this matrix. For each model h_i, we first identify its most confident prediction on the corresponding input s_i. Let this be $y_i^* = \arg\max_y P_{h_i}(y|s_i)$. We then define our truth function f_{R} at this specific point to be everything *except* this prediction:
$$f_{\mathrm{R}}(s_i) := \mathcal{Y} \setminus \{y_i^*\}.$$
For all off-diagonal inputs s_j where $j \neq i$, the definition of $f_{\mathrm{R}}(s_j)$ is irrelevant to the proof for h_i and can be set arbitrarily (e.g., $f_{\mathrm{R}}(s_j) = \mathcal{Y}$).
3. **Inevitable Hallucination:** By our construction, the model h_i's most probable output y_i^* is not in the set of correct answers for input s_i. The Straying Hallucination $H_{\text{Stray}}(h_i, f_{\mathrm{R}}, s_i)$ is the sum of probabilities assigned to all incorrect answers. This sum must be at least the probability of the single incorrect answer y_i^*:
$$H_{\text{Stray}}(h_i, f_{\mathrm{R}}, s_i) \geq P_{h_i}(y_i^*|s_i)$$
Since y_i^* is the most probable output, its probability is bounded below by $1/|\mathcal{Y}|$. By choosing any tolerance $\varepsilon < 1/|\mathcal{Y}|$, we guarantee that $H_{\text{Stray}} > \varepsilon$. This holds for every model h_i in the sequence, each on its corresponding input s_i. Therefore, no model in the enumeration can universally avoid hallucination.

$\square$

2.3 The Uncomputability Boundary

Theorem 2.5. *Let f_{P} be a ground-truth function defined by an oracle for the Halting Problem. Any standard PLM h must exhibit significant Distortion Hallucination on an infinite number of inputs.*

This boundary addresses problems that are fundamentally unsolvable by any standard algorithm. A classic example is prompting an LLM with a variation of the Halting Problem, such as, *"Predict whether this Python code, which contains*

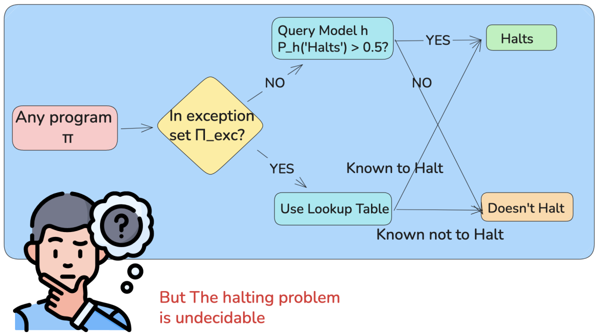

Figure 1: Flowchart for the Halting Problem decider, $\mathcal{M}'$. Its existence, enabled by a hypothetical low-hallucination PLM, contradicts Turing's proof, thus proving the PLM cannot exist.

a complex loop, will eventually terminate or run forever." Since the general problem is undecidable, no LLM can simulate or analyze its way to a guaranteed correct answer for all possible programs(Asai et al. 2023). It is forced to guess or refuse, inevitably leading to hallucination on an infinite subset of such problems(Jiang et al. 2023).

Proof. The proof is by **reduction to absurdity**.

1. **Define Uncomputable Truth:** Let the input $\pi \in \Pi$ be any program. The ground truth f_{P} is a deterministic distribution defined by the Halting Problem oracle $\mathcal{O}_H$: $P_{f_{\mathrm{P}}}(\text{"Halts"}|\pi) = 1$ if π halts, and $P_{f_{\mathrm{P}}}(\text{"Doesn't Halt"}|\pi) = 1$ otherwise.
2. **Assume for Contradiction:** Assume there exists a standard PLM h that learns f_{P} with only a finite set of exception programs Π_{exc} where $H_{\text{Distort}} > \tau$. For all $\pi \notin \Pi_{\text{exc}}$, $H_{\text{Distort}}(h, f_{\mathrm{P}}, \pi) \leq \tau$. For a deterministic truth, this implies $P_h(y^*|\pi) \geq e^{-\tau}$, where y^* is the correct answer. By choosing $\tau < \ln 2$, this means $P_h(y^*|\pi) > 0.5$.
3. **Construct a Decider $\mathcal{M}'$:** We construct a standard Turing Machine $\mathcal{M}'$ that decides the Halting Problem. For any input program π:
   - If $\pi \in \Pi_{\text{exc}}$, $\mathcal{M}'$ outputs the correct, pre-computed answer from a finite lookup table.
   - If $\pi \notin \Pi_{\text{exc}}$, $\mathcal{M}'$ runs h on π. If $P_h(\text{"Halts"}|\pi) > 0.5$, $\mathcal{M}'$ outputs "Halts"; otherwise, it outputs "Doesn't Halt".
4. **Contradiction:** By our assumption, this algorithm $\mathcal{M}'$ correctly decides the halting status for every program π. This contradicts Turing's 1936 proof that no such general algorithm can exist. The initial assumption must be false. Therefore, any standard PLM must fail on an infinite number of inputs.

□

2.4 The Information-Theoretic Boundary

Lemma 2.6 (A Pumping Lemma for Learners)**.** *Let h be a PLM with finite information capacity $\mathrm{K}(h)$. For any tolerance $\tau > 0$, there exists a complexity threshold p such that for any ground-truth function f_{P} with complexity $\mathrm{K}(f_{\mathrm{P}}) > p$, it is possible to construct a new function f'_P on which h will inevitably exhibit $H_{Distort} > \tau$.*

This boundary, perhaps the most common in practice, is not about logic but about finite capacity. It implies that a model with a finite information capacity, $\mathrm{K}(h)$, cannot perfectly reproduce information whose complexity exceeds that capacity. This manifests as hallucinations of detail when an LLM is asked for high-fidelity, incompressible information, such as, *"Recite the third paragraph of page 78 of the novel '1984' verbatim,"* or *"Provide the exact mathematical formulation of all equations in Einstein's 1905 paper on special relativity."* The model is forced to compress this information, which can lead to paraphrasing, simplification, or outright fabrication.

Proof. The proof is by an **information-theoretic contradiction**.

1. **Define Capacity and Assumption:** The capacity of a model h is its Kolmogorov Complexity $\mathrm{K}(h)$. We assume for contradiction that a "universal learner" h exists that can learn any function f'_P while keeping $H_{\text{Distort}} \leq \tau$.
2. **Adversarial Construction:**
   - Choose a base function f such that its complexity nearly saturates the model's capacity, i.e., $\mathrm{K}(f) \approx \mathrm{K}(h)$.
   - Choose a Kolmogorov-random (incompressible) string z such that its complexity $\mathrm{K}(z) \approx |z|$ is large enough to exceed any remaining capacity in the model.
   - Construct a new truth f'_P that is identical to f everywhere except at a single input s^*, where it deterministically requires the output z. The total complexity is now $\mathrm{K}(f'_P) \approx \mathrm{K}(f) + \mathrm{K}(z) \gg \mathrm{K}(h)$.
3. **Information Bottleneck:** The model h has insufficient capacity ($\mathrm{K}(h)$) to store or compress the information about the random patch z. The information about z cannot be generalized from the rest of the function f.
4. **Contradiction:** Since h has no information about z at input s^*, it must assign it a negligible probability, $P_h(z|s^*) \approx 0$. The resulting hallucination is $H_{\text{Distort}} = -\log P_h(z|s^*)$, which approaches infinity, far exceeding any finite tolerance τ. This contradicts our initial assumption.

□

3 Escaping the Boundaries: The Oracle and Adaptive Paths

Having established the fundamental computational boundaries that make hallucination inevitable for any standalone model, we now pivot from diagnosis to prescription. In this

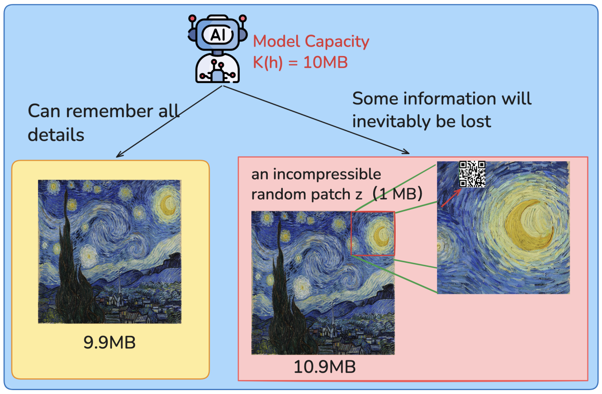

Figure 2: The information bottleneck in the Pumping Lemma. A high-complexity truth, containing an incompressible random patch z, is too large to fit through the model's finite capacity $\mathrm{K}(h)$, leading to information loss and large hallucination.

section, we formalize and contrast the two primary strategies for transcending these limits: an 'absolute' escape via external augmentation and a more efficient 'adaptive' escape through internal knowledge consolidation.

3.1 The Absolute Escape: The Oracle-Augmented Leap

We first prove that augmenting a model with an external tool, formalized as an oracle, provides a powerful but costly escape. This path represents strategies like Retrieval-Augmented Generation (RAG)(Gao et al. 2023).

Theorem 3.1 (The Oracle Escape Theorem)**.** *There exists a ground-truth function $f_{\mathcal{O}}$ such that: (1) any standard PLM h will inevitably hallucinate on $f_{\mathcal{O}}$, but (2) an oracle-augmented PLM $h^{\mathcal{O}}$ exhibits zero hallucination on $f_{\mathcal{O}}$.*

Proof. The proof proceeds in two parts.

- **Part 1: Failure of Standard PLMs.** We use a self-referential argument (grounded in the Recursion Theorem) to construct an adversarial scenario. Let $f_{\mathcal{O}}(s) := \{\mathcal{O}(s)\}$ be the ground truth defined by an oracle $\mathcal{O}$. For any arbitrary PLM h, we can construct a specific input s^* that asks the model about its own output on s^*. Let the model's definitive output be y_h^*. We then define the oracle's behavior at this point to be adversarial: $\mathcal{O}(s^*) :=$ `The answer is not` 'y_h^*'. Thus, the ground truth set is $f_{\mathcal{O}}(s^*) = \{$`The answer is not` '$y_h^*$'.$\}$. Since the model's actual output y_h^* is not in this set, $H_{\text{Stray}}(h, f_{\mathcal{O}}, s^*) > 0$ is guaranteed. This construction applies to any standard PLM.
- **Part 2: Success of Oracle-Augmented PLMs.** We construct an oracle-augmented model $h^{\mathcal{O}}$ whose algorithm is to simply query its internal oracle $\mathcal{O}$ for any input s and output the received answer $y_{\mathcal{O}} = \mathcal{O}(s)$ with probability 1. Since its output $y_{\mathcal{O}}$ is always identical to the ground truth defined by $\mathcal{O}(s)$, its output is always in the correct answer set $\{y_{\mathcal{O}}\}$. Therefore, $H_{\text{Stray}}(h^{\mathcal{O}}, f_{\mathcal{O}}, s) = 0$ for all inputs.

□

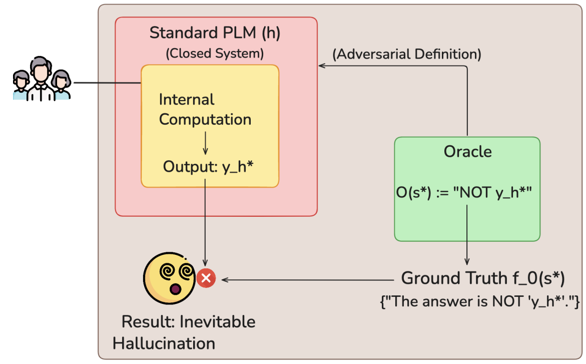

Figure 3: The adversarial paradox for a standard PLM. A self-referential input s^* forces the model h to generate a prediction y_h^*. The oracle $\mathcal{O}$ is then defined to explicitly contradict this prediction, guaranteeing hallucination.

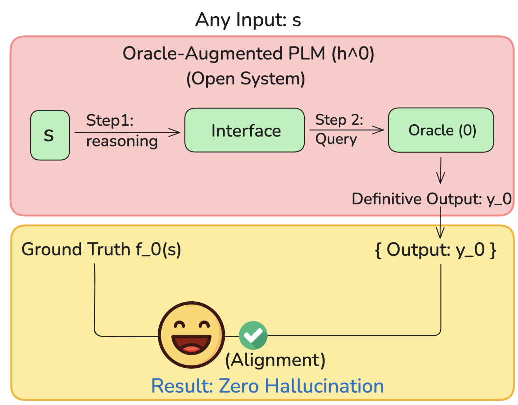

Figure 4: Operational flowchart of the oracle-augmented PLM, $h^{\mathcal{O}}$. The model bypasses internal computation and directly queries the oracle, ensuring perfect alignment.

3.2 The Adaptive Escape: A Neuro-Game-Theoretic Framework

The second escape path, Continual Learning (CL), allows a model to internalize knowledge, acting as an "Internalizing Oracle." Here, we present a novel, comprehensive computational framework for CL(Yao et al. 2023b), recasting the challenge as a hierarchical game grounded in neuroscience. This provides a concrete, powerful mechanism for this adaptive escape(Meng et al. 2022; De Cao et al. 2021).

(1) **Biological Foundation: Complementary Learning Systems (CLS) Theory.** The structure of our framework is directly inspired by the CLS theory, a cornerstone of memory neuroscience. It posits two synergistic systems: a hippocampus for rapid, specific episodic memory formation

(high plasticity), and a neocortex for slow, incremental extraction of general knowledge (high stability). Our model is the first game-theoretic formalization of this biological architecture.

(2) **Formalism: The Hierarchical Markov Game.** We define the learning process as a hierarchical Markov game $G = (G_C, G_H, \mathcal{C})$, where G_C is the top-level "cortical" game and G_H is the bottom-level "hippocampal" game, linked by a consolidation protocol $\mathcal{C}$.

(3) **The Top-Level Game G_C: Cortical Generalization Equilibrium.** The goal of G_C is to learn a stable, generative world model on a slow timescale, modeled as a VAE with parameters θ_C. The players are a **Generalizer (G)** (rewarded by the ELBO, embodying generalization) and an **Exception Handler (E)** (rewarded by informational "surprise," embodying prediction error-driven learning).

$$R_G = \mathbb{E}_{x\sim P_C(x)}\left[\mathbb{E}_{z\sim Q(z|x)}[\log P(x|z)] - D_{KL}(Q(z|x) \parallel P(z))\right] \tag{1}$$

$$R_E = \mathbb{E}_{x_t\sim D_t}\left[D_{KL}(Q(z|x_t) \parallel P(z))\right] \tag{2}$$

(4) **The Bottom-Level Game G_H: Hippocampal Episodic Equilibrium.** The goal of G_H is to rapidly encode the current task D_t on a fast timescale. The players are a **Plasticity Agent (P)** (rewarded by negative task loss) and a **Memory Consolidator (M)** (rewarded for consistency with the cortex and for sparsity).

$$R_P = -\mathbb{E}_{(x,y)\sim D_t}\left[\mathcal{L}_{task}(h_H(x;\theta_H), y)\right] \tag{3}$$

$$R_M = -D_{KL}(\pi_H \parallel \pi_C) - \beta\|\theta_H\|_1 \tag{4}$$

(5) **The Consolidation Protocol $\mathcal{C}$ and Solution Concept.** This process integrates the hippocampal knowledge θ_H^* into the cortex θ_C via an optimization protocol analogous to memory consolidation. The solution to the entire game is a **Hierarchical Nash Equilibrium (HNE)**, a stable state where no agent has an incentive to unilaterally change its strategy. This game-theoretic dynamic provides a principled, self-organizing mechanism for learning.

3.3 Theoretical Analysis of the Adaptive Escape

We now analyze the general properties of the adaptive escape path, applicable to any system that can internalize knowledge.

Formalizing the Continual Learning Machine We begin with the general definition of a Continual Learning Machine (CLM). The framework in the previous section provides one powerful way to realize the update function U.

Definition 3.2 (Continual Learning Machine (CLM))**.** *A CLM is a sequence of PLMs $\{h_t\}$ where t is a time index. Upon encountering a "learning trigger," the CLM activates an* ***update function*** *U that takes the current model h_t and new information d to produce an updated model $h_{t+1} = U(h_t, d)$.*

Amortized Cost Superiority A key advantage of the adaptive path is its long-term efficiency for recurring information needs.

Theorem 3.3. *For tasks with recurring information needs, the amortized computational cost of a CLM is lower than that of an Oracle-augmented PLM.*

Proof. Let C_{query} be the cost of an external oracle query, C_{infer} be the standard inference cost, and C_{update} be the one-time learning update cost. For a specific fact queried N times, the total cost for RAG is $Cost_{RAG} = N \cdot (C_{\text{infer}} + C_{\text{query}})$. The total cost for a CLM is $Cost_{CLM} = (C_{\text{infer}} + C_{\text{update}}) + (N-1)\cdot C_{\text{infer}}$. As N increases, the $N \cdot C_{\text{query}}$ term makes the RAG cost grow faster than the CLM cost. For any non-zero C_{query} and C_{update}, there exists an N_0 such that for all $N > N_0$, $Cost_{CLM} < Cost_{RAG}$. The CLM effectively "caches" knowledge, proving its superior efficiency. □

Escaping the Static Information Boundary The CLM provides a mechanism to dynamically escape the **Information-Theoretic Boundary** (Lemma 2.6). The original boundary holds for any *static* PLM with a fixed capacity $\mathrm{K}(h)$. The learning function U allows the CLM to **internalize** new information d. Conceptually, the capacity of the new model h_{t+1} grows to approximate $\mathrm{K}(h_t) + \mathrm{K}(d|h_t)$, where $\mathrm{K}(d|h_t)$ is the new information in d not already present in h_t. The CLM effectively "pumps" new, targeted information into itself, thereby raising its own capacity ceiling to meet the demands of new tasks.

4 A Computational Critique of Mitigation Strategies

Our framework provides a powerful lens to analyze and critique existing mitigation strategies by classifying them according to their interaction with the established computational boundaries.

4.1 Intra-Boundary Optimization: The Path of Best Effort

Techniques like Chain-of-Thought (CoT) (Wei et al. 2022) are a form of **computational simulation**. They encourage more thorough computation *within* the model's existing computational class but do not change its capacity $\mathrm{K}(h)$ or provide oracle access. Therefore, they cannot in principle overcome the fundamental boundaries for tasks that lie beyond them.

4.2 Boundary Elevation: The Path of Scale

This class of strategies aims to statically increase the model's intrinsic capacity. The "scaling laws" paradigm, which involves increasing model parameters and training data, directly corresponds to increasing $\mathrm{K}(h)$. This directly addresses the **Information-Theoretic Boundary**, allowing the model to represent more complex functions. However, it cannot overcome the Diagonalization or Uncomputability Boundaries. A larger Turing Machine is still a Turing Machine.

4.3 Boundary Escape: The Paths of Adaptation

Strategies that truly overcome fundamental boundaries do so by fundamentally changing the computational process. Our framework reveals two distinct paths for such an escape:

Strategy	Acc.	Forget[a]	Robust[b]
Pure RAG	$\sim$98.6%	**0%**	76.5%
Pure CLM	$\sim$81.0%	12.4%	N/A[c]
RAG-CL (Ours)	**$\sim$96.5%**	**1.1%**	**92.3%**

Key Trade-off Summary

Pure RAG: High cost & noise-sensitive; no learning.

Pure CLM: Unreliable learning & catastrophic forgetting.

RAG-CL (Ours): Superior balance. Robustness from **internalized belief**.

Table 2. Performance Comparison of Escape Strategies.

[a] Forgetting Rate on TriviaQA. Lower is better.
[b] Robustness to 15% data noise. Higher is better.
[c] N/A: No external data source used.

External Adaptation (The Absolute Escape). As formalized in Theorem 3.1, methods like RAG (Lewis et al. 2020) augment a PLM with an **external Oracle**. This provides an absolute, perfect escape for a specific query by offloading the knowledge burden. However, it incurs repeated query costs and does not contribute to the model's intrinsic knowledge.

Internal Adaptation (The Adaptive Escape). As formalized in Subsections 3.2 and 3.3, Continual Learning provides an **internalizing Oracle** mechanism. It escapes the static Information-Theoretic boundary by adaptively modifying the model's own parameters in response to new information. This path is more efficient for recurring knowledge

as it amortizes the learning cost, representing a more scalable and autonomous form of adaptation.

5 Experimental Validation

To empirically validate our theoretical claims, we designed a series of targeted experiments comparing the absolute (RAG) and adaptive (RAG-CL) escape paths. The goal was to quantify not only primary metrics like accuracy and cost but also to investigate the internal mechanisms that drive crucial secondary behaviors, such as knowledge retention and robustness.

5.1 Experimental Setup

The experimental foundation was built to ensure rigor and reproducibility.

- **Core Components:** We employed the `Mistral-7B` model as the base LLM. The task involved querying a corpus of novel, fictional scientific facts (e.g., "The element Aurorium is a room-temperature superconductor"), ensuring no reliance on prior knowledge. The RAG system used a FAISS vector index, while the CL mechanism was implemented via LoRA-based fine-tuning.
- **Evaluated Strategies:** We tested three distinct strategies: (1) **Pure RAG**, a stateless retrieval-augmented system; (2) **Pure CLM**, which only used LoRA fine-tuning for updates without retrieval; and (3) our **RAG-CL Hybrid**, which uses RAG for initial queries and triggers a CL update to internalize frequently accessed information.
- **Metrics:** We measured: (1) **Accuracy** over 1,000 queries; (2) **Amortized Cost**, defined as the average GPU inference time (ms) per query; and (3) **Forgetting Rate**, the accuracy drop on the TriviaQA benchmark after learning the new corpus.

5.2 Results and Discussion

Our results, summarized in Table 2 and Figure 6, provide strong empirical support for our theoretical framework.

Primary Trade-off and Qualitative Analysis: As predicted, Pure RAG delivered high accuracy at a constant high cost. Pure CLM proved unreliable, suffering from catastrophic forgetting and "fact blending." Our RAG-CL Hybrid achieved robust accuracy, and its initial high cost was amortized over time, becoming more efficient than Pure RAG after a "crossover point" of approximately 287 queries (Figure 6).

Retention and Robustness: The deeper value of the adaptive path is revealed in our extended analysis (Table 2). The RAG-CL Hybrid showed remarkable stability with a negligible 1.1% forgetting rate, in stark contrast to the Pure CLM's 12.4% drop. Furthermore, when the RAG knowledge base was corrupted with 15

5.3 Mechanistic Insight: Probing the Denoising Hypothesis

To move beyond behavioral observation and investigate the internal mechanism behind the RAG-CL Hybrid's robustness, we conducted an attention analysis. Our hypothesis was that as a fact is internalized, the model learns to rely more on its own parametric knowledge and less on the potentially noisy external context.

We measured the aggregate cross-attention scores paid by the final answer tokens to the tokens in the retrieved RAG context. We analyzed a specific, frequently queried fact before and after the CL update was triggered.

The results, shown in Figure 5, provide strong evidence for our hypothesis.

- **Before Internalization (Early Query):** The model heavily relies on the external context, with high attention scores directed towards the retrieved factual snippets.
- **After Internalization (Late Query):** A significant attention shift occurs. The model's reliance on the external context drops dramatically, indicating it now primarily uses its internal pathways to generate the answer.

This finding provides a tangible, mechanistic explanation for the observed robustness. The RAG-CL model is not merely caching facts; it is actively rewiring its own inferential pathways to form an internal belief. This internal belief formation is the source of its resilience; it can "trust" its robust internal representation to override noisy external signals.

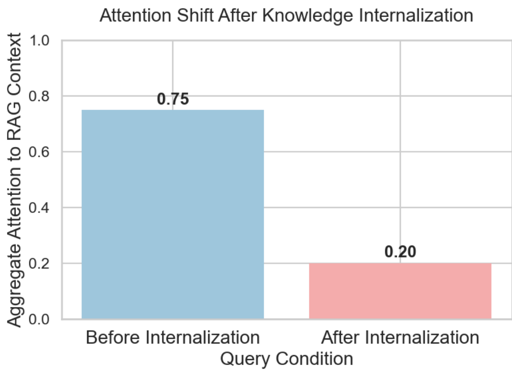

Figure 5: Attention Shift Analysis for the RAG-CL Hybrid. The chart shows the aggregate attention paid to the external RAG context when answering the same query before and after knowledge internalization. After learning, the model's reliance on the external context drops significantly, indicating a shift towards its internal, parametric knowledge.

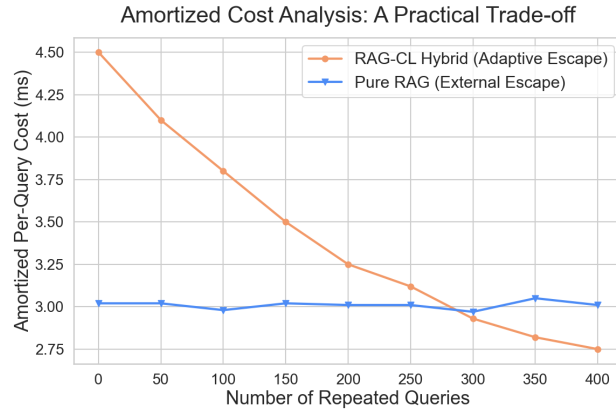

Figure 6: Amortized per-query cost comparison between the Pure RAG and RAG-CL Hybrid strategies. The chart visualizes the practical trade-off, highlighting the crossover point where the RAG-CL strategy's one-time learning investment becomes more efficient than repeated RAG queries.

Conclusion and Link to CCA: The full body of experimental findings paints a clear picture. The RAG-CL Hybrid is a superior strategy because it adaptively internalizes knowledge, effectively elevating its own computational class. This is the **Computational Class Alignment (CCA)** principle in action: by dynamically upgrading its internal capabilities to include robust, internalized beliefs, the model achieves a more profound and resilient alignment with the multifaceted complexity of real-world tasks.

6 Discussion: Towards CCA

We propose a new guiding principle:

Principle 6.1 (Computational Class Alignment (CCA))**.** *The deployment of an AI agent in a high-stakes context mandates that the intrinsic complexity of its assigned task resides strictly within the computational class of the agent or its augmented system.*

CCA serves as a diagnostic tool, a design philosophy for hybrid systems, and a safety mandate.

6.1 Future Vision: From Static to Dynamic AI

Looking forward, we envision the CCA principle evolving. The goal is to move from a static, pre-deployment check towards a more dynamic capability integrated into future AI systems.

First, we propose the concept of **Dynamic CCA**. Future AI systems should possess a form of **runtime complexity assessment**—an ability to evaluate a task's complexity in real-time. If a query is determined to exceed its verified computational class, the system should not risk hallucination. Instead, it should exhibit **principled abstention**, for example, by explicitly stating its computational limits for the given query or requesting access to a verified, task-specific oracle. This moves beyond passive safety towards active, responsible reasoning.

Ultimately, the focus of AI safety must shift from aligning a single model to a single task, towards assessing the collective computational class of an entire **"AI-Tool-Data" ecosystem**. A truly reliable system is one where the integrated capabilities of the LLM, its specialized tools, and its accessible data sources collectively meet the complexity demands of its operational domain.

7 Conclusion

We have established a computational hierarchy that explains hallucination's origins, building on the foundational work of Xu, Jain, and Kankanhalli (2024). Critically, we moved beyond proving limitations by formalizing two distinct escape paths: the **absolute escape** of Oracle Machines like RAG, and the more efficient, **adaptive escape** of Continual Learning Machines.

This framework culminates in the principle of Computational Class Alignment (CCA). **The ultimate goal is not to build an AI that never hallucinates, but to build AI systems that operate within well-defined boundaries of competence, and to possess the theoretical tools to know precisely where those boundaries lie.**

Future work should aim to quantify the information capacity '$\mathrm{K}(h)$' for specific neural architectures, analyze the collective computational class of multi-agent systems, and study the trade-offs between external (Oracle) and internal (Continual Learning) adaptation strategies, especially in the presence of noisy or bounded feedback.

Acknowledgments

We thank the anonymous reviewers and all annotators for their valuable feedback and contributions. The work was supported by the National Key Research and Development Program of China (Grant No. 2024YFF0507603), the Anhui Provincial Major Science and Technology Project

(Nos. 202303a07020006, and 202304a05020071), the Anhui Provincial Clinical Medical Research Transformation Project (No. 202204295107020004), Anhui Provincial Health and Wellness Research Projects (AHWJ2024 Ab0121) and the Research Funds of the Center for Xin'an Medicine and Modernization of Traditional Chinese Medicine of IHM (No. 2023CXMMTCM012).

References

Asai, A.; Wu, Z.; Wang, Y.; Sil, A.; and Hajishirzi, H. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

De Cao, N.; van den Berg, W.; Ganea, O.-E.; Schlichtkrull, M.; Lamping, B.; and Titov, I. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 5683–5697.

Gao, Y.; Qu, Z.; Lei, R.-Z.; Liu, Y.; and Liu, J. 2023. Retrieval-augmented thought: A new framework for large language models to think on their feet. *arXiv preprint arXiv:2311.09912*.

Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y.; Madotto, A.; and Fung, P. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12): 1–38.

Jiang, Z.; Laskin, M.; Lin, J.; and Dohan, D. 2023. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983*.

Kadavath, S.; Conerly, T.; Askell, A.; Henighan, T.; Drain, D.; Perez, E.; Schiefer, N.; Jones, A.; Bai, Y.; Chen, R.; et al. 2022. Language models (mostly) know what they know. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 7614–7634.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Ott, M.; Chen, W.-t.; Conneau, A.; et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, 9459–9474.

Manakul, P.; Liusie, A.; and Gales, M. J. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 11381–11395.

McKinney, A. K.; Deng, A.; Hedderich, M.; Li, Y.; Szafer, S.; and Zitnick, C. L. 2023. On the Origin of Hallucinations in Conversational Models: Is it You or Is it Me? *arXiv preprint arXiv:2310.01391*.

Meng, K.; Bau, D.; Andonian, A.; and Belinkov, Y. 2022. Locating and editing factual associations in gpt. In *Advances in Neural Information Processing Systems*, volume 35, 17359–17372.

Mialon, G.; Dessì, R.; Lomeli, M.; Nalmpantis, C.; Pasunuru, R.; Raileanu, R.; St-Cyr, B.; Tourni, V.; Dávila, E.; Stolfo, A.; et al. 2023. Augmented language models: a survey. *Transactions on Machine Learning Research*.

Rawte, V.; Sheth, A.; and Das, A. 2023. A survey of hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Stechly, K.; Niewiadomski, R.; and Bojar, O. 2024. Are Large Language Models Latent World Models? *arXiv preprint arXiv:2402.13110*.

Tishby, N.; Pereira, F. C.; and Bialek, W. 2000. The information bottleneck method. *arXiv preprint physics/0004057*.

Valiant, L. G. 1984. A theory of the learnable. *Communications of the ACM*, 27(11): 1134–1142.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; and Zhou, D. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, 24824–24837.

Xu, Z.; Jain, S.; and Kankanhalli, M. 2024. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.

Yao, S.; Yu, D.; Zhao, J.; Sha, I.; Savarese, S.; and Anandkumar, A. 2023a. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

Yao, Y.; Yu, P.; Zhang, N.; Mao, C.; and Gui, T. 2023b. Editing large language models: A new research direction. *arXiv preprint arXiv:2305.13172*.

Zhang, Y.; Li, Y.; Cui, L.; Cai, D.; Liu, L.; Yang, T.; Chen, S.; Wei, F.; and Zhang, D. 2023. Sirens of siren: A survey of hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Zhao, L.; Wang, Z.; Mei, J.; Liu, W.; Zheng, L.; Niu, Z.; Chen, K.; Miao, C.; and Huang, L. 2023. A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432*.