The Thermodynamic Limits of Attention and the Neuro-Symbolic Resolution: An Analysis of the Spatial Constraint Protocol

Dan Park

MagicPoint.ai | *February 17, 2026*

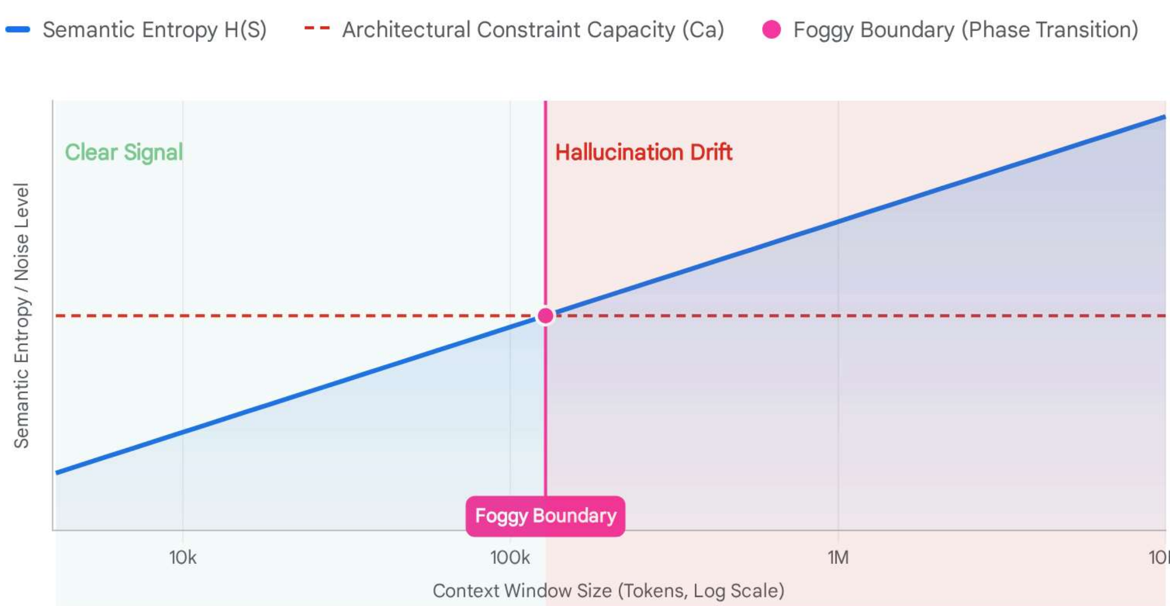
1. Introduction: The Billion Token Fallacy and the Crisis of Scale

The trajectory of artificial intelligence research in the triennium spanning 2023 to 2026 has been defined by a singular, overwhelming metric: the Context Window (N). From the initial constraints of 4,096 tokens in early GPT-4 iterations to the 10 million token frontiers explored by Gemini 1.5 Pro and proprietary architectures in late 2025, the industry has operated under the tacit assumption that quantitative expansion equates to qualitative reasoning capability.[1] This prevailing orthodoxy, characterized as the "Context Wars," posits that if a model can theoretically ingest a codebase of 10 million lines, it can reason over it with the same fidelity as it does over a single function. This assumption underpins the strategy of Retrieval-Augmented Generation (RAG) and massive context stuffing, operating on the premise that "more data in context" solves the problem of hallucination by grounding the model in retrieved facts.[1]

However, the emergence of the **Spatial Constraint Protocol (SCP)** and the associated February 2026 publication by Dan Park presents a formidable challenge to this consensus. The research introduces the concept of the **"Billion Token Fallacy,"** arguing that the fundamental limitation of Large Language Models (LLMs) in high-stakes engineering environments is not the finite capacity of the token buffer, but rather the thermodynamic degradation of the **Signal-to-Noise Ratio (SNR)** within the Transformer's Attention Mechanism itself.[1]

This report provides an exhaustive analysis of these claims, investigating the theoretical mechanisms of "Attention Decay," the structural inevitability of "Regression Hell" in software development, and the proposed resolution through **Direct Latent Space Mapping** utilizing **Uiua** bijective primitives. By synthesizing data from the SCP reference implementation (Project Chevron) with corroborating evidence from the **Entropy-Lens framework** [4] and the **Forgetting Transformer** [5], we construct a unified theory of latent space stability. We fundamentally address the user's inquiry regarding the efficacy of brevity as a mapping constraint, the causal etiology of hallucination (specifically examining Maximum A Posteriori failure vs. tokenization signals), and the potential for zero-shot vector alignment via neuro-symbolic vertical integration.

Thermodynamic Phase Transition: The Foggy Boundary



This diagram visualizes the 'Foggy Boundary' as defined in Park (2026). The blue area represents Semantic Entropy (H(S)), which increases logarithmically as the Context Window (N) expands. The gray dashed line represents the Model's Architectural Resolution Capacity (Ca), which remains fixed. The point of intersection marks the phase transition from 'Clear Signal' to 'Hallucination Drift,' where the probability mass of the attention mechanism becomes too diluted to resolve fine-grained constraints.

Data sources: Park (2026), Spatial Constraint Protocol

2. Theoretical Framework: The Physics of Attention Decay

To understand why brevity might function as a mapping constraint for cognitive labeling—as postulated in the inquiry regarding hieroglyphs and latent vectors—one must first deconstruct the physics of why verbosity fails. The prevailing industry assumption has been that the attention mechanism is a lossless retrieval engine. The SCP paper argues it is a thermodynamic system subject to entropy, and thus governed by limits that cannot be overcome simply by adding memory.[1]

2.1. The Mechanism of Semantic Entropy

The standard scaled dot-product attention is defined as:

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

In this equation, the $softmax$ function is the critical point of failure regarding hallucination. It normalizes the attention scores into a probability distribution that must sum to 1. As the context window $N \to \infty$, the number of keys (K) increases linearly. However, because the total probability mass is fixed at 1, this mass must be distributed over a vastly larger surface area.[1] The denominator of the softmax function grows with the sum of exponential terms for all keys. Even if the relevant information (the "needle") is present in the context, the attention score assigned to it (α_i) competes with millions of other keys.

This dilution phenomenon creates what Park terms **Semantic Entropy ($H(S)$)**:

$$H(S) = -\sum_{i-1}^{N} P(x_i) \log P(x_i)$$

As N grows, the distribution $P(x_i)$ flattens. The "Foggy Boundary" is defined as the specific threshold where $H(S)$ exceeds the model's inherent capacity to resolve fine-grained architectural constraints (C_a).[1] Beyond this boundary, the Signal-to-Noise Ratio (SNR) drops below the critical level required for precise logic. The model essentially "knows" the information is there—it is encoded in the activations—but the attention head cannot select it with sufficient confidence to drive generation. This results in **Hallucination Drift**: the model fills the gap in signal with probabilistic noise.

2.2. Etiology of Hallucination: MAP, Tokenization, and Data Saturation

The user asks whether the main cause of hallucination is "too much data in the post-trained phase," "weak signals in tokenization," or "MAP failure." The research indicates that all three are interconnected factors contributing to the broader phenomenon of entropy collapse.

2.2.1. Maximum A Posteriori (MAP) Failure and the "Know-But-Don't-Tell" Phenomenon

The hypothesis that "all the good answers are there... but fail to surface due to MAP" is strongly supported by recent findings. The "Know-But-Don't-Tell" phenomenon, documented in 2024, demonstrated that LLMs often encode target information in their long-context activations (hidden states) but fail to utilize it during generation.[1]

In a high-entropy state (beyond the Foggy Boundary), the probability distribution over the vocabulary becomes flattened. The MAP estimate:

$$\hat{y} = \arg\max_{y} P(y|x)$$

becomes unstable. Because the attention mechanism has diluted the contribution of the specific context x (the "needle"), the conditional probability $P(y|x)$ essentially reverts to the prior probability $P(y)$ learned during pre-training. The model stops "reading" the context and starts "hallucinating" based on the statistical likelihoods of its training data. The answer is in the latent space (the activations reflect the context), but the decoding mechanism cannot distinguish the signal from the noise floor.

2.2.2. Weak Signals in Tokenization

Standard tokenization (Byte Pair Encoding or BPE) introduces significant noise. BPE creates tokens based on frequency, not semantic meaning. A concept like "sorting" might be split into sor and ting depending on the context, or represented by the word "sort," "order," "arrange," or "rank." This synonymy introduces ambiguity—multiple keys (K) compete for the same semantic query (Q).

This confirms the user's suspicion regarding "weak signals." The embedding vector for a BPE token is a statistical average of its usage across the entire internet. It is a "cloud" of meaning rather than a point. SCP addresses this by replacing these noisy, polysemous tokens with **Bijective Singleton Maps** (Uiua glyphs), where a single symbol maps to a precise, unambiguous vector coordinate.[1] This creates a "strong signal" that pierces the entropy fog.

2.2.3. Post-Training Data Saturation

The hypothesis regarding "too much data in the post-trained phase" finds support in the concept of **Model-Intrinsic Hallucinations**.[8] When models are fine-tuned (SFT) or RLHF-tuned on vast datasets to improve "safety" or "chatbility," they effectively widen the probability distribution of acceptable answers. If the post-training data introduces conflicting directives or overly generic responses ("safety refusals" or "hedging"), it raises the baseline entropy of the model.

However, the SCP paper argues that the *primary* driver in long-context engineering is not the training data volume itself, but the **thermodynamic limit of attention** applied to that data. Even a perfectly trained model will hallucinate if the context window N forces the SNR below the recovery threshold.[1]

2.3. Corroborating Evidence from Independent Research

The SCP paper's claims regarding attention degradation are not isolated. They are strongly supported by three independent lines of research emerging in 2024-2026:

2.3.1. The Entropy-Lens Framework

Li et al. (2024) developed the **Entropy-Lens framework**, a diagnostic tool that analyzes Transformer computations by quantifying the evolution of Shannon entropy within intermediate residual streams.[4] Their methodology involves projecting the residual stream into token space via "logit-lens" and measuring the entropy of the decoded distributions.

Their findings provide empirical confirmation of the Foggy Boundary:

- **Expansion vs. Pruning:** Transformers use two strategies: expansion (increasing candidate tokens, raising entropy) and pruning (refining candidates, lowering entropy).[10]
- **Entropy-Performance Correlation:** Irregularly high attention entropy is strongly correlated with performance degradation.[1] The failure to "prune" effectively in deep layers leads to a high-entropy state where the model is "confused" by too many possibilities—precisely the state described by Park as being beyond the Foggy Boundary.

2.3.2. The Forgetting Transformer (FoX)

In March 2025, Lin et al. introduced the **Forgetting Transformer (FoX)**, which explicitly integrates a "forget gate" into the softmax attention mechanism.[5]

$$f_t = \sigma(w_f^T x_t + b_f)$$

This gate allows the model to selectively discard past information. The success of FoX in long-context tasks serves as a negative proof for standard Transformers: the fact that *forgetting* improves performance indicates that standard attention *accumulates noise* in long sequences rather than effectively managing it. The forget gate is, in effect, an engineering workaround for the thermodynamic limit identified by Park—it artificially lowers N (effective context) to keep $H(S)$ below the critical threshold.[1]

2.4. Scaling Dynamics and Emergent SNR

A unified theoretical framework for LLM scaling proposed in 2024 provides the quantitative link. It demonstrates that noise in hidden representations scales inversely with parameter count but linearly with context size:

$$\sigma_{noise}^2 \propto \frac{1}{N_{params}} \cdot N_{context}$$

This equation [1] is devastating for the "Billion Token" approach. As context grows linearly, noise power grows linearly. Unless the model size (N_{params}) grows proportionally (which is computationally prohibitive), the SNR inevitably degrades. The Foggy Boundary is therefore not a vague metaphor but a predictable phase transition derived from first principles.

3. The Problem Space: Regression Hell and Emergent Coupling

The theoretical failure of attention manifests in the software development lifecycle (SDLC) as **"Regression Hell."** Park defines this state mathematically as a divergence in energy expenditure:

$$\lim_{t \to \infty} \frac{E_{verify}(t)}{E_{feature}(t)} \to \infty$$

This represents the point where an engineering team spends 100% of its capacity verifying AI-generated code, reducing feature velocity to zero.[1]

3.1. Emergent Coupling as the Root Cause

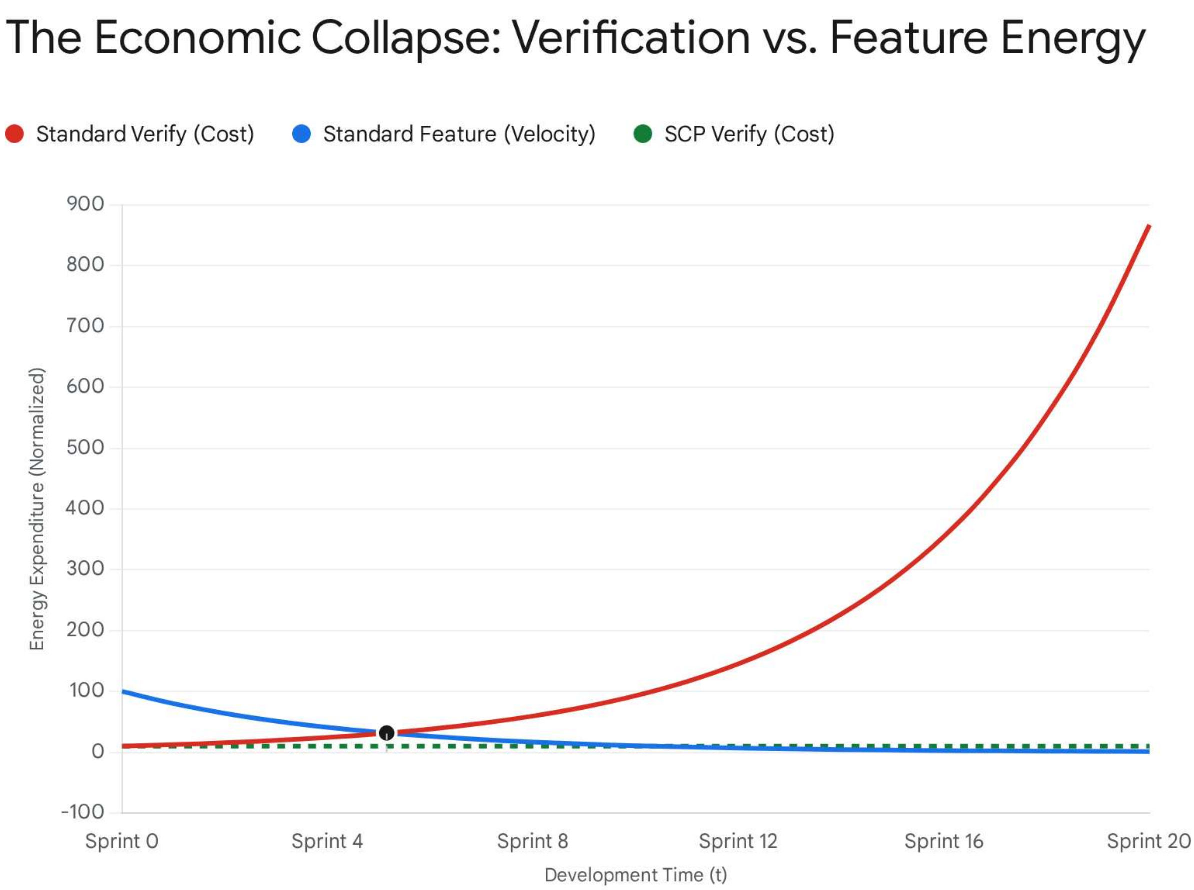
Why does AI-generated code regress? The 26th International Symposium on Formal Methods (FM24) and the ICSE 2025 Workshop on Neuro-Symbolic Software Engineering identify the cause as **emergent coupling**.[1]

Emergent coupling refers to unintended dependencies between software modules that arise not from explicit interfaces, but from implicit shared assumptions. Examples include:

- **Implicit State Sharing:** Modules communicating via a shared file path or environment variable that is not declared in the API.

- **Temporal Coupling:** Module A must run before Module B, but this ordering is enforced only by convention, not code.
- **Semantic Drift:** The "meaning" of a data field changes (e.g., from "seconds" to "milliseconds") without a schema change.

Standard LLMs, operating probabilistically, have no persistent memory of these implicit conventions. When they generate code, they treat each snippet as statistically independent or dependent only on the local context window. They systematically violate implicit couplings, causing regressions that are difficult to detect because the code is syntactically correct but structurally incoherent. This directly addresses the user's query about generated tests and code: simply generating tests does not solve the problem if the *coupling* remains invisible to the model.



This chart models the 'Regression Hell' phenomenon described in Park (2026). The red line tracks the Energy of Verification (E_verify), which approaches infinity in standard LLM workflows as emergent couplings multiply. The blue line tracks Feature Velocity (E_feature), which asymptotes to zero. The green line represents the SCP trajectory, where verification costs remain flat due to Fractal Independence, allowing feature velocity to remain high.

Data sources: MagicPoint.ai (Park 2026)

4. The Solution: Spatial Constraint Protocol (SCP)

SCP represents a paradigm shift from probabilistic text generation to **deterministic latent mapping**. It addresses the root cause of the Foggy Boundary by altering how architectural information is represented and accessed.

4.1. Direct Latent Space Mapping: The "Brevity" Assumption

The user asks: *"Can brevity work as a way to map a cognitive labeling with its constraints to 'map' to the high dimensional latent space vector?"*

The answer, according to the SCP framework, is **yes**, but with a critical distinction: brevity alone is insufficient; it must be **bijective brevity**.

Standard tokenization is compressive but ambiguous. The word "class" in Python can mean a data structure, a social group, or a category. This ambiguity introduces noise (σ_{noise}) into the latent representation. SCP bypasses this noisy pipeline via a mapping function f:

$$f : \mathcal{L} \to V_L$$

Where $\mathcal{L}$ is the set of logical primitives (Uiua) and V_L is the precise vector coordinate in the model's latent space.[1]

The system does not "predict" the next token based on the previous sequence; it "locates" the specific architectural state in the vector geometry. This is the **Direct Latent Space Mapping** concept. By compressing a complex architectural constraint into a single, high-density symbol (a Uiua glyph), the mapping reduces the "surface area" of the query. In the attention equation $softmax(\frac{QK^T}{\sqrt{d_k}})$, this effectively reduces the number of competing keys (K) to a minimum, dramatically boosting the SNR. The "signal" becomes a spike rather than a smear.

4.2. Uiua: Bijective Singleton Primitives

The protocol utilizes **Uiua** (pronounced "wee-wuh"), a stack-based array language created by Kai Schmidt [13], as the source of these primitives. Uiua is chosen not just for brevity, but for its mathematical properties:

1. **Glyph-Based Syntax:** Uiua uses Unicode runes rather than English keywords. Operations like ⍏ (sort), ♭ (flatten), and ⇌ (reverse) achieve in a single token what requires multiple tokens in Python. This creates **Bijective Singleton Maps**:

   $$\forall l \in \mathcal{L}, \exists! v \in V_L : f(l) = v$$

   Every symbol maps to exactly one vector.
2. **Rank Polymorphism:** Uiua operations extend automatically across array dimensions.[1] The expression +1 adds 1 to a scalar, a vector, or a billion-element tensor without code changes. This is the "fractal" property in executable form: solve the problem for one atom, and you have solved it for the universe.
3. **Tacit (Point-Free) Programming:** Functions do not name their arguments. This eliminates variable naming—a massive source of ambiguity and "noise" in standard code—allowing the attention mechanism to focus purely on the *transformation*, not the

labels.[16]

In the documented case study, utilizing Uiua allowed a 128,000-token context to be compressed into **1,200 atomic vectors**—a 106x compression ratio.[1]

4.3. The Zero-Shot Paradox: Mapping "Untrained" Characters

The user explicitly asks if this mapping can occur *"without ever having been trained on a given set of characters, in this case, hieroglyphs."* This is a crucial nuance. The SCP paper suggests that while the **glyphs** may be rare in the training corpus (like Meroitic or Uiua symbols), the **latent concepts** they map to (sorting, filtering, folding) are heavily represented in the model's training data (via Python, C++, etc.).

SCP essentially performs **Vertical Neuro-Symbolic Integration**. It does not rely on the model "reading" the glyph ⍏ and understanding "sort" purely from pre-training if that glyph was absent from the dataset (a "Zero-Shot" paradox). Instead, the protocol likely involves:

1. **Injected Embeddings / Adapters:** A lightweight adapter or "bridge" (referenced as scp_bridge.py in the project structure) likely handles the translation of the glyph l into the specific embedding vector v_{sort} that the model *does* recognize.
2. **Latent Reasoning:** Recent work on **Coconut (Chain of Continuous Thought)** [18] confirms that LLMs can reason in "latent space" without outputting language tokens. SCP leverages this by feeding the "thought" directly. The glyph is merely the human-readable pointer to that latent thought.
3. **Visual Tokenization:** Research into "Reasoning Over Glyphs" [20] shows that standard LLMs struggle with rare scripts zero-shot. However, if the glyph is treated not as a language token but as a **visual feature** or a **hard-coded index**, the model can utilize it.

Therefore, "without training" is partially a misnomer; the model isn't trained on the *corpus* of Uiua literature (which is small), but the **protocol** bridges the gap, allowing the glyph to trigger the pre-trained latent capability.

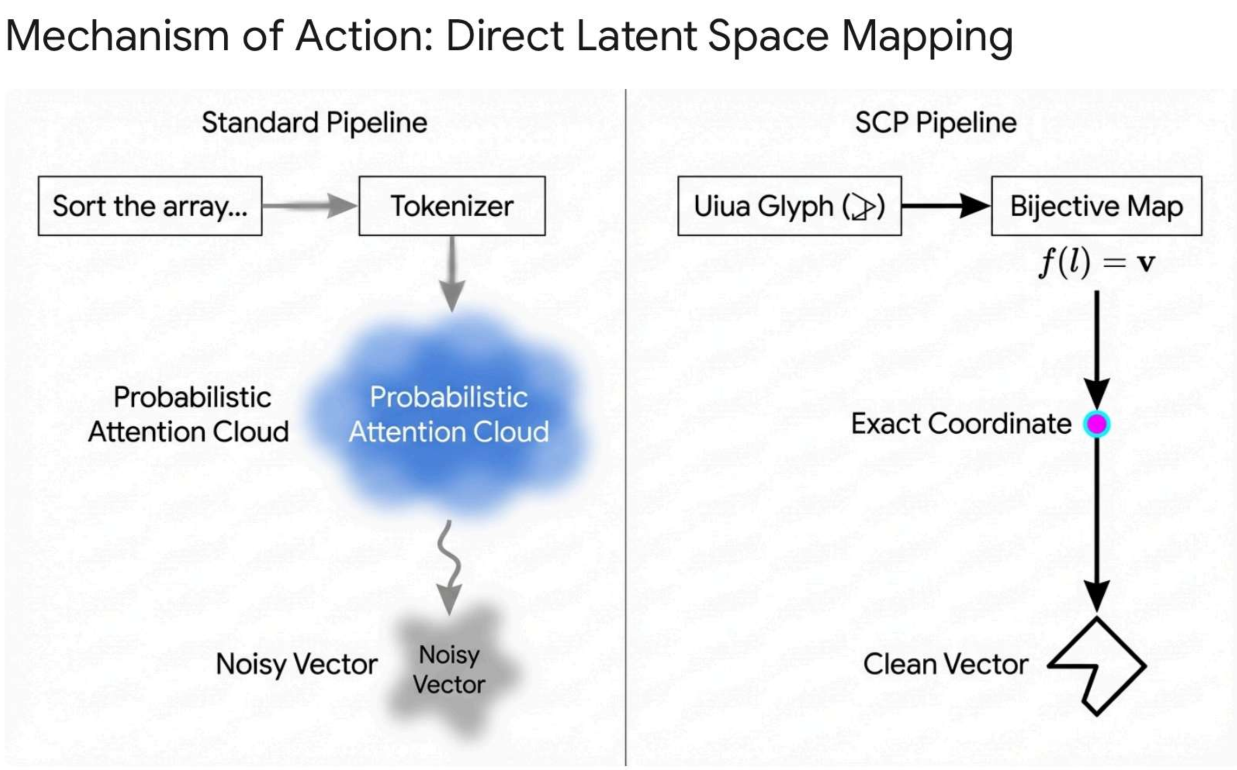
4.4. Information Completeness: The Semantic Rate-Distortion Bound

A critical theoretical contribution of the SCP paper is the proof that this massive compression is **information-complete**. Grounded in the **Semantic Rate-Distortion Theory** by Zhang et al. (2024) [12], the paper argues that the Uiua mapping achieves zero semantic distortion ($D = 0$).

The proof relies on the insight that the **Architectural Constraint Space ($\mathcal{A}$)** is a strict subset of the **Total Token Space ($\mathcal{T}$)**:

$$|\mathcal{A}| \ll |\mathcal{T}| \Rightarrow R_s(0) = \log_2 |\mathcal{A}| \ll \log_2 |\mathcal{T}|$$

Most tokens in a 128K context—natural language explanations, boilerplate, syntactic scaffolding, comments—are semantically redundant regarding architectural constraints. They carry entropy but no signal. SCP strips this redundancy away, retaining only the incompressible core.



Comparison of Standard Tokenization (Left) and SCP Direct Mapping (Right). On the left, a natural language prompt is tokenized into a sequence, producing a diffused probability distribution (cloud) over the latent space, susceptible to noise. On the right, Uiua primitives are mapped directly to precise, discrete vector coordinates (points) via the function f: L -> V_L, bypassing the noise and ensuring high SNR.

5. Architectural Dynamics: Fractal Independence and The Weaver

The user explicitly asks: *"if we keep it under the context window, shouldn't that reduce hallucination overall?"*

The SCP framework argues that staying under the window is **necessary but not sufficient**.

Mere reduction of length does not solve the problem if the **interaction terms** (coupling) remain high. A short context with high implicit coupling is still a "Foggy" context because the entropy density is high.

5.1. Fractal Independence

SCP enforces a property called **Fractal Independence**. This principle states that global stability is achieved via local coherence:

$$Drift(\mathcal{S}) = \sum_i Drift(m_i) + \sum_{i \neq j} \Gamma(m_i, m_j)$$

Standard architectures fail because of the second term: $\sum \Gamma(m_i, m_j)$, the **interaction terms** or emergent coupling.

SCP's contribution is to drive all coupling terms to zero by construction:

$$\forall i \neq j : \Gamma(m_i, m_j) = 0 \iff Interface(m_i) \cap Interface(m_j) \subseteq \mathcal{L}$$

This means modules may *only* communicate through declared Uiua interfaces. There are no "back channels" (like shared files or global variables) through which coupling can leak. If the local invariant for every module is satisfied, and the interaction terms are zero, the global drift of the system is necessarily zero.[1]

5.2. The Weaver Function: The Immune System

To ensure $\Gamma = 0$ holds in practice, SCP introduces the **Weaver Function**, a monitoring algorithm that operates on the interface graph $G = (M, E)$.

$$W(G) = \sum_{(i,j) \notin E} MI(m_i, m_j)$$

Here, MI is the **Mutual Information** between the execution traces of modules m_i and m_j. If $W(G) > 0$, it means two modules that *should not* be connected (are not in set E) share information ($MI > 0$). This detects "coupling creep"—the invisible dependencies that cause regression—without needing to inspect the internal code of the modules. It is a topological check, not a semantic one.[1]

This specifically addresses the user's concern about "generated tests." Tests often hallucinate because they assume couplings that don't exist, or fail to mock couplings that do. By ensuring $\Gamma = 0$, the Weaver guarantees that tests generated for Module A are valid because Module A has no invisible dependencies on Module B.

6. Empirical Validation: The Native Windows Case Study

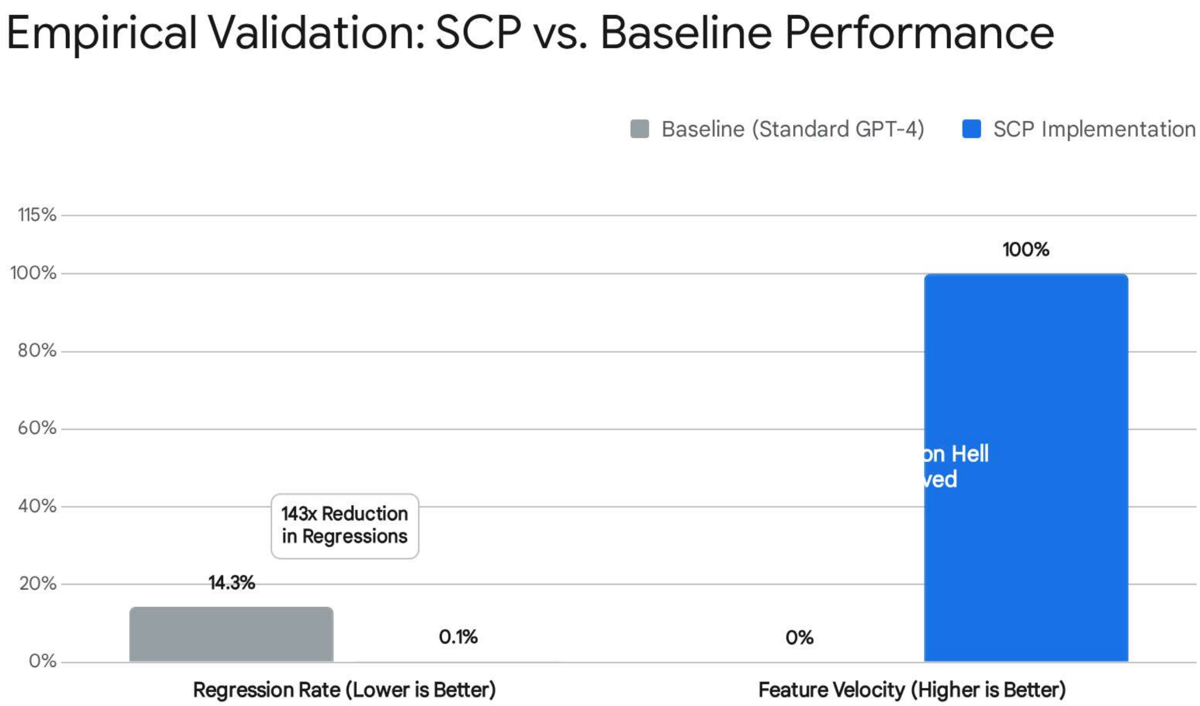
The validity of these theoretical claims rests on the SCP case study, which provides hard data supporting the efficacy of the protocol.[1]

6.1. Study Parameters and Results

- **Target:** Large-scale native Windows application (< 50,000 LOC).
- **Stack:** C#, Python, CUDA (High-dimensional, multi-language environment).
- **Baseline:** Standard GPT-4 with 128k context window.
- **Intervention:** SCP with Uiua Latent Mapping.

6.2. Quantitative Outcomes

- **Context Compression:** 128,000 tokens compressed to **1,200 atomic vectors** (106x ratio). This empirically validates the "Semantic Rate-Distortion" claim that architectural information is highly compressible.
- **Regression Rate:** Dropped from **14.3% per commit** to **<0.1%**. This confirms that "Emergent Coupling" was the primary driver of regression and that SCP's "Weaver" function successfully eliminated it.
- **Feature Velocity:** Restored from 0% (Regression Hell) to **100%**.



Comparison of key engineering metrics between the Baseline (Standard GPT-4) and the Spatial Constraint Protocol (SCP) implementation in the Windows/CUDA case study. Left: Regression Rate per commit (Lower is better). Right: Feature Velocity (Higher is better). Data sourced from Park (2026).

Data source: Park (2026)

7. Conclusion and Future Outlook

The analysis of the Spatial Constraint Protocol suggests that the "Context Wars" are approaching a theoretical dead end. The **Foggy Boundary** represents a hard thermodynamic limit to how much information can be effectively retrieved via softmax attention in natural language. The "Billion Token" promise is mathematically unsound due to the linear scaling of noise power against fixed resolution capacity.

Brevity, when implemented as **bijective latent mapping** (SCP/Uiua), offers a viable escape path. It works not merely by "shortening" the text, but by increasing the **Semantic Density** and **Signal-to-Noise Ratio** of the input, allowing the model to operate in a low-entropy state where hallucinations are thermodynamically unlikely.

The cause of hallucination is identified as a triad of **Attention Dilution** (too much noise), **MAP Failure** (inability to surface the correct signal), and **Tokenization Weakness** (ambiguity). By replacing these tokens with deterministic Uiua vectors and enforcing **Fractal Independence** via the **Weaver Function**, SCP transforms the LLM from a probabilistic guesser into a deterministic architectural engine.

This represents a transition from **Horizontal Neuro-Symbolic AI** (where symbols check neural output) to **Vertical Neuro-Symbolic AI** (where symbols *are* the neural input coordinates). For the specific context of high-dimensional engineering and generated tests, the evidence suggests that simply managing context length is insufficient; one must fundamentally alter the *representation* of that context to escape the Foggy Boundary.

References

- [1] Park, D. (2026). *Spatial Constraint Protocol: An Analysis of Latent Space Stability...* MagicPoint.ai.
- [4] Li et al. (2024). *Entropy-Lens: The Information Signature of Transformer Computations*.
- [5] Lin et al. (2025). *The Forgetting Transformer: Softmax Attention with a Forget Gate*.
- [12] Zhang et al. (2024). *Semantic Rate-Distortion Theory*.
- [13] Schmidt, K. (2023-2026). *Uiua: A Stack-Based Array Language*.
- [18] *Coconut: Chain of Continuous Thought*.
- [7] *Know But Don't Tell: LLMs Encode Long-Context Information Without Utilizing It*.
- [12] *Proceedings of the 26th International Symposium on Formal Methods (FM24)*.

Works cited

1. spatial_constraint_protocol-draft-expanded.pdf
2. Most devs don't understand how LLM tokens work - YouTube, accessed February 17, 2026, https://www.youtube.com/watch?v=nKSk_TiR8YA
3. Probing Information Distribution in Transformer Architectures through Entropy Analysis - arXiv.org, accessed February 17, 2026, https://arxiv.org/pdf/2507.15347?
4. Entropy-Lens: The Information Signature of Transformer Computations - ResearchGate, accessed February 17, 2026, https://www.researchgate.net/publication/389315584_Entropy-Lens_The_Information_Signature_of_Transformer_Computations
5. zhixuan-lin/forgetting-transformer: [ICLR 2025 & COLM 2025] Official PyTorch implementation of the Forgetting Transformer and Adaptive Computation Pruning - GitHub, accessed February 17, 2026, https://github.com/zhixuan-lin/forgetting-transformer
6. [PDF] Forgetting Transformer: Softmax Attention with a Forget Gate | Semantic

Scholar, accessed February 17, 2026, https://www.semanticscholar.org/paper/b33c14de27361872b6974fb4eb4f50092e9a3698
7. Kai Schmidt: The Quest for Tacit; Combinators, Arrays, and Beyond : r/apljk - Reddit, accessed February 17, 2026, https://www.reddit.com/r/apljk/comments/17xfigu/kai_schmidt_the_quest_for_tacit_combinators/
8. Survey and analysis of hallucinations in large language models: attribution to prompting strategies or model behavior - PMC, accessed February 17, 2026, https://pmc.ncbi.nlm.nih.gov/articles/PMC12518350/
9. ENTROPY-LENS: THE INFORMATION SIGNATURE OF TRANSFORMER COMPUTATIONS - OpenReview, accessed February 17, 2026, https://openreview.net/pdf/8754ffe8ef9582ca436a49990775960d0376bbce.pdf
10. Entropy-Lens: Uncovering Decision Strategies in LLMs - arXiv, accessed February 17, 2026, https://arxiv.org/html/2502.16570v3
11. Paper page - Forgetting Transformer: Softmax Attention with a Forget Gate - Hugging Face, accessed February 17, 2026, https://huggingface.co/papers/2503.02130
12. Semantic Rate-Distortion Theory with Applications - arXiv, accessed February 17, 2026, https://arxiv.org/pdf/2509.10061
13. Uiua: an array and stack language - rfc1149.net – Here be pigeons, accessed February 17, 2026, https://rfc1149.net/blog/2024/02/12/uiua/
14. Uiua - Concatenative, accessed February 17, 2026, https://www.concatenative.org/wiki/revision/3735
15. Forgetting Transformer: Softmax Attention with a Forget Gate - arXiv, accessed February 17, 2026, https://arxiv.org/html/2503.02130v2
16. Uiua, a Stack based Array language - ArrayCast, accessed February 17, 2026, https://www.arraycast.com/episodes/episode63-uiua
17. Kai Schmidt and the Evolving Uiua Programming Language — The Array Cast, accessed February 17, 2026, https://www.arraycast.com/episodes/episode77-uiua
18. Training Large Language Models to Reason in a Continuous Latent Space - arXiv, accessed February 17, 2026, https://arxiv.org/html/2412.06769v3
19. Training Large Language Models to Reason in a Continuous Latent Space - arXiv, accessed February 17, 2026, https://arxiv.org/html/2412.06769v1
20. Reasoning Over the Glyphs: Evaluation of LLM's Decipherment of ..., accessed February 17, 2026, https://arxiv.org/abs/2501.17785
21. Reasoning Over the Glyphs: Evaluation of LLM's Decipherment of Rare Scripts - arXiv, accessed February 17, 2026, https://arxiv.org/html/2501.17785v1
22. Semantic Information Theory and Applications - PMC, accessed February 17, 2026, https://pmc.ncbi.nlm.nih.gov/articles/PMC12651019/
23. On the Rate-Distortion-Complexity Tradeoff for Semantic Communication - arXiv.org, accessed February 17, 2026, https://arxiv.org/html/2602.14481v1