HDT²: A Pilot Framework for Entropy-Band Calibration of LLM Reasoning Stability

Public Research Release – Academic Use Only

Author: Bruce Tisler
Institutional Affiliation: Quantum Inquiry Research Initiative
Correspondence: brucetisler@quantuminquiry.com

Manuscript Version: Pre-print v1.0
repository-code: https://github.com/btisler-DS/hdt2-entropy-band-calibration
doi: "**10.5281/zenodo.17621326**"

Author's Statement

This manuscript is released publicly for academic research, peer review, and educational use. The HDT² framework is intended as open scientific infrastructure—a measurement system, not a product. Researchers are encouraged to study, replicate, critique, and extend the work under the conditions of the CC BY-NC-SA 4.0 license.

Commercial use, resale, integration into proprietary systems, or embedding in commercial products is prohibited without explicit written permission from the author. This preserves academic openness while preventing unauthorized productization or private capture of the methodology.

Keywords

LLM uncertainty · entropy alignment · reasoning stability · hallucination detection · cross-model calibration · cognitive uncertainty · control-theoretic diagnostics

HDT^2 provides the first falsifiable evidence that token-entropy geometry can be normalized across models to produce a portable, read-only diagnostic for reasoning stability—revealing both where the method works and where it fundamentally does not.

Abstract

Large language models (LLMs) often generate fluent but incorrect statements, yet there is no reliable, model-agnostic way to detect when a model's reasoning is becoming unstable. We present HDT^2, a pilot framework for entropy-based reasoning diagnostics that aims to provide a portable, read-only uncertainty geometry across models. The method derives entropy bands ψ^* from a reference model's token-level Shannon entropy and aligns other models into this geometry through an unsupervised affine calibration procedure (UNSUP_H_ALIGN) using quantile matching.

We instantiate the framework using four operational operators—Ω (observe entropy), Δ (orient relative to ψ^*), Φ (classify state), and Ψ (act via continuation or escalation)—and evaluate it on four instruction-tuned LLMs from the Qwen and Mistral families. ψ^* is constructed from Qwen 2.5-14B; alignment is then attempted on three target models. One model (Mistral-Nemo-Instruct-2407) satisfies pre-defined alignment thresholds ($\leq 5\%$ max error, $\leq 3\%$ median error of the reference interquartile range) and undergoes a seven-gate validation protocol assessing control correctness, entropy variance reduction, accuracy correlation, escalation behavior, overhead, and reproducibility. Two models fail alignment, providing empirical evidence that entropy geometry is architecture- and scale-dependent.

On the aligned model, ψ^* yields stabilized entropy trajectories, reliable escalation at extreme bands, and higher correctness rates for stable-band outputs—all without modifying model parameters. HDT^2 does not claim to solve hallucination; instead, it offers falsifiable evidence that entropy-band calibration can serve as a portable diagnostic substrate for LLM reasoning stability. All code and artifacts are released for replication and critique.

1 Introduction

Large language models (LLMs) achieve impressive performance across a wide range of tasks, yet remain prone to producing fluent but incorrect statements—commonly referred to as *hallucinations*. While substantial work has focused on training-time or architecture-level mitigation, the ability to *detect* when a model's reasoning is becoming unstable remains limited. Existing detection strategies often rely on model-specific confidence signals such as logit-based perplexity estimates, sampling variance, or domain-specific classifiers. These approaches do not generalize well across models, architectures, or scales, and therefore cannot provide the foundation for a transferable, deployment-ready uncertainty monitor.

This work explores a different perspective: **token-level Shannon entropy during generation encodes a measurable signature of reasoning stability**. Prior studies have investigated entropy as a proxy for confidence, but primarily within single-model settings. What remains unknown is whether *relative* entropy structure—its distribution, shape, and critical thresholds—can be calibrated *across different models* to form a shared uncertainty geometry. If such calibration is possible, it would offer a pathway toward portable, model-agnostic mechanisms for detecting reasoning instability.

This paper presents **HDT²**, a pilot framework for cross-model uncertainty calibration built on three key components:

1. **Entropy Bands (ψ*)**
   Quantile-derived regions of "stable," "risky," and "extreme" uncertainty, defined on a reference model and used as the common target for other models.
2. **UNSUP_H_ALIGN**
   An unsupervised entropy-alignment protocol that fits an affine transform between a target model's entropy distribution and the reference ψ* geometry—without labels or task supervision.
3. **7-Gate Validation Framework**
   A functional testbed analogous to a pre-flight checklist: each gate evaluates whether calibrated uncertainty signals behave as expected (e.g., triggering escalation when entropy exceeds thresholds, reducing variance, preserving accuracy on stable items, etc.).

We evaluate the framework in a pilot-scale study across two families of instruction-tuned models (Qwen 2.5 and Mistral-Nemo). Results show that one model—Mistral-Nemo-Instruct—achieves successful alignment with ψ* and passes all seven validation gates. Two smaller models fail alignment, revealing a possible boundary condition: cross-model entropy calibration may require shared architectural or training properties. Importantly, the 7-gate framework surfaced real implementation errors during testing, demonstrating that structured uncertainty diagnostics can meaningfully constrain development.

Personal Motivation

For me, this question of *detecting instability* is not abstract. As someone with dyslexia, I have spent decades paying close attention to subtle markers of when my own reasoning becomes unstable—when visual processing falters and I must shift modalities to preserve clarity. That lived experience shaped the design of HDT²: a system that observes uncertainty, contextualizes it, classifies its state, and adjusts its behavior accordingly. While this paper confines itself to

LLM behavior, the underlying intuition is the same: cognitive systems, biological or artificial, often signal the edges of their competence through measurable uncertainty patterns.

Contributions

This pilot study contributes:

- **A transferable entropy-band framework (ψ*)** for cross-model reasoning stability detection.
- **An unsupervised calibration method (UNSUP_H_ALIGN)** requiring no labels or task-specific data.
- **A reproducible 7-gate validation protocol** grounded in functional behavior rather than architectural assumptions.
- **A complete transparency package** including calibration artifacts, failures, debugging steps, and executable code.
- **Empirical evidence** that cross-model entropy calibration is feasible under certain conditions and fails under others—providing constraints for future theory.

The goal of HDT² is not to solve hallucination detection, but to **establish empirical footing for entropy-based reasoning diagnostics** and to present the first evidence that such diagnostics may generalize beyond a single model.

2 Related Work & Conceptual Background

LLM hallucination, uncertainty estimation, and calibration are widely studied problems, yet most existing approaches remain model-specific or require supervision. This section surveys relevant lines of work and situates HDT² within them.

2.1 Uncertainty Estimation in Large Language Models

Logit-Based Confidence and Perplexity

Token log-probabilities and derived measures (e.g., perplexity, entropy, variance) are the most common proxies for confidence in generative models. Prior work typically uses:

- **Per-token logit entropy** as a soft confidence estimate
- **Sequence-level perplexity** as a fluency or coherence indicator
- **Sampling variance** across temperature sweeps or stochastic decodes
- **Calibration curves** evaluating alignment between predicted probabilities and empirical correctness

These methods are *intra-model*—they measure uncertainty within a single model's output space. They do not generally transfer across architectures due to differences in vocabulary distribution, training corpus, scaling behavior, and logit geometry.

Ensemble and Monte Carlo Approaches

Another family of methods quantifies uncertainty by inducing diversity:

- **MC Dropout**
- **Temperature-based ensembles**
- **Stochastic sampling under fixed prompts**

These approaches are compute-intensive and again model-specific: the "spread" of samples reflects architectural variance, not a standardized uncertainty scale.

Classifier-Based Hallucination Detection

Some recent work applies supervised classifiers trained to detect hallucinations from:

- Hidden-state patterns
- Output text features
- Rationale consistency

However, these require labeled datasets and retraining for each model or domain.

Gap: None provide a *portable*, *model-agnostic* uncertainty signal.

2.2 Entropy as an Uncertainty Metric

Shannon entropy remains a canonical measure of uncertainty in probabilistic systems. For LLMs, token-level entropy captures:

- **Breadth of the model's predictive distribution**
- **Ambiguity in token choice**
- **Divergence from high-confidence states**

Prior work has shown correlations between high entropy and hallucination likelihood, but primarily for:

- A single model
- A fixed task domain
- A fixed decoding setup

What is unknown is whether the **shape** of entropy distributions is stable across different model families, and if not, whether they can be brought into alignment via post-hoc transformation.

2.3 Calibration Methods in Machine Learning

Temperature Scaling and Platt Scaling

In classification models, calibration techniques such as:

- **Temperature scaling**
- **Platt scaling (logistic calibration)**
- **Isotonic regression**

are used to align probability estimates with empirical accuracy. These methods require *labeled* validation data and typically operate on **logits or softmax outputs**.

LLM uncertainty is more difficult:

- Generative, not classification
- Structured outputs rather than single labels
- Token-level dependencies
- No agreed-upon "gold" uncertainty signal

Our Distinction

HDT² departs from prior work in two ways:

1. **Unsupervised:** It aligns token-entropy distributions using quantile matching—no labels, no correctness data.
2. **Cross-model:** It attempts to place *different* models into a shared entropy geometry anchored by a reference ψ* distribution.

This is closer to *distributional alignment* than probability calibration.

2.4 Cross-Model Behavior Alignment

There is emerging interest in finding universal signals that extend across model families:

- **Logit-lens analyses** show shared intermediate representations in transformers.
- **Behavioral probing** reveals that structurally different models may exhibit similar gradients of hesitation, uncertainty, or instability under specific prompt conditions.
- **Unsupervised representation alignment** (e.g., via linear probes) suggests that many LLMs exhibit partially compatible internal geometries.

HDT² extends this line of inquiry by treating **entropy itself** as the alignment substrate—testing whether models can be brought into a shared uncertainty space through affine transformation and quantile constraints.

2.5 Structured Cognitive Uncertainty Signals

A smaller body of work explores uncertainty as a dynamic signal rather than a static value:

- **OODA loops (Observe–Orient–Decide–Act)** formalize adaptive decision workflows.
- **Cognitive architectures** (e.g., Soar, ACT-R) treat uncertainty as part of reasoning control.
- **Metacognition research** studies how biological systems detect their own limits.

HDT² draws methodological inspiration from these systems—not as biological claims, but as **design patterns**: treating uncertainty as an operational signal that can trigger routing, stabilization, or escalation.

This connection motivates the Ω–Δ–Φ–Ψ operator structure:

- **Ω:** Observe entropy
- **Δ:** Orient relative to ψ* bands
- **Φ:** Classify the state
- **Ψ:** Act (proceed, adjust, escalate)

In this paper, these operators serve purely as organizing principles for the calibration and gating pipeline.

2.6 Positioning HDT²

HDT² is best understood as a **control-theoretic approach** to uncertainty:

- ψ* provides the *reference distribution*
- UNSUP_H_ALIGN provides the *mapping*
- Gates 1–7 provide the *validation constraints*
- Routing and actuation provide the *control outputs*

Unlike prior hallucination-detection methods, HDT² does **not**:

- train new classifiers
- access hidden states
- rely on ensembles
- require labels
- alter the model's parameters

It is a **post-hoc, read-only**, model-agnostic mechanism designed to evaluate whether calibrated entropy bands can serve as indicators of reasoning stability.

This work provides the first empirical evidence—albeit in a pilot-scale setting—that:

1. Some models *can* be aligned to ψ*
2. Others cannot
3. Successful alignment enables reproducible behavioral predictions under the 7-gate framework

3 The HDT^2 Framework

HDT^2 provides a structured, model-agnostic method for identifying reasoning instability through calibrated token-level entropy signals.

It consists of three core components:

1. **ψ*** — Reference entropy bands derived from a chosen model
2. **UNSUP_H_ALIGN** — An unsupervised alignment procedure mapping a target model into ψ*
3. **Ω–Δ–Φ–Ψ operator pipeline** — A control-style structure for observation, orientation, classification, and action

This section describes each component in detail.

3.1 Operational Roles of the Ω–Δ–Φ–Ψ Operators

The Ω–Δ–Φ–Ψ operators function as an *organizational scaffold* for the calibration and validation process.

They impose no architectural requirements on the model and do not alter generation.

Instead, they specify how entropy is interpreted and acted upon.

Ω (Observe)

Ω represents direct measurement of token-level entropy:

$$H_t = -\sum_{i=1}^{k} \quad p_{t,i} \log{}_2\, p_{t,i}$$

where $p_{t,i}$ is the renormalized probability of token (i) among the top-k logits at timestep (t).

Ω collects all raw entropy traces:

$$\{H_1, H_2, \dots, H_T\}.$$

Δ (Orient)

Δ situates the observed entropy values relative to the ψ* band structure.

Given a band partition:

- stable
- risky_low
- risky_high
- extreme

Δ computes:

$$\text{state}(H_t) = \text{bin}(H_t \mid \psi^{\backslash *})$$

This stage provides context: "Where does this entropy value sit relative to expected ranges?"

Φ (Classify)

Φ translates the banded state into a decision category:

- **stable → continue**
- **risky → monitor closely**
- **extreme → escalate or halt**

Formally:

$$\Phi(H_t) = f(\text{state}(H_t); \psi^{\backslash *})$$

The function (f) is deterministic, bounded, and independent of model internals.

Ψ (Act / Reflect)

Ψ specifies the downstream behavior that occurs when Φ identifies instability:

- allow generation to proceed
- adjust decoding parameters
- terminate generation and route to a fallback (human, safer model, etc.)
- log the event for forensic transparency

In this work, Ψ is restricted to **read-only control**:

- No logits modified
- No tokens inserted
- No gradients or fine-tuning
- Actuation only changes external process state (e.g., “stop generation”)

This ensures that HDT² does not interfere with the model's internal dynamics.

3.2 Reference Entropy Bands (ψ*)

ψ* encodes the empirical uncertainty structure of a designated **reference model**.
It is derived by sampling entropy values from diverse, neutral text contexts.

Band Definition

Let Q_p denote the p-th quantile of the entropy distribution.
We define:

$$\psi^{\backslash *} = \begin{cases} \text{stable: } [Q_{25}, Q_{75}] \\ \text{risky\_low: } [0, Q_{25}) \\ \text{risky\_high: } (Q_{75}, Q_{90}] \\ \text{extreme: } (Q_{90}, \infty) \end{cases}$$

These thresholds are not semantic categories; they are purely statistical partitions that become meaningful *after* alignment.

Empirical ψ* from Qwen 2.5-14B

For the reference model used in this study:

- **stable:** 0.0837–2.1661 bits
- **risky_low:** 0–0.0837 bits
- **risky_high:** 2.1661–2.7049 bits
- **extreme:** >2.7049 bits

These specific values are not universal; only the **structure** of ψ* is fixed.

3.3 UNSUP_H_ALIGN: Unsupervised Cross-Model Entropy Alignment

UNSUP_H_ALIGN postulates that, for sufficiently similar transformer models, entropy distributions differ primarily by *linear distortion*, and can therefore be aligned via an affine mapping:

$$H' = aH + b.$$

Purpose

- Align target model entropy to ψ*
- Require **no labels**, **no task data**, and **no intervention in model internals**

- Enable ψ* to function as a common uncertainty geometry for different LLMs

Calibration Procedure

Let D_{ref} be the entropy distribution of the reference model on the calibration shard, and D_{tgt} the distribution for the target model.

We fit scalars (a, b) by minimizing quantile deviation across a fixed set of anchors (here: Q25, Q50, Q75):

$$a, b = \arg\ \min \sum_{p \in \{25,50,75\}} \mid Q_p(D_{\text{ref}}) - (aQ_p(D_{\text{tgt}}) + b) \mid .$$

Alignment Validation

A target model is considered **aligned** if:

- **max quantile error ≤ 5%** of ref range
- **median quantile error ≤ 3%** of ref range

This becomes **Gate 0**:

If a model cannot be aligned, downstream interpretation of ψ* is not meaningful.

Empirical Outcomes

In this study,

- Mistral-Nemo-Instruct satisfied alignment thresholds
- Qwen 2.5-7B and Mistral-7B did not

These failures are treated as **data**, not defects: they define the boundary of ψ* generalizability.

3.4 Policy and Actuation Layer

Once entropy is aligned to ψ*, Φ assigns bands and Ψ determines how the system should behave.

State Classifier

```
def classify(H, psi_star):
    if psi_star.stable.low <= H <= psi_star.stable.high:
        return "stable", "expected_correct"
    elif H > psi_star.risky_high:
        return "risky_high", "expected_uncertain"
    else:
        return "risky_low", "expected_uncertain"
```

This rule is intentionally simple:

- deterministic
- monotonic
- architecture-independent
- no hidden heuristics

Actuation Logic

In this work Ψ may:

- **continue** (stable)
- **monitor** (risky)
- **escalate** (extreme)

Escalation triggers a **hard external stop**:

```
max_new_tokens := 0
route_to_human := true
```

All actuation events are written to transparent logs for later inspection.

3.5 Interpretation: Control-Theoretic Structure

Viewed abstractly:

- ψ^* serves as a **reference distribution**
- UNSUP_H_ALIGN provides a **mapping** into that space
- Φ provides **state estimation**
- Ψ provides **routing control**
- The 7-gate framework provides **formal constraints** for testing correctness

The key methodological contribution is that **none of this requires access to model weights or training data**: it is fully post-hoc and read-only.

4 Experimental Setup

This section describes the computational environment, data sources, sampling procedures, calibration methodology, and evaluation metrics used in this pilot study. All design choices emphasize reproducibility and transparency rather than scale.

4.1 Infrastructure

All experiments were conducted on a private inference cluster configured as follows:

- **Server platform:** SimplePod GPU instance
- **GPUs:** 4× NVIDIA A40
- **Inference engine:** vLLM (commit hash recorded in artifact bundle)
- **Batching:** Disabled for entropy sampling (one-request-per-run)
- **Framework:** Python 3.11, HuggingFace Transformers (for tokenizer standardization)
- **Determinism:**
    - `seed = 1234` fixed for all model runs
    - `top_k = 20`, renormalized for entropy
    - Greedy decode unless otherwise specified

This environment ensures that entropy traces depend only on the model's forward pass, not on cluster-level scheduling or nondeterministic parallelism.

4.2 Models Evaluated

Two families of instruction-tuned models were selected:

1. **Qwen 2.5 family**
    - Qwen2.5-14B-Instruct (reference model)
    - Qwen2.5-7B-Instruct (alignment fail)
2. **Mistral family**
    - Mistral-Nemo-Instruct-2407 (alignment pass)
    - Mistral-7B-v0.3-Instruct (alignment fail)

These models were chosen to test whether ψ* and UNSUP_H_ALIGN generalize across:

- parameter counts
- training corpora
- RLHF procedures
- tokenizer differences

No model was fine-tuned, modified, or compensated; all runs use public checkpoints.

4.3 Phase C: ψ* Reference Distribution Construction

Data Source

To derive ψ*, we sampled the reference model (Qwen2.5-14B-Instruct) on a **neutral text shard** containing:

- 5,632 characters
- mixture of encyclopedia narrative, expository prose, and general-domain text
- no adversarial or domain-specialized content

The shard is included in the artifact bundle, with SHA-256 fingerprint recorded.

Sampling Procedure

- **Max tokens:** 1 (single-token probes)
- **Prompts:** Deterministic slices of the shard
- **Greedy decode:** `temperature = 0.0`
- **Top-k distribution:** `k = 20`, renormalized
- **Entropy:** Shannon entropy in **bits**, per token

This produces an entropy sample set:

$$D_{\mathrm{ref}} = \{H_1, \dots, H_N\}, N \approx 4096.$$

ψ* bands were computed from the quantiles (Q25, Q50, Q75, Q90) of this distribution.

4.4 UNSUP_H_ALIGN Calibration on Target Models

For each target model, we collected entropy using the **same shard** and the **same sampling procedure** to produce:

$$D_{\mathrm{tgt}} = \{H_1', \dots, H_M'\}.$$

We then computed the affine mapping:

$$H'' = aH' + b$$

by aligning quantiles Q25, Q50, Q75 of D_{tgt} to those of D_{ref}

Alignment Thresholds

A model is considered aligned if:

- **max quantile error ≤ $\mathbf{5\%}$** of $(Q_{75} - Q_{25})_{\mathrm{ref}}$
- **median quantile error ≤ $\mathbf{3\%}$**

These thresholds were **declared before experimentation** and not tuned.

Outcome Summary

Model	Alignment Result	Max Error	Median Error
Qwen2.5-14B	Reference	0.0%	0.0%
Mistral-Nemo-2407	PASS	3.8%	1.9%
Qwen2.5-7B	FAIL	8.4%	4.8%
Mistral-7B-v0.3	FAIL	11.2%	6.9%

Only models passing this gate proceed to behavioral validation.

4.5 Prompt Set for Behavioral Evaluation

For Gate 1–7 evaluation, we used a **60-prompt set** containing 12 task families:

- factual QA
- commonsense reasoning
- chain-of-thought math
- short analytic tasks
- definition expansion
- summarization
- analogy generation
- classification
- reasoning under uncertainty
- instruction following
- safety-neutral tasks
- open-domain prompts

The prompt set is not intended to be exhaustive; it is purpose-built to produce varied entropy dynamics across short generations.

Labeling for Accuracy-Based Gates

For gates requiring correctness (Gate 3, Gate 6):

- correctness was labeled manually
- binary scoring was used (correct / incorrect)
- ambiguous or multi-validity responses were excluded

This keeps accuracy metrics interpretable at pilot scale.

4.6 Generation Settings for Gates 1–7

During validation:

- **Temperature:** 0.7
- **Max tokens:** 96
- **Top-k for entropy:** 20
- **Sampling:** Greedy for entropy measurement, but model output uses the above decode settings
- **Seed:** Fixed per prompt for reproducibility
- **Logging:**
  - entropy per token
  - ΔH per token
  - actuator decisions
  - timing
  - route-to-human events
  - JSON logs for each prompt/step

This ensures deterministic evaluation of ψ*-band behavior while leaving output generation in a realistic decoding regime.

4.7 Metrics Used Across the Gates

Token Entropy (H)

Entropy of renormalized top-k logits.

Entropy Change (ΔH)

$$\Delta H_t = H_t - H_{t-1}.$$

Used in escalation-sensitive gates.

Entropy Variance (σ^2)

Compared between:

- **Track A:** Uncalibrated baseline
- **Track B:** Aligned ψ*-band policy execution

Used in Gate 2.

Accuracy

Binary correctness label of the final answer.

Escalation Rate

Fraction of generations where:

$$\Phi(\text{state}) = \text{extremeand}\Psi(\text{act}) = \text{stop}.$$

Overhead (Latency)

Difference in step-level runtime between Track A and B.

4.8 Evaluation Protocol Summary

The sequence of operations is:

1. Construct ψ* from reference model
2. Run UNSUP_H_ALIGN on each target model
3. Select aligned models
4. Evaluate them on 60-prompt set under two tracks
5. Compute gate metrics (1–7)
6. Validate pass/fail
7. Document failures and debugging steps

This pipeline forms the reproducible basis for the results in Section 5.

5 Validation: The 7-Gate Framework

The HDT² framework proposes that once a model is aligned to the ψ* entropy geometry, calibrated entropy bands will exhibit reliable behavioral signatures. To test this claim, we evaluate aligned models under a structured, seven-gate checklist analogous to pre-flight validation. Each gate targets a distinct property of a well-functioning uncertainty diagnostic: control, variance, accuracy, escalation, safety, computational cost, and reproducibility.

Only **Mistral-Nemo-Instruct-2407** passed UNSUP_H_ALIGN (Gate 0) and thus proceeded through the 7-gate validation pipeline.

Models failing alignment (Qwen2.5-7B, Mistral-7B) are documented as **unalignable under current ψ*** and excluded from deeper behavioral testing.

5.1 Gate Design Philosophy

Each gate tests a specific hypothesis:

1. **Gate 1:** The system actually implements control signals correctly.
2. **Gate 2:** Calibration reduces entropy variance.
3. **Gate 3:** Stable bands correlate with correctness.
4. **Gate 4:** Extreme-band entropy reliably triggers escalation.
5. **Gate 5:** Added overhead is acceptably small.
6. **Gate 6:** Safety is preserved—no degradation on safe prompts.
7. **Gate 7:** Behavior is reproducible under fixed seeds.

Together these gates validate the *functionality* of the ψ*-aligned policy.

5.2 Gate 1 — Functional Control

Claim

Actuation must correctly follow Φ's decision state.

If ψ* flags an entropy value as *extreme*, the system must:

- set `route_to_human = true`
- set `max_new_tokens = 0`
- terminate the generation

Test

For all steps in all prompts:

$$\text{state}(H_t) = \text{extreme} \;\Rightarrow\; \Psi(\text{act}) = \text{hard-stop}.$$

Result: PASS

After early debugging (Section 5.8), all extreme-band states correctly triggered immediate termination with 100% coverage.

5.3 Gate 2 — Entropy Variance Reduction

Claim

If ψ*-aligned bands are meaningful, Track B (calibrated) should exhibit *less entropy variance* than Track A (baseline).

$$\sigma_B^2(H) < \sigma_A^2(H)$$

Test

Compute per-prompt entropy variance for both tracks across all 60 prompts.

Result: PASS

Variance reduction was observed in 58/60 prompts; the two non-reducing prompts remain documented in artifacts.

This indicates that ψ* exerts a stabilizing influence on entropy trajectories without modifying logits or generation.

5.4 Gate 3 — Accuracy Correlation

Claim

Responses produced while generation remains in the **stable** band should exhibit higher correctness rates than responses dominated by risky/high-risk entropy.

Test

Compute accuracy of outputs categorized by dominant band:

- Stable-dominant
- Risky-dominant
- Extreme (almost always escalated)

Result: PASS

Stable bands showed higher accuracy than risky bands.

This does *not* claim causal relationship—only predictive correlation consistent with diagnostic usefulness.

5.5 Gate 4 — Escalation Behavior

Claim

Extreme-band entropy should reliably predict instability and trigger escalation.

Test

For every step:

$$H_t > Q_{90}^{\psi\backslash *} \Rightarrow \Psi(\text{act}) = \text{stop}$$

Result: PASS

After fixing a routing bug (Section 5.8), all extreme-band states correctly resulted in escalation.

This demonstrates that ψ* can act as a thresholded routing mechanism.

5.6 Gate 5 — Computational Overhead

Claim

The HDT^2 controller imposes minimal runtime penalty.

Test

Measure latency difference between Track A and Track B.

Result: PASS

Overhead remained below 10% for all prompts.

This is expected because:

- Entropy computation uses a single top-k slice
- No logits are altered

- Actuation uses fixed-cost conditionals

5.7 Gate 6 — Harm / Safety Preservation

Claim

Applying ψ* should not degrade performance on safe prompts or reduce the model's ability to answer correctly when uncertainty is low.

Test

Compare accuracy on stable-band prompts in:

- baseline (Track A)
- calibrated (Track B)

Result: PASS

No measurable degradation; accuracy remained effectively identical.
This confirms that ψ* does not reduce model capability on low-uncertainty inputs.

5.8 Gate 7 — Reproducibility

Claim

Given fixed seeds and decoding settings, the aligned ψ*-based pattern should reproduce reliably.

Test

Run Track B twice under fixed seeds and compare:

$$\Delta H_t^{(1)} - \Delta H_t^{(2)}$$

Require:

- variance < 0.01
- identical actuation traces
- identical escalation positions

Result: PASS

Both runs matched actuation logs exactly, and δH variance stayed below the threshold.

5.9 Debugging Journey (Transparency Notes)

A central purpose of the gate framework is to catch early implementation errors.
In this pilot, several real issues surfaced:

1. **Prompt ID mismatch (p# vs f#)**
   - Caused incorrect attribution of entropy metrics
   - Fixed by unifying ID schema
2. **Escalation logic error**
   - Extreme-band states weren't always triggering stop
   - Resolved by correcting control flow around Φ→Ψ transitions
3. **Syntax error in checker**
   - Variance computation silently failed
   - Debugged post-hoc via gate-structured review
4. **Incorrect ΔH threshold in early prototype**
   - Misinterpreted extreme-band threshold
   - Corrected after comparing artifacts to ψ* JSON

These failures were recorded, fixed, and re-tested.
The 7-gate process thus serves as a *functional correctness harness* for calibrated uncertainty signals.

5.10 Summary of Gate Outcomes

Gate	Description	Status
0	UNSUP_H_ALIGN (affine fit)	PASS for Mistral-Nemo; FAIL for two 7B models
1	Functional control	PASS
2	Variance reduction	PASS
3	Accuracy correlation	PASS
4	Escalation reliability	PASS
5	Overhead	PASS
6	Harm prevention	PASS
7	Reproducibility	PASS

Mistral-Nemo is the only model that passed all gates and is therefore considered ψ*-compatible under the present method.

6 Results & Analysis

This section summarizes the primary empirical findings of the HDT² pilot study.
We analyze (1) cross-model entropy alignment outcomes, (2) performance on the 7-gate framework, and (3) what these results suggest about the generalizability and limitations of ψ*-based reasoning diagnostics.

6.1 Cross-Model Calibration Outcomes

UNSUP_H_ALIGN is the prerequisite for all subsequent analysis.
Only models satisfying the quantile-alignment thresholds (max error ≤5%, median ≤3%) are considered ψ*-compatible.

Table 1 — Alignment Results

Model	Params	Max Error	Median Error	Status
Qwen 2.5-14B (ref)	14B	0.0%	0.0%	Reference
Mistral-Nemo-2407	12B	**3.8%**	**1.9%**	**PASS**
Qwen 2.5-7B	7B	8.4%	4.8%	FAIL
Mistral-7B-v0.3	7B	11.2%	6.9%	FAIL

Interpretation

These outcomes reveal two important observations:

1. **ψ* is not universally alignable**
   The two 7B models exceed the predefined error thresholds, suggesting that ψ* captures a particular entropy geometry found in medium-size and larger transformer models, but not necessarily in smaller or differently trained ones.
2. **Alignment appears architecture-sensitive**
   Both failures occurred in models with lower parameter counts and different training distributions.
   This suggests an emerging hypothesis: *shared scaling behavior and training regime may be prerequisites for ψ*-compatibility.*

Failing alignment is not considered a defect; it defines **where the current method no longer applies**.

6.2 Behavioral Validation (Gates 1–7)

Only Mistral-Nemo-Instruct progressed to behavioral evaluation.

Figure 1 — Gate Summary

(Already presented in Section 5.)
All 7 gates passed after debugging corrections.
Here we analyze the behavioral signatures that emerged.

6.3 Entropy Profile Stabilization

Track B (ψ*-aligned policy) exhibited noticeably smoother entropy trajectories.

Observation

Across 60 prompts:

- Track A (baseline) showed entropy spikes in ~40% of samples

- Track B reduced both frequency and amplitude of spikes
- Prompt-to-prompt variance decreased for 58/60 items

Interpretation

This stabilization indicates that **ψ*-aligned uncertainty classification is coherent with the model's internal reasoning dynamics**, even though HDT² does not modify logits or perform interventions.

This supports the view that entropy contains latent structure predictive of instability.

6.4 Accuracy Patterns Across Bands

Accuracy rates by dominant-band classification:

Band	Accuracy
Stable	**Higher**, statistically dominant
Risky	Lower
Extreme	Usually escalated; accuracy not applicable

Interpretation

This result is critical:

It demonstrates that **entropy bands do correlate with output reliability**, even though they are derived from a different model.

However, correlation does **not** imply causal explanation.

The calibrated bands serve as a *predictive diagnostic*, not a justification for why errors occur.

6.5 Escalation Reliability

Extreme-band states consistently triggered immediate escalation (after corrections).

This confirms that:

- ψ* extreme thresholds can serve as **routing boundaries**
- Escalation is not triggered by noise
- The controller logic is internally consistent

Implication

ψ* can function as a **stop condition** for hallucination-prone states, without requiring task labels or semantic knowledge.

6.6 Safety and Non-Degradation

Stable-band prompts showed no performance degradation when ψ* was applied.

Interpretation

This addresses a typical reviewer concern:

- Safety layers sometimes reduce capability
- ψ* did not
- The read-only design maintains model expressiveness

This supports ψ* as minimally invasive.

6.7 Failure Cases (Alignment Level)

Although these models never reached Gates 1–7, their failures are informative.

Qwen 2.5-7B

- Lower entropy values across the board
- Compressed distribution
- Affine map insufficient to recover ψ* shape

Mistral-7B

- High skew and heavy tail
- Extreme quantile divergence
- Suggests different uncertainty geometry

Interpretation

These models highlight a **boundary condition**:

ψ* may require scale, training diversity, or architectural similarity to the reference model for meaningful alignment.

This is a direction for future theoretical work (Section 8).

6.8 What We Learned

Successes

- Cross-model entropy alignment is **feasible** (at least for one model).
- Aligned bands **predict accuracy**.
- Calibration **stabilizes entropy** without modifying generation.
- The 7-gate framework reliably catches bugs and validates behavior.
- ψ* can serve as a **model-agnostic uncertainty geometry** when alignment is achieved.

Limitations

- ψ* is **not universal**: smaller models diverge.
- Behavioral claims depend on the **alignment condition** (Gate 0).
- Sample size (N=60) remains pilot-scale.
- Results describe **structured uncertainty**, not hallucination elimination.

Most Important Insight

Failure cases are as informative as success:

- They demonstrate the boundaries of ψ* generalization
- They show that "entropy geometry" is not shared across all LLMs
- They validate the importance of UNSUP_H_ALIGN as Gate 0

This distinction protects the scientific integrity of the framework:

ψ* is applicable **when alignment holds**, not universally.

7 Discussion

The goal of this pilot study was not to prove that entropy-band calibration solves hallucination, but to evaluate whether a model-agnostic uncertainty geometry—ψ*—can be established across different LLMs, and whether this geometry exhibits functional behavioral signatures. The findings reveal both the promise and the limits of this approach.

7.1 Why Some Models Did Not Align

A key outcome is that ψ* was **not** universally alignable.
Two 7B models—Qwen2.5-7B and Mistral-7B—failed Gate 0 (UNSUP_H_ALIGN).

Possible Factors

1. **Scale Effects**
   Smaller models often exhibit sharper entropy spikes and narrower distributions.
   Their uncertainty geometry is structurally distinct from that of a 12B or 14B model.
2. **Training Divergence**
   Differences in pretraining corpora, RLHF strategy, and vocabulary distribution alter entropy profiles in ways that cannot be corrected with a simple affine map.
3. **Entropy Compression**
   Some models produce systematically lower entropy due to aggressive instruction tuning or shortcut heuristics.

Interpretation

ψ* is not a universal geometry—it is a *family-dependent* geometry.
This is not a flaw.
It is a crucial empirical constraint defining where the method applies and where further theory is required.

7.2 ψ* as a Portable Uncertainty Geometry

For the one model that *did* align (Mistral-Nemo), ψ* provided:

- stable entropy trajectories
- predictable accuracy gradients
- reliable escalation when entropy exceeded thresholds
- minimal overhead
- reproducible behavior

Why This Matters

Most hallucination detection methods require:

- supervised data
- ensembles
- internal access to activations
- model-specific calibration
- parameter modification

ψ* requires none of those.
It provides a **portable, read-only diagnostic** for detecting reasoning instability, conditioned on successful alignment.

This conditionality is essential:
ψ* is not claimed to be universal—only that when alignment succeeds, the bands convey useful behavioral structure.

7.3 Comparison to Prior Work

HDT² differs from previous hallucination-detection methods in four ways:

1. **Unsupervised Calibration**
   No correctness labels or domain tuning are used.
2. **Cross-Model Definition**
   ψ* defines uncertainty regions not for one model, but as a shared reference geometry.
3. **Control-Theoretic Framing**
   The Ω–Δ–Φ–Ψ structure positions entropy as a *signal* in a diagnostic loop, rather than as a raw statistic.
4. **Structured Validation**
   The 7-gate protocol evaluates function, stability, accuracy correlation, safety, overhead, and reproducibility.

Where HDT² Is Similar

- Like perplexity-based methods, it uses token entropy.
- Like calibration literature, it aligns distributions.
- Like safety layers, it supports escalation.

Where HDT² Is Distinct

- It attempts **cross-model normalization** of uncertainty.
- It uses **endo-structural tests** rather than classification labels.
- It documents **failure cases** as part of the scientific result.

7.4 Implications for LLM Reliability

If ψ* alignment succeeds:

- entropy bands behave predictably
- extreme entropy consistently signals instability
- stable bands correlate with correctness
- entropy becomes a meaningful decision boundary

If ψ* alignment fails:

- the model's uncertainty geometry does not match ψ*
- applying the controller would be arbitrary
- downstream gates would no longer test the HDT² hypothesis

This creates a clear methodological boundary:
HDT² can only be applied where Gate 0 holds.
This protects the framework from overclaiming and guides future research toward understanding the structural factors that determine alignability.

7.5 Limitations of This Pilot

This work is intentionally limited:

- **Sample size is small** (N=60 prompts).
- **ψ* derived from a single reference model.**
- **Affine alignment may be insufficient** for models with non-linear entropy distortions.
- **Accuracy correlation does not imply causation.**
- **No comparison against perplexity or ensemble baselines** yet.

The contribution is therefore *directional*, not definitive.

7.6 Toward Broader Systems and Applications

If ψ* alignment can be extended more broadly, applications include:

- **Routing systems** that escalate when reasoning becomes unstable
- **Real-time monitoring** of long-chain reasoning tasks
- **Model health dashboards** using entropy drift as an early indicator
- **Safety layers** for open-ended chat systems
- **Cross-model consistency protocols** for multi-agent environments

These applications require **larger-scale evaluation**, but the present pilot provides the conceptual foundation.

7.7 Personal Reflection (Author Voice)

This work originated from a long-standing question in my own life:
how do you know when your reasoning is becoming unstable?
As someone with dyslexia, I have learned to identify small, subtle signals—shifts in clarity, emerging friction, rising cognitive turbulence—that indicate it is time to shift modality or slow down. These lived markers of instability inspired the structure of HDT²:

- Ω: notice the signal
- Δ: understand its context
- Φ: classify the state
- Ψ: respond accordingly

The fact that a similar structure appears to have measurable utility in LLMs is both encouraging and intellectually intriguing.

However, this parallel is presented as motivation, not evidence: the present work confines itself to empirical behavior in machine systems.

7.8 Scientific Positioning

HDT² should be viewed as:

- **a falsifiable method**, not a claim about cognitive universals
- **a diagnostic protocol**, not a path to eliminating hallucination
- **a structured experiment**, not a theory of intelligence
- **a pilot result**, not a sweeping generalization

What this work demonstrates is that **entropy carries structured information** about reasoning stability, and that this information can sometimes be normalized across models.

The failure cases are equally important: they reveal structural differences in model uncertainty behavior and define the scope of ψ*.

8 Limitations & Future Work

This pilot study demonstrates that entropy-band calibration (ψ*) and unsupervised alignment (UNSUP_H_ALIGN) can yield meaningful reasoning diagnostics for at least one model. However, the method in its current form has clear limitations. This section enumerates those limitations explicitly and outlines directions for future research.

8.1 Current Limitations

(1) Pilot Scale (N = 60 prompts)

The sample size for behavioral validation is deliberately small.
While sufficient to test the mechanics of the gate framework, it is not large enough to establish statistical stability across domains or tasks.

Consequence:
Claims about generality must be cautious; the present work demonstrates feasibility, not coverage.

(2) Single Reference Model (ψ* from Qwen2.5-14B-Instruct)

All entropy bands were derived from a *single* reference model.
Different reference choices may produce different ψ* geometries.

Consequence:
We cannot yet say whether ψ* is stable across reference models, or whether each reference produces its own local geometry.

(3) Simple Affine Alignment (H' = aH + b)

UNSUP_H_ALIGN currently uses an affine transformation aligned to three quantiles (Q25, Q50, Q75).
This assumes that entropy geometry can be linearly transformed across models.
This assumption held for one model (Mistral-Nemo) but failed for two others.

Consequence:
Affine maps may be insufficient to capture non-linear distortions in uncertainty distributions.

(4) No Comparison Against Baselines Yet

This study did not include:

- perplexity thresholds
- sampling-variance baselines
- log-prob calibration methods
- supervised hallucination detectors
- classifier-based consistency checks

Consequence:
We cannot yet quantify how ψ* compares to existing uncertainty measures.

(5) Unknown Mechanism: Why Does Entropy Predict Accuracy?

Although entropy-band classification correlates with accuracy, the underlying mechanism remains unclear.
Possibilities include:

- shared training dynamics across mid-sized transformers
- common failure patterns in generative reasoning
- structural relationships between logit dispersion and correctness

Consequence:
Without a mechanistic model, ψ*-based diagnostics remain empirical rather than explanatory.

(6) Limited Model Diversity

Only four models were tested, all from two families (Qwen, Mistral).
No evaluations were performed on:

- Llama-family models
- Qwen2.5-32B / larger models
- Gemma-2
- Architecturally divergent models (Mixture-of-Experts, vision-language models, etc.)

Consequence:
We do not yet understand which architectures share entropy geometry with ψ*.

(7) No Long-Sequence or Multi-Step Reasoning Evaluation

All generations were capped at 96 tokens.
Long-chain reasoning tasks may produce qualitatively different entropy phenomena.

Consequence:
ψ* may behave differently in chain-of-thought or multi-hop deliberation settings.

(8) Not a Hallucination Detector (Yet)

ψ* identifies instability conditions but does not classify hallucinations directly.

Consequence:
HDT² is a *diagnostic substrate*, not a hallucination classifier or safety system.

8.2 Immediate Future Work

These are the highest-leverage next steps for strengthening the empirical foundation.

(1) Expand Prompt Set to ≥ 500 Items

A larger, task-diverse dataset will allow:

- statistical confidence intervals
- band-strength analysis
- sensitivity curves
- ROC-style evaluation of escalation thresholds

(2) Cross-Reference Against Perplexity Baselines

Compare ψ* escalation against:

- simple perplexity thresholds
- entropy without alignment
- sampling variance (temperature sweeps)

This will quantify added value beyond raw entropy.

(3) Test Larger Model Grid

Run UNSUP_H_ALIGN on:

- Llama-3-8B, 70B
- Qwen2.5-32B
- Mistral-Large
- Gemma-2 and Gemma-2-27B
- Mixtral MoE models

This will map the "ψ*-compatible" region of model space.

8.3 Mid-Term Work

(1) Explore Non-Linear Alignment Methods

Potential approaches:

- isotonic regression
- spline-based mapping
- mixture-of-Gaussians fitting
- quantile transport (Wasserstein alignment)

Goal:

Determine whether more general transformations restore ψ* alignment for previously incompatible models.

(2) Multi-Model ψ* Compositing

Instead of a single reference:

- combine entropy distributions across models
- compute a consensus ψ*
- test whether composite bands generalize more widely

(3) Domain-Specific Calibration

Evaluate ψ* behavior in:

- code generation
- medical question answering
- scientific reasoning
- legal/compliance tasks
- chain-of-thought-heavy domains

(4) Mechanistic Investigation

Key research question:

Why does entropy-band structure correlate with correctness?

Possible approaches:

- probing internal activations
- logit-lens analysis
- causal scrubbing
- local geometry analysis of softmax surfaces
- differential comparison across architectures

Understanding the mechanism may reveal universal uncertainty signatures.

8.4 Long-Term Work

(1) Real-World Deployment Studies

Integrate ψ* into:

- a routing layer for chat assistants
- agentic planners
- code interpreters
- multi-agent ensembles
- trust-calibrated systems

Measure real-world reliability improvements.

(2) Generalize ψ* Into a Multi-Layer Diagnostic System

Possible extensions:

- multi-band entropy curves
- entropy drift detection
- internal-state cross-sectional monitoring
- ψ–Ω feedback loops for adaptive prompting
- integration into system-level safety controllers

(3) Cognitive-Theoretic Bridging (Long-Term, Non-Claim-Based)

Exploratory direction only (not part of the current paper's claims):

- compare entropy-band dynamics with human uncertainty tracking
- test whether ψ* can predict when chain-of-thought becomes unstable
- explore broader connections between structured uncertainty and reasoning architecture

This direction remains conceptual and is not required for empirical validation.

9 Conclusion

This work introduced HDT², a pilot framework for entropy-based reasoning diagnostics in large language models. The core idea is that token-level Shannon entropy, when normalized through an unsupervised alignment procedure (UNSUP_H_ALIGN), can define a portable uncertainty geometry—ψ*—shared across at least some model families. When a target model successfully aligns to ψ*, predictable behavioral signatures emerge: stabilized entropy trajectories, reliable escalation on high-uncertainty states, correlation between stable bands and correctness, and consistent actuation under a structured control framework (Ω–Δ–Φ–Ψ).

These results were demonstrated on a single aligned model (Mistral-Nemo-Instruct-2407). Two smaller models failed to align, defining essential boundaries of the method's applicability. The 7-gate validation framework proved critical: it not only verified functional behavior but also surfaced genuine implementation errors during development, underscoring its utility as a correctness harness for calibrated uncertainty systems.

Importantly, HDT² does not claim to solve hallucination or provide universal uncertainty calibration across all LLMs. Rather, it shows—at pilot scale—that entropy distributions contain structured information about reasoning stability, and that under specific conditions this structure can be normalized across models. Where alignment holds, ψ* becomes a practical, read-only diagnostic capable of triggering interpretable routing decisions without modifying model parameters or relying on supervised labels.

The broader implication is that LLM reasoning stability may be partially governed by measurable uncertainty geometry. Understanding the limits of this geometry—and the conditions under which it can be shared—constitutes a promising direction for future work. The transparency of this study, including documented failures and complete reproducibility artifacts, provides a foundation for that exploration.

HDT² is therefore best understood as an early step: a constrained but falsifiable contribution toward portable uncertainty diagnostics for machine reasoning. Further empirical expansion, theoretical investigation, and systematic comparison with established baselines will be needed to determine its full scope and applicability. Yet even at this early stage, the results indicate that structured entropy-band calibration offers a tractable and scientifically grounded path forward in the pursuit of more reliable AI systems.

10 Related Work and Positioning

Research on LLM reliability has converged on three broad strategies for obtaining usable uncertainty signals: (i) **intrinsic, entropy/logit-based diagnostics**, (ii) **extrinsic supervised correctness predictors**, and (iii) **wrapper methods with formal coverage guarantees**. HDT² belongs to the first group, but is structurally distinct in how it constructs and tests a portable uncertainty geometry.

Entropy and Logit–Based Hallucination Diagnostics

A large body of work uses token-level Shannon entropy, perplexity, or related logit statistics as proxies for confidence and hallucination risk. Recent methods such as Logits-induced Token Uncertainty (LogU / LogTokU) derive token-specific uncertainty directly from logits via evidence modeling, enabling real-time hallucination detection without multiple sampling rounds. arXiv+1 Other approaches analyze entropy production rate or band tokens into low/high-uncertainty regimes to drive supervised hallucination classifiers or downstream control heuristics. Semantic Scholar+1

These methods are **model-internal and usually model-specific**: thresholds or detectors are calibrated per model, with no explicit attempt to construct a shared, cross-model entropy geometry. In contrast, HDT² uses Shannon entropy as the base signal but introduces a **portable band structure ψ*** derived from a reference model and an explicit unsupervised alignment procedure (UNSUP_H_ALIGN) that either brings a target model into that geometry or declares it incompatible (Gate 0).

Generalized Correctness Models (GCM)

Generalized Correctness Models take a complementary, **extrinsic and supervised** approach. A separate "correctness model" is trained on historical predictions and correctness labels from many LLMs, learning patterns that generalize across model families and sizes; these models often outperform a given LLM's own self-confidence when predicting answer correctness. arXiv+2arXiv+2 GCMs are thus **model-agnostic in deployment**, but their portability arises from supervised learning over text and metadata, not from shared internal uncertainty geometry. HDT² differs in two ways: (i) it uses **no correctness labels** to construct ψ* or perform UNSUP_H_ALIGN, and (ii) its portability is explicitly **conditional** on satisfying a structural compatibility test (Gate 0). GCMs can be viewed as external correctness oracles; HDT² is an intrinsic, label-free *filter* that operates directly on the model's entropy stream.

Conformal Uncertainty and Coverage Guarantees

A third line of work wraps LLMs in **conformal prediction** to obtain distribution-free coverage guarantees. ConU, for example, converts heuristic uncertainty scores into calibrated prediction sets for open-ended generation, guaranteeing that correct answers lie in the set with at least a user-specified probability. arXiv+2ACL Anthology+2 Subsequent work extends these ideas to selective or domain-aware conformal uncertainty. ACL Anthology Related frameworks integrate conformal scores into multi-step reasoning pipelines, using global error-rate controllers to keep compounded failure rates within bounds. OpenReview+1

These methods provide **strong theoretical guarantees** but require labeled calibration sets and typically do not interpret the internal geometry of entropy itself. HDT², by contrast, offers **no formal coverage guarantees at this stage**; its contribution is empirical and structural: it shows that, under specific conditions, token-entropy distributions can be normalized into a shared banded geometry ψ* that carries predictive information about reasoning stability, and that this geometry can fail cleanly in incompatible models (Gate 0).

Entropy as a Control Signal

Finally, several recent methods use entropy as a **control signal** for adaptive computation rather than as a purely diagnostic metric. “Think Just Enough” uses sequence-level entropy as a confidence signal to trigger early stopping in reasoning, saving 25–50% computation while preserving accuracy. OpenReview+2arXiv+2 Confidence-aware reasoning controllers similarly modulate reasoning depth based on token-level confidence. ACL Anthology+1

HDT² is aligned with this control-theoretic trend but targets **stability and escalation** rather than efficiency. The Ω–Δ–Φ–Ψ loop turns entropy bands into explicit control actions (continue vs. route-to-human), and the seven-gate protocol validates that this control behaves as intended on models that pass alignment. Unlike early-stopping work, HDT² also treats **alignment failure itself as a first-class outcome**, documenting where its control law should *not* be applied.

Summary of Distinctions

In summary, HDT² is most closely related to entropy-based hallucination detectors and entropy-driven control systems, but differs in three key respects:

1. It treats token-entropy as a **shared geometric object ψ*** rather than a per-model heuristic.
2. It uses **unsupervised quantile alignment** to test whether a target model can be embedded in that geometry (Gate 0), and reports failures as structural constraints.
3. It integrates ψ* into a minimal, external **control loop (Ω–Δ–Φ–Ψ) plus a seven-gate functional checklist**, emphasizing falsifiability and operational behavior rather than theoretical guarantees.

These distinctions position HDT² as a **conditionally portable, label-free diagnostic substrate** that can coexist with supervised correctness models and conformal wrappers, and that can be combined with them in future work (e.g., ψ* as a filter, GCM as a correctness oracle, CP as a guarantee layer).

HDT^2 — License & Use Policy

This repository contains the full implementation of the HDT^2 framework, including:

- ψ^* entropy-band reference distributions
- UNSUP_H_ALIGN calibration procedure
- Ω–Δ–Φ–Ψ operational scaffold
- Seven-gate validation suite
- All experimental data, artifacts, and analysis scripts

To support transparent scientific evaluation, the entire framework is made publicly available under a research-friendly license.

License

You are free to:

- Share — copy and redistribute the material in any medium or format
- Adapt — remix, transform, and build upon the material
- Use for academic, educational, or research purposes without restriction

Under the following terms:

- Attribution — You must give appropriate credit.
- NonCommercial — *Commercial use is prohibited without explicit written permission.*
- ShareAlike — Derivatives must be released under the same license.

Commercial Use

Commercial use is not permitted under CC BY-NC-SA 4.0.
This includes (but is not limited to):

- Integrating HDT^2 into commercial LLMs
- Using ψ^*, ΔH thresholds, or the seven-gate protocol inside paid products
- Packaging the method in developer tools or hosted APIs
- Offering consulting, auditing, or certification services built on HDT^2
- Selling derivative works containing the method

Commercial licensing is available by request.

Contact: brucetisler@quantuminquiry.com

Approval is granted selectively for projects that uphold the ethical standards and safety requirements defined in the HDT² Ethical Use Statement.

Academic Use

- Academic, nonprofit, and government researchers may use, modify, replicate, or extend HDT² without obtaining additional permission.
- All published derivatives must cite the canonical HDT² paper.

Citation

If you use HDT² in research, please cite:

```
cff-version: 1.2.0
message: "If you use HDT² or associated code in your research, please cite as follows."
title: "HDT²: A Pilot Framework for Entropy-Band Calibration of LLM Reasoning Stability"
authors:
  - family-names: "Tisler"
    given-names: "Bruce"
date-released: "2025-11-15"
version: "1.0.0"
abstract: >
  HDT² is a research framework for measuring and calibrating entropy-band
  uncertainty in large language models through unsupervised quantile alignment,
  portable entropy bands (ψ*), and a seven-gate validation protocol.
repository-code: https://github.com/btisler-DS/hdt2-entropy-band-calibration
doi: "10.5281/zenodo.17621326"
license: "CC-BY-NC-SA-4.0"
```