# Iowa Initiative for Artificial Intelligence
# Final Report

| | |
|---|---|
| Project title: | Experimental Studies Demonstrating the Application of Asymptotic Equipartition Properties to Generative AI Models |
| Principal Investigator: | Mudumbai Raghu, Bell Tyler, Dasgupta Soura, ECE |
| Prepared by (IIAI): | Mudireddy Avinash and Ivan Johnson-Eversoll |
| Other investigators: | |
| Date: | March 18, 2026 |

| | |
|---|---|
| Were specific aims fulfilled: | Y |
| Readiness for extramural proposal? | Yes |
| If yes … Planned submission date | Already submitted<br>1. DOE EXPRESS: Submitted in March 2025, Duration: 07/2025 - 06/2027 (2 years)<br>2. NSF MFAI: Submitted in October 2024, Duration: 04/2025-03/2028 (3 years) |
| Funding agency | 1. DOE EXPRESS: AI Stereotypes: Fundamental Constraints on the Outputs of Generative Models DE-FOA-0003545<br>2. NSF MFAI: Asymptotic Equipartition Properties and Typical Sets for Generative AI Models, and their Consequences |
| Grant mechanism | Core NSF |
| If no … Why not? What went wrong? | |

## Executive Summary

This project investigated how the Asymptotic Equipartition Property (AEP) manifests in modern generative language models across two continuous phases of work. The first phase established the theoretical and experimental foundation for studying entropy, log-perplexity, and typical-set behavior in self-generated text. The second phase converted that foundation into a stronger cross-model framework based on source-masked scoring and empirical cross-entropy, allowing the project to move from qualitative divergence observations to quantitative cross-model comparison.

- Empirical Validation of AEP: Delivered the first large-scale empirical confirmation that generative language models behave in accordance with AEP principles—specifically, log-perplexity consistently converges to entropy in self-generated text.
- Modular Testing Framework: Designed and implemented a reusable, test-driven experimentation framework, enabling scalable and replicable analysis across various models and parameter configurations.
- Codebase Modernization: Refactored a legacy codebase and verified functional parity through regression testing, improving maintainability and setting the foundation for future extensions.
- Scalable Experimentation Infrastructure: Developed pipelines to support parallel cross-model analyses with statistical rigor and efficiency, allowing for more extensive experimentation.

- Advanced Hugging Face API Integration: Fully transitioned to modern Hugging Face pipelines, significantly reducing computational overhead and enabling finer control over logits and sampling parameters.
- Cross-Model Typical Set Analysis: Showed that typical sets diverge across different models, suggesting new possibilities for model classification and detection of AI-generated content.
  Cross-AEP Quantitative Closure: Added source-masked cross-model scoring and aggregated 444 comparison files, showing that cross-model cross-entropy h_(A,B) is heterogeneous, structured, and useful for model-attribution style analysis.

# Research Report

## Aims (as specified in the project proposal)

The primary objective of this project was to investigate and demonstrate the application of the Asymptotic Equipartition Property (AEP) to generative AI models through a combined theoretical and empirical program. Across the semester and final closure phase, the aims matured from validating self-model convergence to constructing a rigorous cross-model formulation that could compare how one model scores text generated by another.

1. Prove the theoretical AEP – Develop and support a mathematical framework demonstrating that the log-perplexity of long texts generated by a language model asymptotically converges to the average entropy of its token distribution.
2. Demonstrate divergence of typical sets between models – Show that different language models produce outputs belonging to distinct *typical sets*, which represent vanishingly small subsets of all grammatically correct outputs.
3. Explore practical applications of model detection and classification – Investigate whether the theoretical insights could be extended to real-world tasks, such as identifying AI-generated text or determining if a given text was included in a specific model's training data.

As the project progressed, these original aims were sharpened rather than replaced. The final phase focused on methodological closure: formalizing the cross-AEP target, aligning running log-perplexity l_B with empirical cross-entropy h_(A,B), and building a source-masked comparison workflow that could support scalable, publication-ready cross-model analysis.

## Data

Unlike conventional machine learning projects that rely on curated labeled datasets, this research centered on generating and analyzing text directly from large language models (LLMs). The data story therefore has two connected layers: self-generated long-form sequences used to validate AEP-style convergence within a model, and cross-model scoring datasets used to measure how that same text behaves when evaluated by a different scorer model.

By treating the models as black-box stochastic systems and focusing on their output behavior under controlled conditions, the project sought evidence for two linked claims: (1) running log-perplexity approaches empirical entropy for long self-generated sequences, and (2) cross-model scoring reveals model-specific typicality signatures that are not eliminated by prompt or seed variation.

## Data Sources and Generation

Three integrated data sources were utilized to support the experimental objectives:

- Self-Generated Texts: Long-form sequences were generated using fixed prompts across several large language models, including LLaMA 3.1 and Gemma-family variants. These sequences formed the primary evidence base for within-model entropy and log-perplexity convergence.
- External Reference Texts: Curated literary excerpts and other held-out passages were scored to explore over-typicality, memorization-style behavior, and how known texts differ from self-generated baselines.

Cross-Model Comparison Corpus: In the closure phase, previously generated sequences were rescored under alternate models using a source-masked workflow. This produced 444 usable comparison files, together with per-sequence summaries, running averages, and step-level cross-entropy traces for cross-model analysis.

## Data Organization

Significant effort was dedicated to ensuring data consistency and comparability across both project phases. This organizational groundwork was essential not only for reproducibility but also for merging self-model and cross-model analyses into a single continuous experimental narrative.

- Controlled Vocabulary Development: Established a standardized vocabulary for key metrics—such as log-perplexity, entropy, and deviation measures—to ensure conceptual clarity and reduce ambiguity in analysis.
- Structured Data Formats: Designed internal data schemas using nested JSON and CSV formats to store per-token outputs alongside their corresponding probability distributions and metadata.
- Systematic Naming Conventions: Implemented consistent naming based on model type, prompt ID, random seed, and generation parameters. This facilitated reproducibility and supported scalable aggregation for statistical evaluation.
  Artifact-Level Report Packaging: Comparison runs were stored with explicit artifact paths for plots, summary statistics, and sequence-level diagnostics, enabling figure traceability and later manuscript-style assembly.

## Token-Level Metadata Collection

For each generated sequence, comprehensive metadata was captured at the token level, providing the granularity required to evaluate convergence behaviors and statistical properties relevant to the Asymptotic Equipartition Property (AEP). Specifically, the following data was recorded at each generation step:

- Token position and index
- True next-token rank (within the model's predicted distribution)
- Full probability distribution over the model's vocabulary
- Sampled and greedy next-token outputs

This rich metadata enabled detailed analysis of log-perplexity, entropy, deviation trends, and cross-model scoring behavior. In the final phase, the same token-level framework was extended to retain scorer identity, source-model identity, running cross-entropy estimates, and sequence-level summary statistics needed for aggregate comparison.

## AI/ML Approach

This project diverged from traditional AI/ML workflows that involve fitting predictive models to labeled examples. Instead, it treated large language models (LLMs) as black-box stochastic systems and used information-theoretic analysis to study their generated outputs and cross-model scoring behavior.

Key components of the approach included:

1. Theoretical Framework Development: A generalized AEP framework was formulated for practical language models without assuming token independence, and later extended to a cross-AEP setting where text generated by model A is scored by model B.
2. Probabilistic Analysis: Core concepts from information theory—especially the Law of Large Numbers, entropy concentration, and empirical cross-entropy—were used to study how token sequences align with self-model and cross-model expectations over long horizons.
3. Model Sampling and Interrogation: Custom generation and scoring pipelines were implemented to extract full token-level distributions, sampled-token probabilities, and running summaries. This enabled fine-grained analysis of both self-generated typicality and cross-model divergence.

Rather than constructing new models, the project focused on uncovering statistical constraints that govern the behavior of existing generative models. The central objects of study were the model-specific typical set, entropy-tracking dynamics, and the way those quantities transform when text is transferred from one model family to another for scoring.

## Experimental Methods and Validation Approach

The experimental design was organized as a continuity between an initial validation phase and a final closure phase. The initial phase verified that the required token-level measurements were reliable and that long self-generated sequences behaved in line with AEP-style predictions. The final phase reused the same framework but changed the evaluation target: instead of only checking whether l_A tracks h_A on model A outputs, it asked whether l_B tracks h_(A,B) when model B scores text generated by model A.

Initial efforts concentrated on building and verifying a minimal yet robust pipeline capable of tracking the following at the token level:

- Predicted probability distribution
- Observed (sampled) token
- Empirical entropy
- Log-perplexity of the sampled token

Once the reliability of this framework was established across models and prompts, subsequent experiments investigated whether these measures consistently converged as predicted by AEP. In the final phase, the same logic was extended to source-masked cross-model scoring so that cross-entropy could be estimated without collapsing the distinction between source model and scorer model.

Additional experiments pushed the models beyond typical usage constraints by generating sequences exceeding 15000 tokens in length and by rescoring archived generations under alternate models. These long-horizon and cross-model runs provided the asymptotic regime needed for both within-model convergence and cross-model comparison.

## Key Experiments

A series of targeted experiments were conducted to evaluate both convergence behavior and cross-model typicality. Taken together, these experiments show how the project evolved from validating a theoretical phenomenon to building a quantitative comparison framework.

- Basic Self-Generated Convergence Tests: Short-to-long text sequences were generated using models such as LLaMA 3.1 and Gemma variants to observe convergence of running log-perplexity and empirical entropy within the source model.
- Cross-Model Typicality Analysis: Texts generated by one model were evaluated by a different model to determine whether they fell within the scorer model's typical set. The closure phase formalized this through source-masked scoring and empirical cross-entropy h_(A,B).
- Large-Scale Generation Stress Tests: Models were prompted to generate abnormally long sequences across varied random seeds, and the resulting outputs were aggregated into comparison batches so that sequence-level and corpus-level regularities could be quantified.

## Infrastructure and Validation Strategy

The experimental infrastructure was intentionally designed to be simple, modular, and robust so that identical analysis logic could be reused across models, prompts, seeds, and later cross-model scoring passes.

Key principles guided its development:

- Consistent Hyperparameter Control: All generation and scoring experiments used clearly versioned configurations for temperature, top-p, top-k, prompt identity, and seed, preserving comparability between within-model and cross-model runs.
- Test-Driven Development: Core functions—including generation, scoring, aggregation, and report creation—were validated with unit and regression tests to reduce the risk of silent experimental drift.
- Plug-and-Play Experimentation: The system was structured to support easy substitution of models, prompts, and scorers, enabling batch-level parallelism while preserving artifact traceability and comparable outputs.

Validation strategies included:

- Manual inspection of convergence plots to verify expected behavior over sequence length
- Tracking σ-level (standard deviation) differences across tokens to assess convergence stability
- Sanity checks using known edge cases, such as degenerate probability distributions, to confirm system correctness
  Cross-run aggregation checks to verify that batch summaries, histogram statistics, and sequence-level diagnostics agreed with the underlying token-level outputs.

## Results

The results support two linked claims. First, the central theoretical prediction—that log-perplexity converges to entropy over sufficiently long self-generated sequences—was empirically validated across a range of models and prompts. Second, once those same sequences were evaluated across models, the data showed structured divergence that could be summarized using empirical cross-entropy rather than treated as a purely qualitative effect.

- For extended sequences, log-perplexity consistently stabilized within ±2σ of the empirical entropy.
- This convergence pattern held across multiple random seeds, sampling parameters, and model architectures.

In the final closure analysis, the cross-model pipeline processed 444 usable comparison files. The resulting empirical cross-entropy distribution had mean h_(A,B) = 4.1315 bits/token, median = 4.178 bits/token, minimum = 0.6547 bits/token, maximum = 8.3482 bits/token, and six skipped or failed records. These summary statistics show that cross-model behavior is neither random noise nor a single universal offset; it varies substantially by model pair and sequence.

These results support the theoretical assertion that language models generate from a statistically constrained subset of possible outputs—commonly referred to as the model's typical set.

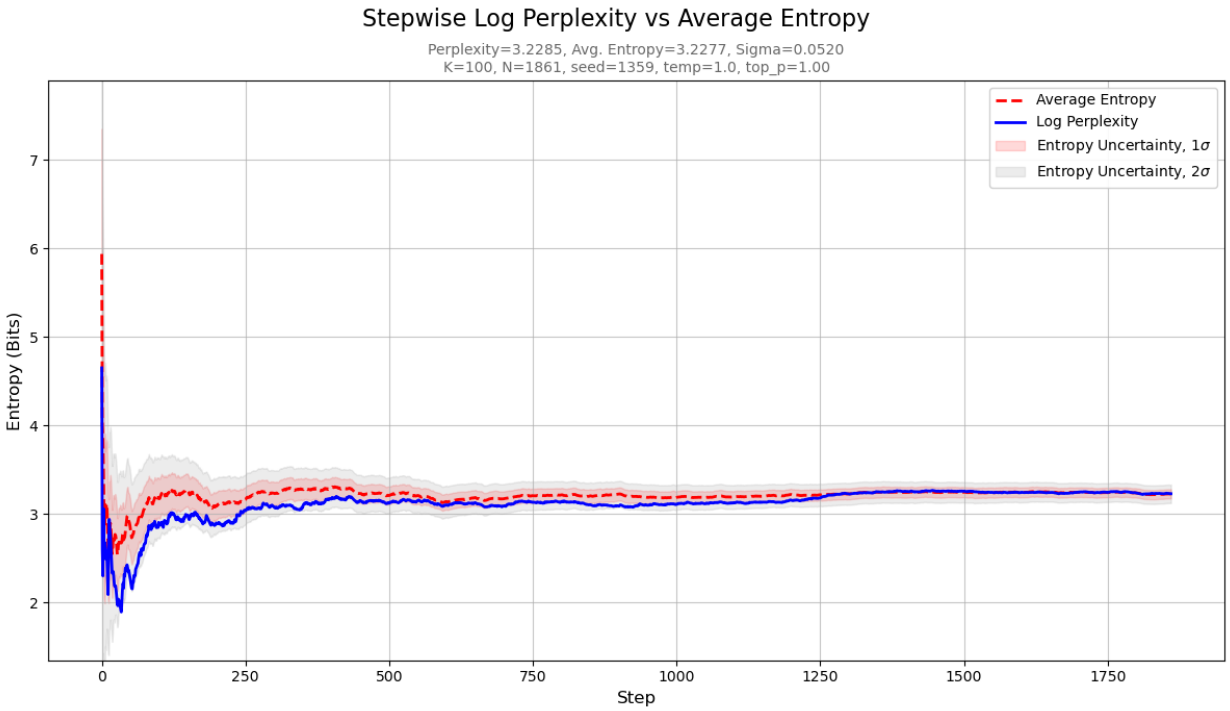


**Figure 1:** Long-Sequence Convergence of Log-Perplexity to Entropy

In cross-model scoring experiments, clear divergence emerged:

- Sequences generated by one model, when evaluated by a different model, often exhibited log-perplexity several standard deviations above the scorer's entropy baseline.
- The difference Δ = (log-perplexity – entropy) served as a practical indicator of under-typicality or over-typicality, quantifying how unexpected a sequence was under the scoring model.
- This divergence persisted even within model families (e.g., Gemma 1B vs. Gemma 12B), suggesting meaningful differences in token distribution behavior

These findings indicate that language models encode statistically unique output distributions, which could, in principle, be leveraged for model classification or detection.

**Figure 2** illustrates a single-sequence divergence test, where expected token log-perplexity fails to align with the model's entropy baseline over time—strongly suggesting under-typicality when text is evaluated by a non-generating model.

**Figure 3** expands this insight to a broader set of cross-model comparisons. Each point represents a text sequence generated by Gemma 12B and evaluated by LLaMA 3.1 8B. The consistent deviations from the entropy baseline confirm that different models exhibit statistically distinguishable generative behavior.

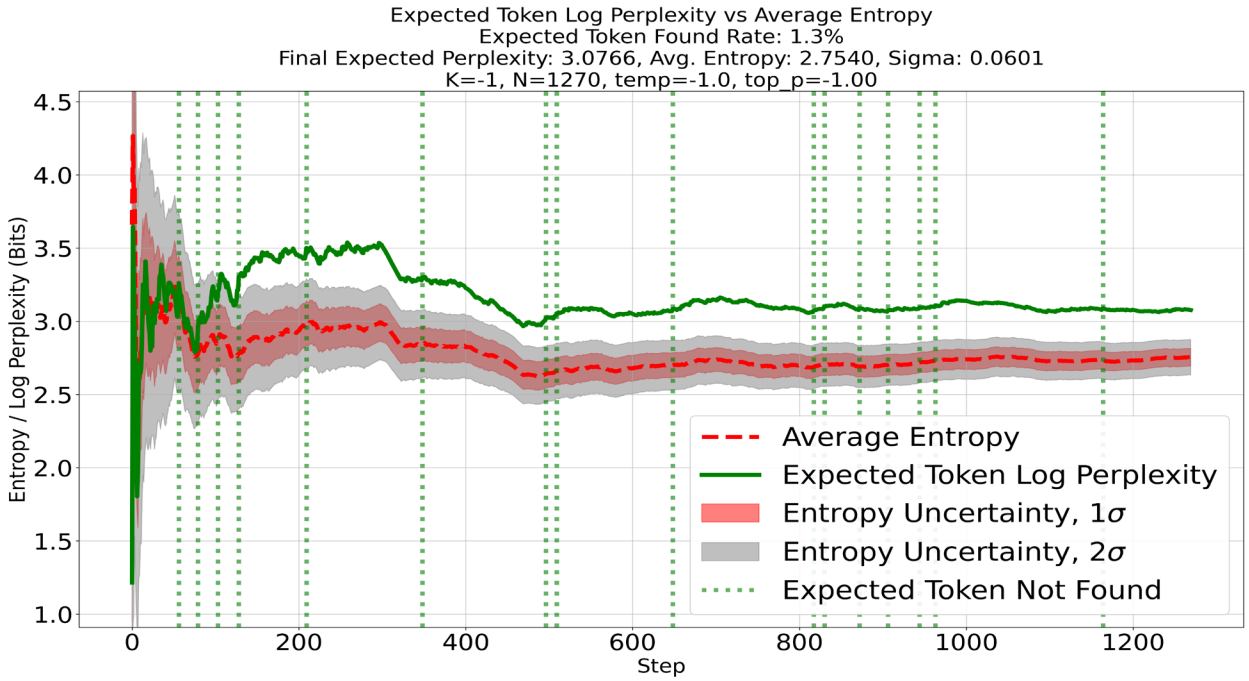


**Figure 2:** Divergence Between Entropy and Log-Perplexity in Single Sequence Evaluation

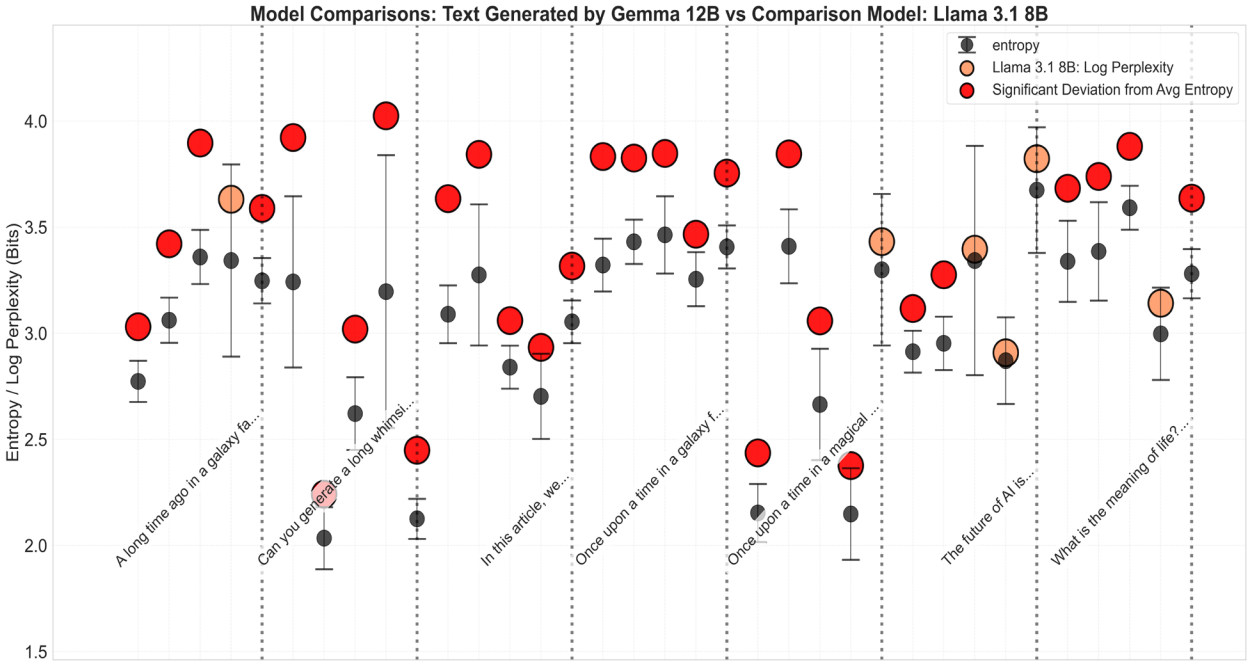


**Figure 3:** Cross-model evaluation shows divergence in typicality.

## Sequence Length and Scaling Behavior

Longer text generations revealed increasingly constrained statistical behavior, consistent with the predictions of AEP and with the final phase emphasis on asymptotic measurement. The longer the sequence, the more stable both entropy-tracking and cross-model summary estimates became.

- The proportion of tokens falling within $\pm 1\sigma$ of the model's entropy steadily increased with sequence length.
- Variance in log-perplexity decreased over time, indicating stronger convergence as the number of generated tokens grew.

These scaling effects provide strong empirical evidence that AEP is not merely a theoretical construct but a measurable and observable property of outputs from autoregressive transformer models.

## Detection of Training Data

An exploratory secondary result emerged when scoring classic literary texts such as The Great Gatsby and Alice in Wonderland:

- These texts exhibited over-typicality, with log-perplexity values falling several standard deviations *below* the empirical entropy baseline.
- This behavior strongly suggests that the texts were memorized during training, allowing the model to predict tokens with unusually high confidence.

These findings demonstrate that the theoretical framework can be leveraged not only for analyzing model behavior but also for identifying content likely included in a model's training dataset—offering a principled method for training data inference.

## Scaling Properties

The experimental findings strongly validated the scaling behavior predicted by the underlying theoretical framework:

- As text length increased, the statistical constraints governing model outputs became more pronounced.
- The relative size of the typical set, compared to the full space of grammatically valid strings, decreased exponentially with sequence length.
- These results support the mathematical conclusion that language models are restricted to generating from a vanishingly small subset of all possible outputs.

To confirm AEP behavior across models, we aggregated all 583 generations.

**Figure 5** shows the distribution of final entropy–perplexity differences for each model. All are tightly centered near zero, confirming consistent convergence regardless of architecture. Slight variance in sharpness reflects model size and sampling behavior, but the overall pattern holds.

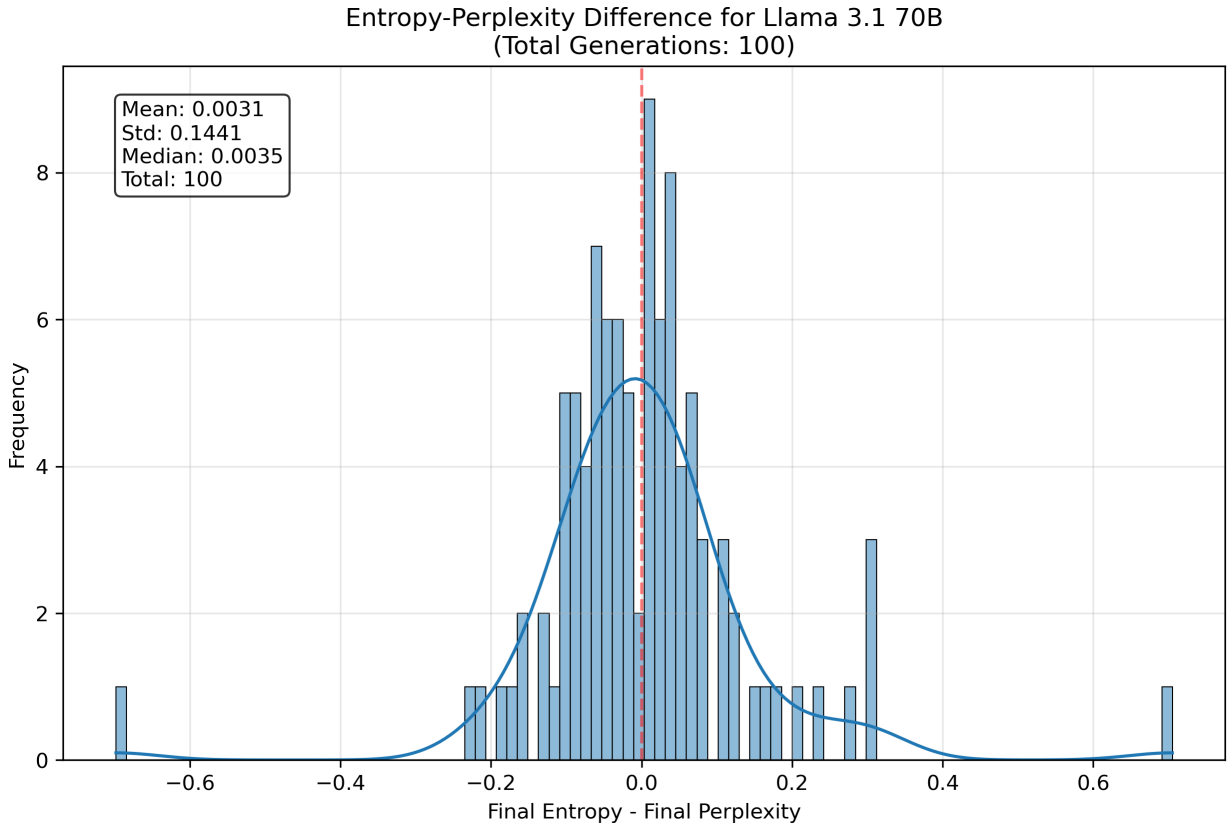


**Figure 4:** Distribution of Final Entropy–Perplexity Differences (LLaMA 3.1 70B)

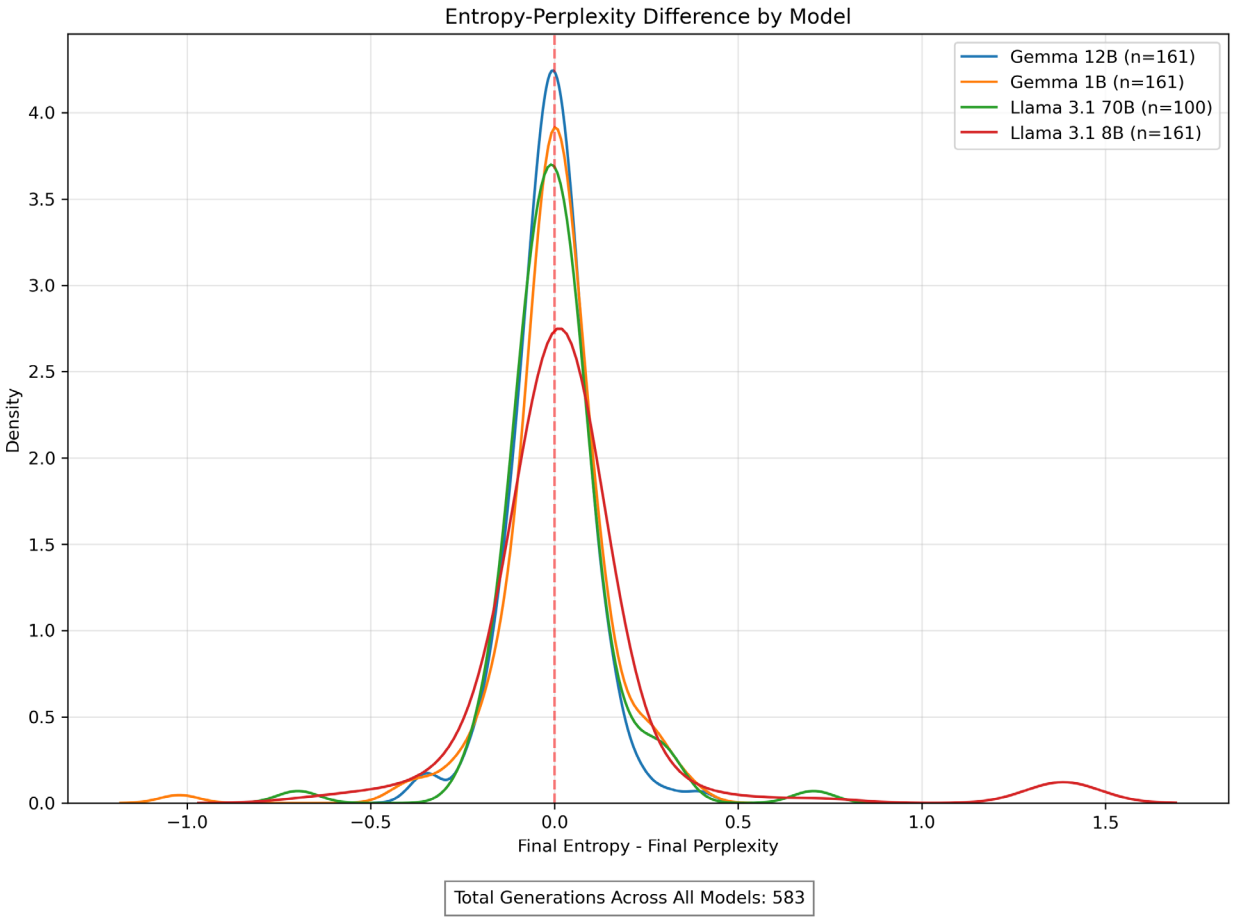


**Figure 5:** Entropy–Perplexity Convergence Across All Models

## Discussion

### *Integrated Final-Phase Cross-Model Results*

The final phase did not replace the original results; it sharpened them. After establishing within-model convergence, the project added cross-model source-masked scoring to quantify how far a scorer model departs from the source model's statistical expectations. The figures below summarize that closure phase and should be read as continuations of the earlier convergence plots rather than as a separate project.

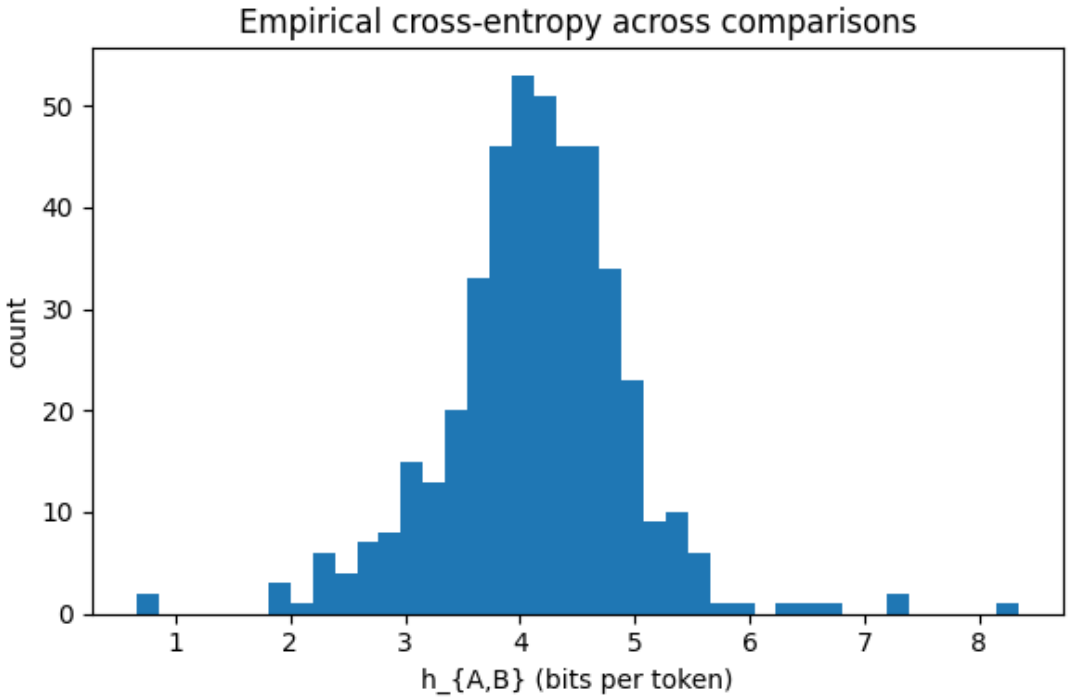


**Figure 6. Aggregate distribution of cross-model cross-entropy h_(A,B).**
Interpretation: The histogram is centered around a moderate cross-entropy level with tails spanning very low and very high divergence. This shows that cross-model comparisons occupy multiple regimes rather than collapsing into a single notion of mismatch.

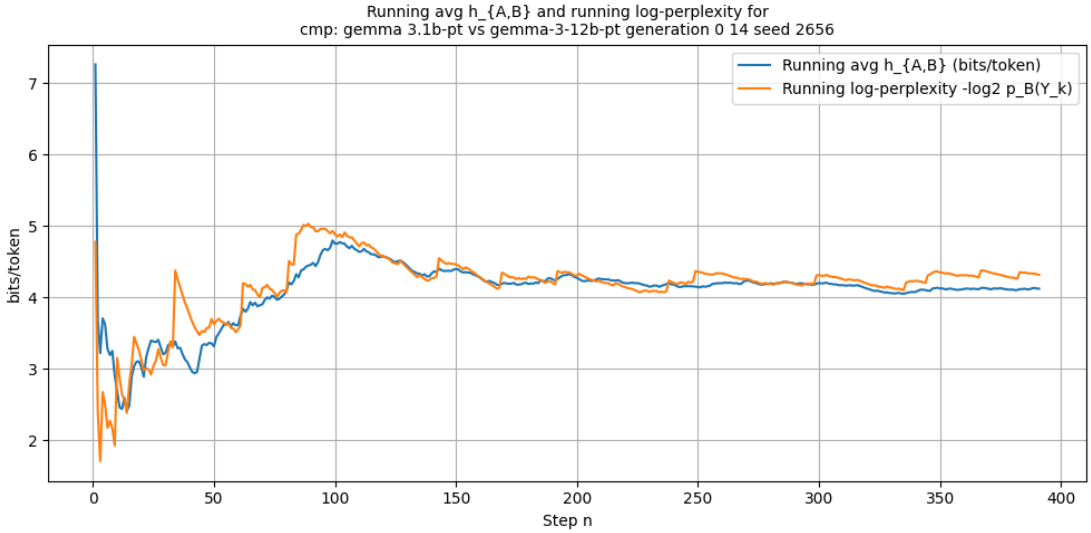


**Figure 7. Running average h_(A,B) and running log-perplexity for a representative mid-divergence sequence.**
Interpretation: The running-average curve stabilizes after an initial transient, while running log-perplexity tracks nearby. This supports the cross-AEP interpretation that long-horizon scorer behavior can be summarized by empirical cross-entropy. That is, After an initial transient, the plot's two curves settle near each other, which is what you'd expect if "B's difficulty predicting A's sampled path" matches "B's difficulty matching A's predicted distribution," once you've seen enough tokens.

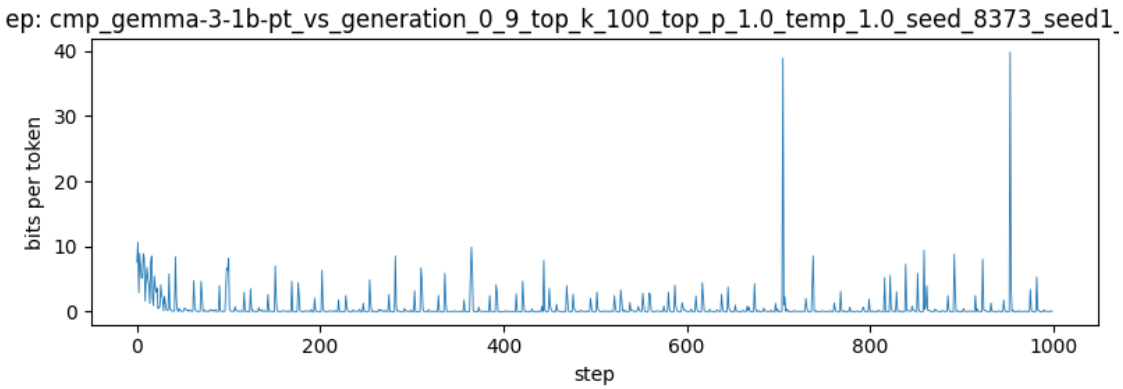


**Figure 8. Per-step cross-entropy profile for a low-divergence sequence.**
Interpretation: Per-step fluctuations persist, but the sequence remains in a relatively low-divergence band. Local token uncertainty therefore does not eliminate the broader sequence-level regularity.

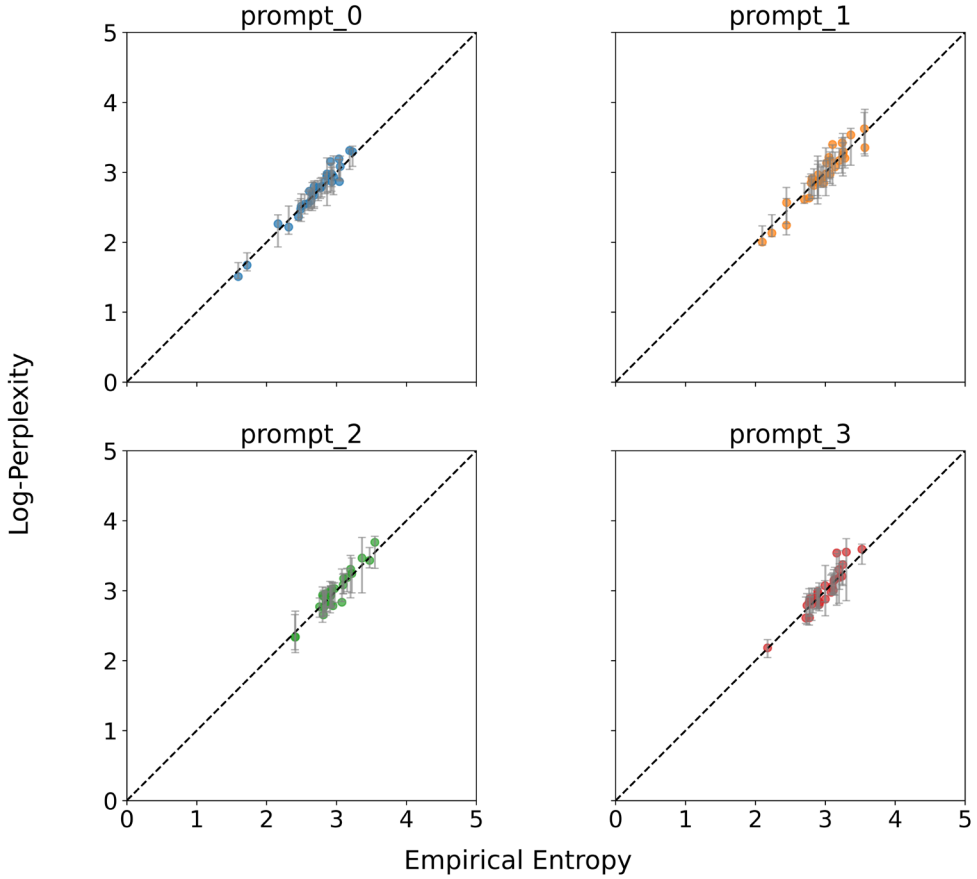


**Figure 9. Combined 2×2 entropy/perplexity diagnostic view for Gemma-3-1b runs.**
Interpretation: The combined panel provides multi-view consistency checks across trajectory, distribution, and summary behavior. Agreement across views strengthens confidence that the convergence signal is not an artifact of any single prompt type.

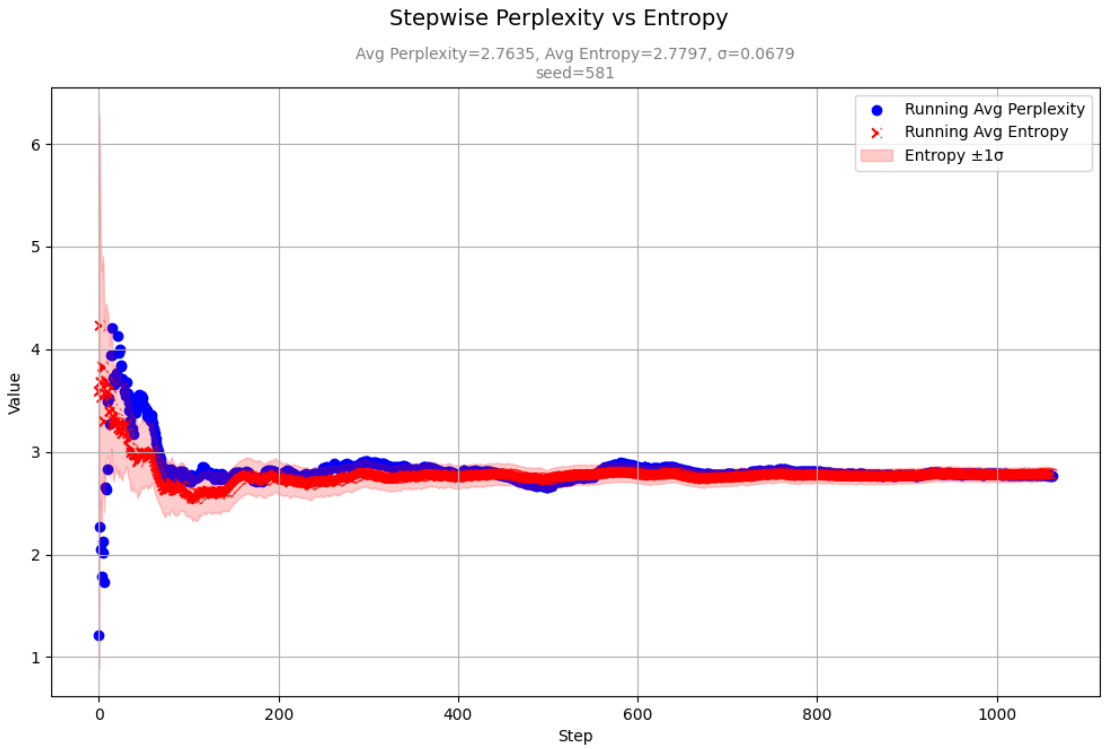


**Figure 10. Cross-model entropy vs. perplexity comparison for a Gemma-3-12b comparison run.**
Interpretation: Gemma 3 12B is the scorer/comparison model used to compute entropy/perplexity along the Llama-generated string, while Llama 3.1 8B is the generator of the string. This comparison shows how scorer-model behavior departs from self-model baselines while remaining structured enough for quantitative analysis. It is a practical visualization of typical-set divergence.

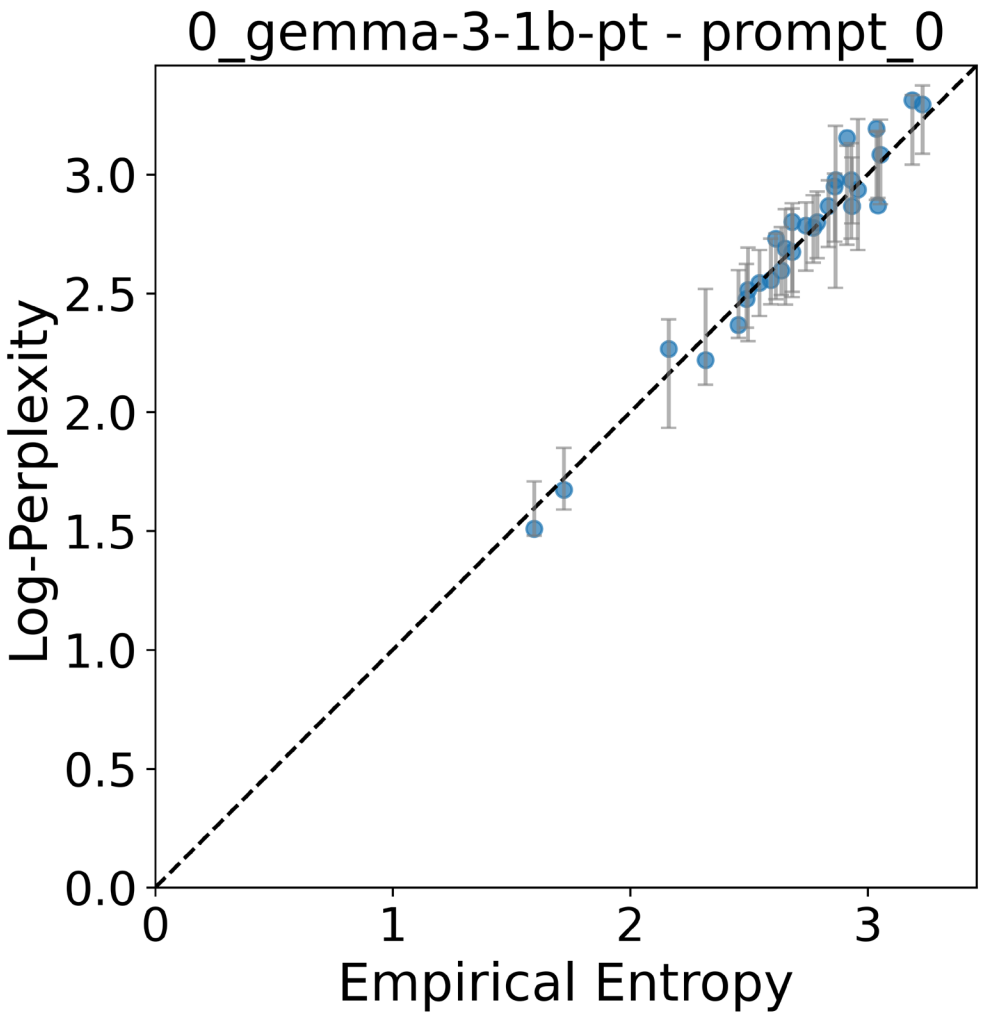


**Figure 12. Prompt 0 scatter: final entropy vs. final log-perplexity clustering behavior.**
Interpretation: Prompt-conditioned scatter remains clustered near a convergence manifold with prompt-specific spread, suggesting robust core behavior with understandable prompt dependence.

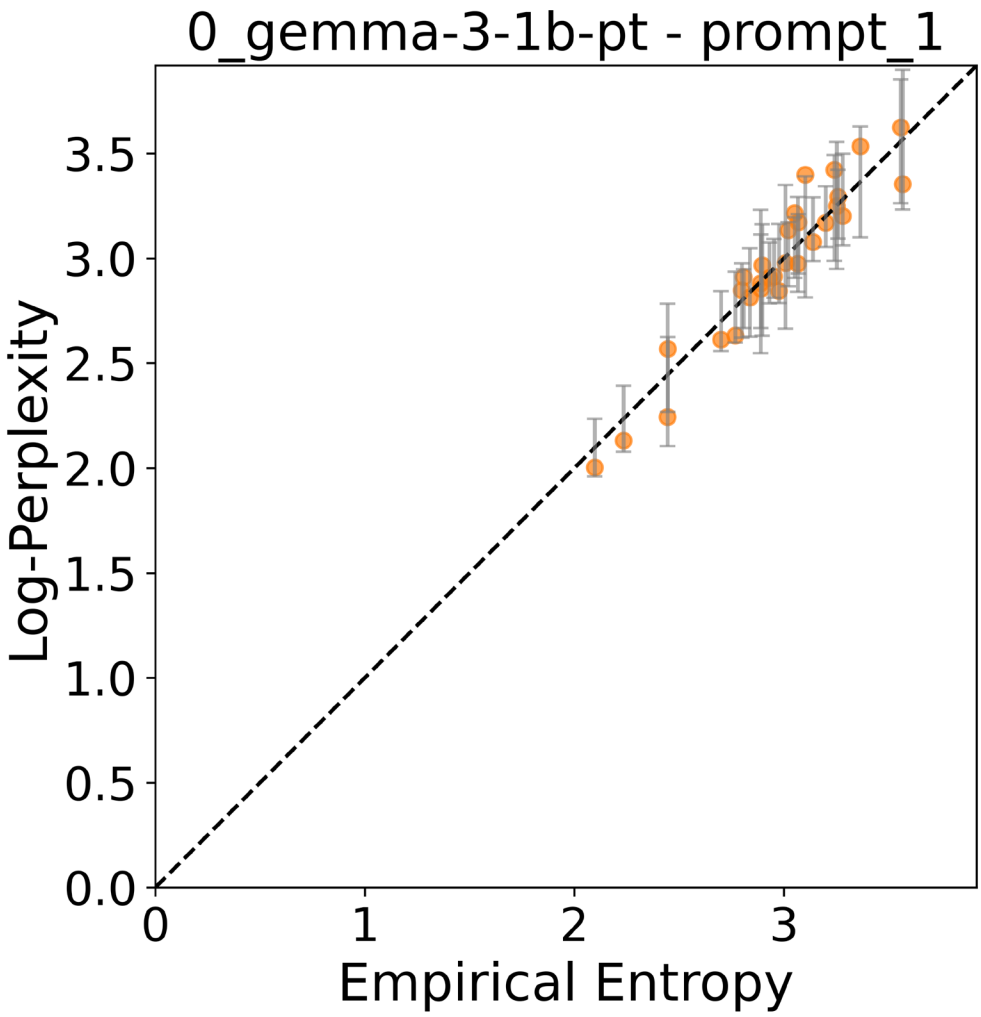


**Figure 13. Prompt 1 scatter: run-to-run spread and concentration near the convergence line.**
Interpretation: The second prompt reproduces the same overall geometry, indicating that the convergence relationship is stable across repeated runs rather than tied to a single prompt template.

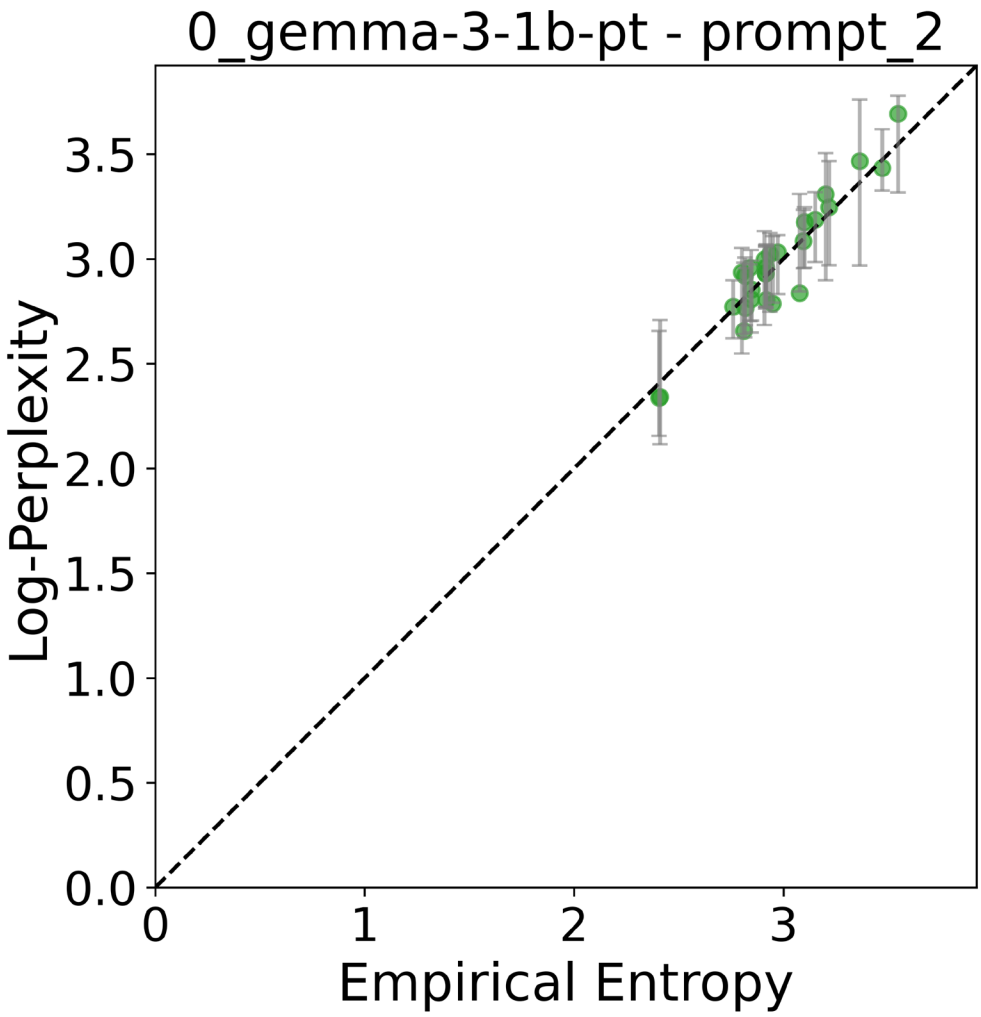


**Figure 14. Prompt 2 scatter: prompt-specific dispersion relative to the typical-set baseline.**
Interpretation: Prompt-specific dispersion changes the spread but does not erase the central alignment between final entropy and final log-perplexity summaries.

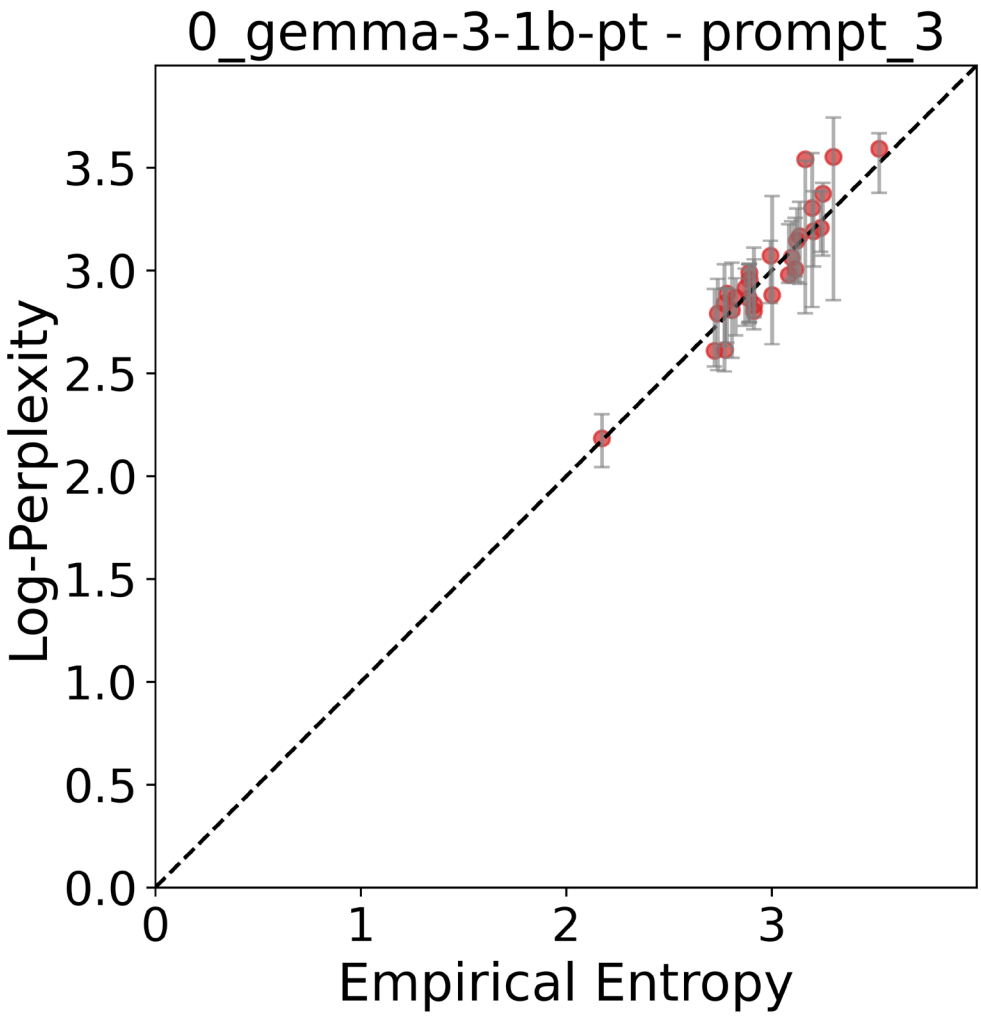


**Figure 15. Prompt 3 scatter: robustness check across prompt-conditioned generations.**
Interpretation: This final scatter view provides an additional robustness check and reinforces the conclusion that self-model convergence and cross-model divergence coexist in a stable, interpretable way.

The results of this project provide both theoretical insight and practical tools for understanding the statistical behavior of generative language models. By empirically validating the Asymptotic Equipartition Property (AEP), the project offers a principled explanation for how and why model outputs cluster within probability space. These findings create a foundation for potential applications in model fingerprinting, AI-generated content detection, and training data inference.

## Theoretical Implications

This work reinforces the connection between information theory and deep learning, demonstrating that even highly complex autoregressive models remain governed by classical statistical laws:

- The observed convergence of log-perplexity to entropy confirms that long-form text generation in language models reflects predictable statistical behavior.
- These results support the hypothesis that model outputs are not arbitrary samples from the full language space, but instead lie within a vanishingly small “typical set.”

Importantly, these findings were achieved using real-world language models without relying on idealized assumptions such as token independence. All results were derived from observable statistics, strengthening the practical relevance of the theory.

## Engineering Contributions

While the theoretical framework was the project's primary focus, substantial engineering work was also completed to enable scalable, reproducible experimentation:

- Developed a clean, test-driven framework for token-level analysis across multiple models and prompts
- Refactored and modernized a legacy codebase to improve robustness, version control, and extensibility
- Built a modular experimentation pipeline capable of handling long-sequence generation and cross-model comparisons

This infrastructure lays the groundwork for future experimentation with minimal additional overhead.

## Challenges and Limitations

Several technical constraints shaped the project's scope and interpretation:

- Quantitative Cross-Model Metrics Remain Incomplete: While the closure phase introduced empirical cross-entropy summaries, a full theoretical treatment of confidence intervals, formal tests, and model-pair effect decomposition remains future work.
- Compute and Model Availability Constraints: Access to all desired model sizes and families was limited, which restricted the breadth of scaling comparisons and left some large-model cross-combinations unexplored.
- External Text Benchmarking Was Secondary: The primary emphasis remained on internally generated and cross-scored data, so broader benchmarking on curated external corpora should still be expanded before making strong generalization claims.

Even with the final quantitative closure, important questions remain about calibration, uncertainty, and how much of the observed divergence is attributable to architecture, size, tokenizer differences, or decoding choices. These limitations do not weaken the core empirical findings, but they define the next layer of required analysis.

# Ideas and Aims for Future Research

Building on the theoretical foundations and empirical findings of both project phases, several future directions follow naturally. These are not separate ideas from two different reports; they are the direct continuation of the same research arc from self-model AEP validation to scalable cross-model typicality analysis.

*1. Integration of Reasoning Models*

While this project focused on traditional generative models, future work could explore how the Asymptotic Equipartition Property (AEP) manifests in reasoning-focused architectures:

- Compare typical set behavior between standard autoregressive models and models optimized for logical or deductive reasoning.
- Analyze whether reasoning models exhibit different convergence rates or entropy characteristics.
- Investigate whether statistical constraints identified here are general to all transformer-based systems or specific to generative objectives.
  Extend the comparison to newer reasoning-oriented systems and test whether cross-AEP signatures remain stable when generation objectives shift from generic continuation to chain-of-thought-heavy reasoning.

This line of inquiry may reveal structural differences in how various model types process, encode, and constrain language.

*2. Database-Driven Large-Scale Analysis*

The current implementation reached the practical limits of manual file-based data management. Transitioning to a structured database system would enable more comprehensive analysis:

- Store token sequences, probability distributions, and convergence metrics in an indexed, queryable format.
- Automate large-scale experimentation and result aggregation across models, prompts, and configurations.
- Facilitate more sophisticated statistical queries, longitudinal tracking, and comparative studies.
- Enable high-throughput testing of AEP behavior across different architectures and training conditions.
  Integrate schema validation, CLI-based experiment orchestration, and continuous-integration checks so that future comparison batches can be reproduced and audited automatically.

This infrastructure upgrade would dramatically scale both the scope and depth of future experiments.

*3. Hyperparameter Sensitivity and Typical Set Behavior*

Sampling hyperparameters—such as temperature, top-p, and top-k—may influence typical set boundaries and convergence patterns. Systematic exploration could reveal critical insights:

- Quantify how specific sampling parameters affect convergence of log-perplexity to entropy.
- Determine whether reducing randomness increases alignment between models' typical sets.
- Assess whether hyperparameter manipulation can obscure model identity, complicating detection efforts.
- Explore how these settings impact the size, structure, and boundaries of a model's typical set.
  Add uncertainty quantification—confidence intervals, hypothesis tests, and sensitivity analyses—to determine which divergences are statistically robust rather than visually suggestive.

Such research could benefit both detection systems and users seeking to optimize generation quality.

*4.. Practical Application Development*

The empirical validation of typical set behavior opens the door to several impactful applications:

- Develop AI-generated text detectors using statistically principled thresholds.
- Create tools for training data inference, identifying whether a text likely appeared in a model's training set.
- Implement model attribution systems that determine which model family likely produced a given text.
- Explore adversarial prompting to test the boundaries and robustness of typical set constraints.
  Translate h_(A,B)-based diagnostics into downstream features for attribution, detection, and training-data inference systems.

These applications could contribute to transparency, attribution, and security in the deployment of generative AI systems.

The progress achieved in this project provides a solid foundation for these future directions. The combination of theoretical rigor and experimental validation offers a framework that can be extended and adapted to address increasingly complex questions in the field of language modeling and AI safety.

Resources used (this section does not count in the page-length of the report):
Final report also requires a list of all resources you used/needed for the semester-long project – please list these resources in the following categories:

- Data Management for AI/ML
  - Git
  - Git-LFS
  - Local and iibi machines
- AI/ML Methods and Approaches and Tools
  - Seaborn
  - Pandas
  - Large Language Models (Llama family, Gemma family, GPT2)
- AI/ML Frameworks
  - TDD
  - Regression testing
  - Dry Code
- AI/ML Interfaces
  - Hugging Face Interface
  - Pytorch
- Operational AI Deployment
  - None