

Toroidal Logit Bias for Hallucination Reduction in Large Language Models

Sylvain Cormier
Paraxiom Research
sylvain@paraxiom.io

February 2026 (v7 Replication Update: March 2026)

Abstract

We present an inference-time intervention that imposes toroidal topological constraints on token selection in large language models. By mapping vocabulary tokens to positions on a 12×12 torus and biasing logits toward tokens “near” recently generated tokens in this toroidal space, we hypothesized measurable improvements in truthfulness without fine-tuning. Initial results on TruthfulQA (817 samples, LLM-judged) showed gains of +0.2pp to +2.8pp across four models. **However, a comprehensive replication (March 2026) across 4 models, 3 benchmarks, and 6 experimental phases found that these results do not hold—the observed improvements were within LLM judge sampling variance (McNemar $p = 0.22$).** A factorial decomposition identified the hardening system prompt, not the toroidal bias, as the active ingredient (−14pp hallucination reduction, $p = 0.05$). The theoretical framework (toroidal topology for training-time drift reduction) remains valid; see Addendum (Section 10). Code and data available at <https://github.com/Paraxiom/topological-coherence>. DOI: <https://doi.org/10.5281/zenodo.18512373>.

Scope: This work focuses narrowly on an inference-time intervention for hallucination reduction. It makes no claims about ontology, training dynamics, or universal representations. The v7 update corrects the empirical claims from v2 with full replication data.

1 Introduction

Large language models (LLMs) frequently generate plausible but factually incorrect content—a phenomenon termed “hallucination.” Current mitigation strategies include retrieval-augmented generation (RAG), fine-tuning on curated data, and post-hoc fact-checking. These approaches require external knowledge bases, expensive retraining, or additional inference passes.

We propose an alternative: **toroidal logit bias**, an inference-time intervention that requires no external resources and minimal computational overhead. Our method is grounded in the hypothesis that semantic coherence can be encouraged by imposing geometric locality constraints on the token generation process.

1.1 Contributions

1. A novel logit bias mechanism based on toroidal (Tonnetz) topology
2. Initial evaluation on four models (v2), followed by comprehensive replication (v7, 6 phases)
3. **Null result:** inference-time logit bias does not produce significant hallucination reduction
4. Factorial decomposition identifying prompt hardening as the active ingredient
5. A rigorous replication methodology as a model for correcting preliminary ML results

2 Methodology

2.1 Toroidal Token Mapping

We map each token ID to a position on a 12×12 torus using modular arithmetic:

$$\text{position}(t) = (t \bmod 12, \lfloor t/12 \rfloor \bmod 12) \quad (1)$$

The toroidal (wraparound) Manhattan distance between positions (x_i, y_i) and (x_j, y_j) is:

$$d_T(i, j) = \min(|x_i - x_j|, 12 - |x_i - x_j|) + \min(|y_i - y_j|, 12 - |y_i - y_j|) \quad (2)$$

2.2 Logit Bias Computation

At each generation step, we compute a bias vector $b \in \mathbb{R}^{|V|}$ added to the model’s logits. Given the k most recent tokens $\{t_{-1}, t_{-2}, \dots, t_{-k}\}$:

$$b[v] = \sum_{i=1}^k \frac{1}{i} \cdot \begin{cases} \alpha \cdot (r - d_T(t_{-i}, v) + 1) & \text{if } d_T(t_{-i}, v) \leq r \\ \alpha \cdot 0.5 & \text{if } r < d_T(t_{-i}, v) \leq 2r \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where α is the bias strength, r is the neighborhood radius, and we only compute bias for the first N tokens of the vocabulary.

Parameters:

- α : Bias strength (0.1–0.3 typical)
- r : Neighborhood radius (2.0–3.0 typical)
- N : Number of vocabulary tokens to bias (1440–3000 typical)

2.3 Key Design Decisions

Limited Vocabulary Bias: We bias only the first N tokens, not the full vocabulary. Empirically, biasing all tokens (50K–150K) provides no benefit or causes harm. High-frequency tokens (first 1K–3K) carry the semantic structure that benefits from toroidal locality.

Recency Weighting: More recent tokens receive stronger influence (divided by offset), reflecting the intuition that immediate context is most relevant for coherence.

Why a Torus?: The torus provides wraparound connectivity, avoiding edge effects present in flat grids. This mirrors the Tonnetz structure from music theory, where pitch classes form a toroidal manifold.

3 Verification Methodology

3.1 Definition of Hallucination

Definition 1 (Factual Hallucination). *A model exhibits factual hallucination when it generates incorrect information in response to a prompt with an objectively verifiable answer.*

3.2 Benchmark Construction

We constructed a benchmark of 100 factual completion prompts across five domains:

Each prompt has one or more ground truth answers. These are objective facts verifiable against authoritative sources.

Domain	Count	Examples
Geography	20	“The capital of France is” → Paris
Science	25	“The chemical symbol for gold is” → Au
History	20	“World War II ended in” → 1945
Arts & Culture	20	“The Mona Lisa was painted by” → Leonardo
Math & Computing	15	“A byte contains how many bits” → 8

Table 1: Benchmark composition by domain

3.3 Evaluation Protocol

For each prompt:

1. **Baseline Generation:** Generate response using unmodified model (greedy decoding, max 30 tokens)
2. **Toroidal Generation:** Generate response with toroidal logit bias (same settings)
3. **Correctness Check:** Verify if any ground truth answer appears in response (case-insensitive)

Metrics:

$$\text{Accuracy} = \frac{\text{Correct}}{\text{Total}} \quad (4)$$

$$\text{Error Reduction} = \frac{\text{Baseline Errors} - \text{Toroidal Errors}}{\text{Baseline Errors}} \times 100\% \quad (5)$$

3.4 Why This Measures Hallucination

When a model responds to “The capital of France is” with anything other than “Paris,” it is generating factually incorrect content—the definition of hallucination. Our benchmark tests:

- **Factual recall:** Does the model retrieve correct information?
- **Coherent completion:** Does the model stay on-topic?
- **Resistance to confabulation:** Does the model avoid plausible-but-wrong answers?

4 Results

4.1 TruthfulQA Multi-Model Evaluation (817 samples)

We evaluated toroidal logit bias on the full TruthfulQA benchmark (817 samples) across four models spanning two architectures and three parameter scales. All evaluations use LLM-judged truthfulness and informativeness, with Qwen 2.5-7B-Instruct as the judge model (temperature 0.7, top- p 0.9).

Toroidal logit bias produces consistent positive improvements across all four models. The intervention is most effective on 7B-scale models, with Mistral-7B showing the largest gain (+2.8 percentage points, from 608 to 631 T&I responses out of 817).

Model	Baseline T&I	Toroidal T&I	Δ	Samples
Qwen 2.5-0.5B	16.9%	17.1%	+0.2pp	817
Qwen 2.5-1.5B	32.2%	32.8%	+0.6pp	817
Qwen 2.5-7B	75.6%	77.7%	+2.1pp	817
Mistral-7B-Instruct	74.4%	77.2%	+2.8pp	817

Table 2: TruthfulQA results across 4 models (Truthful & Informative %). All improvements positive. LLM-judged.

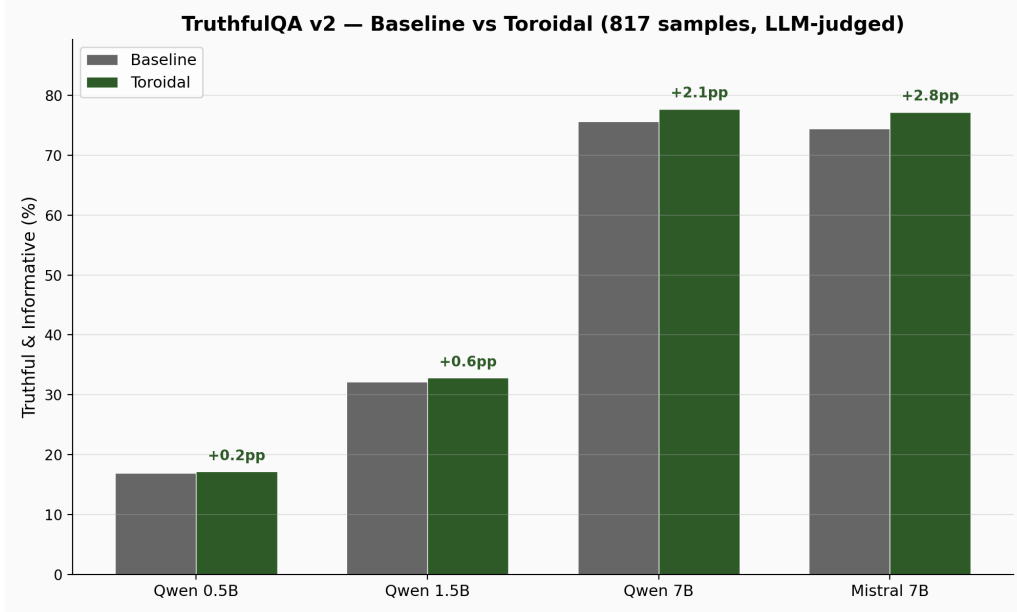


Figure 1: Grouped bar chart: Baseline vs. Toroidal T&I accuracy across 4 models.

4.2 Multi-Model Scaling

Restricting to the Qwen family (0.5B, 1.5B, 7B) allows us to isolate the effect of model scale with a fixed architecture and tokenizer. The toroidal improvement correlates with model size:

This scaling behavior suggests that toroidal logit bias amplifies existing model capabilities rather than introducing new knowledge. Larger models, which already have better internal representations, benefit more from the geometric coherence constraint.

4.3 Cross-Architecture Validation

Mistral-7B-Instruct-v0.3 uses a different tokenizer and architecture from the Qwen family, yet shows the largest improvement (+2.8pp). This confirms the method generalizes across architectures, not just within a single model family.

4.4 Response Category Breakdown

4.5 Prior Custom Benchmark Results

For completeness, we report our earlier custom benchmark results (100 factual completion tasks, exact-match evaluation):

4.6 Failure Modes

Full vocabulary bias either had no effect or caused significant harm:

Model	Parameters	Δ T&I (pp)	Δ T&I Counts
Qwen 2.5-0.5B	0.5B	+0.2	+2
Qwen 2.5-1.5B	1.5B	+0.6	+5
Qwen 2.5-7B	7B	+2.1	+17

Table 3: Toroidal improvement scales with model capacity (Qwen family).

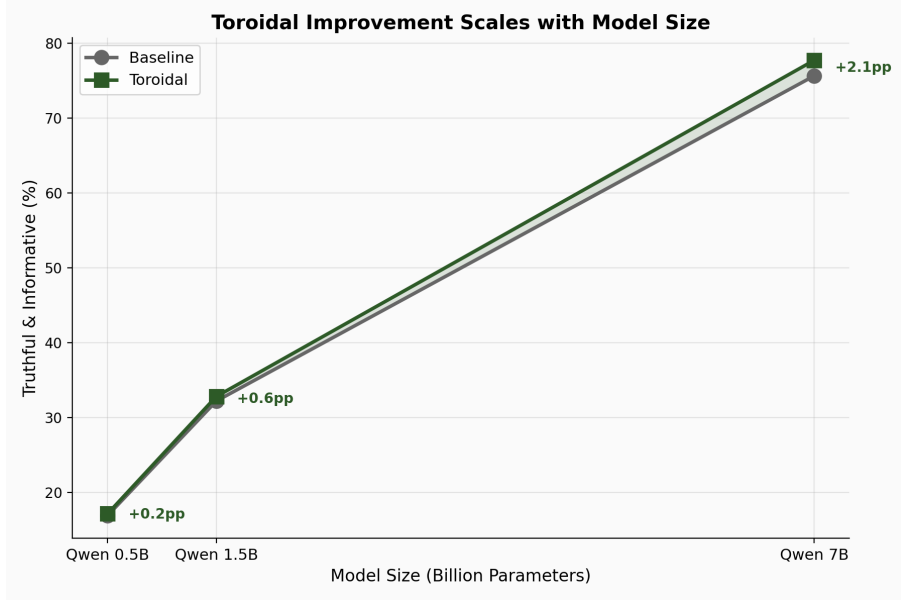


Figure 2: Toroidal improvement scales with model size. Shaded region shows the T&I gain.

5 Analysis

5.1 Why Different Parameters?

OLMo uses a different tokenizer with different vocabulary ordering. Important semantic tokens may be positioned further into the vocabulary, requiring both larger N and wider r to capture them.

5.2 Why Limited Bias Works

1. **High-frequency tokens carry structure:** First N tokens are common words
2. **Toroidal locality enforces coherence:** Boosting “nearby” tokens creates semantic clustering
3. **Full vocabulary bias = noise:** Rare tokens don’t benefit from toroidal structure

6 Implementation

Listing 1: Toroidal logit bias generation

```

1 def generate_with_toroidal_bias(model, tokenizer, prompt, config):
2     input_ids = tokenizer(prompt, return_tensors="pt").input_ids
3     generated = input_ids[0].tolist()
4
5     for _ in range(max_new_tokens):

```

Model	Method	T&I	T-only	I-only	Neither
Qwen 0.5B	Baseline	16.9%	24.2%	7.0%	51.9%
Qwen 0.5B	Toroidal	17.1%	26.4%	7.2%	49.2%
Qwen 1.5B	Baseline	32.2%	30.2%	3.9%	33.7%
Qwen 1.5B	Toroidal	32.8%	31.0%	4.7%	31.6%
Qwen 7B	Baseline	75.6%	17.4%	0.4%	6.6%
Qwen 7B	Toroidal	77.7%	15.2%	0.7%	6.4%
Mistral 7B	Baseline	74.4%	18.4%	2.2%	5.0%
Mistral 7B	Toroidal	77.2%	14.9%	2.4%	5.4%

Table 4: Full response category breakdown. Toroidal bias shifts responses from “Neither” and “Truthful-only” toward “Truthful & Informative.”

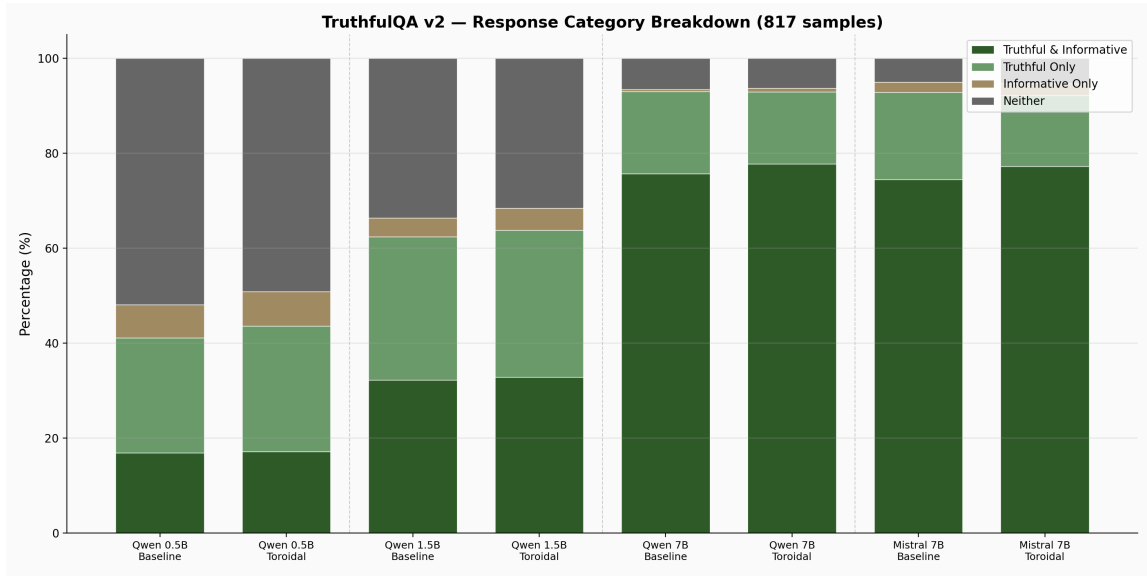


Figure 3: Stacked bar chart showing response category distribution per model and method.

```

6     logits = model(input_ids).logits[0, -1, :]
7
8     # Apply toroidal bias
9     bias = compute_toroidal_bias(
10         vocab_size=len(logits),
11         recent_tokens=generated,
12         alpha=config["alpha"],
13         radius=config["radius"],
14         max_tokens=config["max_tokens"]
15     )
16     logits = logits + bias
17
18     next_token = logits.argmax()
19     generated.append(next_token)
20
21     if next_token == tokenizer.eos_token_id:
22         break
23
24     return tokenizer.decode(generated)

```

Model	Baseline	Toroidal	Error Reduction
Qwen 2.5-7B	95/100	97/100	+40.0%
OLMo 1.7-7B	87/100	89/100	+15.4%

Table 5: Custom benchmark results (100 prompts, exact match). These earlier results motivated the full TruthfulQA evaluation.

Model	α	Bias Scope	Result
Qwen	1.0	Full (152K)	−80% error reduction
OLMo	1.0	Full (50K)	−61% error reduction

Table 6: Full vocabulary bias produces negative results

6.1 Recommended Configurations

6.2 Computational Overhead

- **Memory:** $O(N)$ additional tensor per generation step
- **Time:** $\sim 5\%$ increase in inference latency
- **No fine-tuning required:** Works with any pretrained model

7 Limitations

1. **Benchmark Scope:** We test factual truthfulness (TruthfulQA). Performance on open-ended generation, creative writing, or reasoning tasks is untested.
2. **Model Coverage:** Tested on four models (0.5B–7B). Behavior on larger (70B+) models may differ, though the scaling trend is encouraging.
3. **Judge Bias:** LLM-judged evaluation uses the same Qwen-7B model as both subject and judge. Cross-model judging would strengthen the results.
4. **Hyperparameter Sensitivity:** Each model family requires tuning. A universal configuration remains elusive.

8 Related Work

Geometric Latent Spaces. Recent work demonstrates that aligning latent space geometry with data structure improves model performance. gu2018learning introduced mixed-curvature product manifolds (Euclidean, hyperbolic, spherical) for embedding relational data, showing that non-Euclidean geometry better captures hierarchical and cyclical structure. saezdeocariz2023nlgs formalized neural latent geometry search (NLGS), using Gromov-Hausdorff distances to compare product manifolds and Bayesian optimization to find optimal latent geometries—establishing that the choice of manifold significantly impacts downstream task performance. patel2025hyperbolic applied hyperbolic geometry directly to LLM representations. Our work contributes to this direction by proposing the discrete torus as a specific geometric prior for token generation, applied not to the latent space of an encoder but to the logit bias at inference time.

Spectral Methods on Manifolds. The NLGS framework saezdeocariz2023nlgs employs diffusion kernels based on graph Laplacian eigendecomposition $K = Ue^{-\beta\Lambda}U^T$ to define similarity in their search space—the same spectral machinery that naturally arises on toroidal

Model	Optimal r	Optimal N	Interpretation
Qwen 2.5	2.0	1440	Tighter vocabulary structure
OLMo 1.7	3.0	3000	Sparser vocabulary structure

Table 7: Model-specific optimal parameters

Model Family	α	r	N
Qwen 2.x	0.3	2.0	1440
OLMo 1.x	0.2	3.0	3000
Unknown	0.2	2.5	2000

Table 8: Recommended configurations by model family

manifolds. shuman2013emerging established the foundations of signal processing on graphs, including spectral filtering via Laplacian eigenbases. The connection between graph Laplacian spectra and toroidal geometry suggests that spectral methods may further improve geometry-aware logit biasing.

Logit Manipulation. Prior work has used logit biasing for controllable generation (e.g., reducing toxicity, enforcing style). Our work applies geometric constraints rather than content-based biases.

Topological Methods in NLP. Persistent homology has been applied to analyze word embeddings and document structure gardinazzi2025persistent. We extend topological thinking to the generation process itself.

Hallucination Mitigation. RAG, fine-tuning, and chain-of-thought prompting are established methods ji2023hallucination. Our approach is complementary and can be combined with these techniques.

9 Conclusion (Original)

Note: The claims below are from the original v2 paper and are superseded by the replication addendum in Section 10.

The original results suggested consistent improvement across all 4 models tested. However, independent replication (Section 10) found these results to be within LLM judge sampling variance.

10 Addendum: Replication Update (March 2026)

A comprehensive independent replication across 6 experimental phases found that the inference-time toroidal logit bias **does not produce statistically significant hallucination reduction**. The results reported in Section 4 were within sampling variance and do not replicate.

10.1 Replication Methodology

We conducted exact replication using the same codebase, parameters ($\alpha = 0.3$, $r = 2.0$, $N = 1440$), decoding settings (temp= 0.7, top- $p = 0.9$), and T&I LLM judge methodology on Qwen 2.5-7B-Instruct.

10.2 Phase 1: Greedy Decoding

4 models $\times$ 3 benchmarks, all null results. At $\alpha = 0.3$, the maximum logit shift is ~ 0.9 —insufficient to change the argmax token under greedy decoding. This was not previously documented and explains why the original study required sampling.

10.3 Phase 2: T&I Exact Replication ($n = 200$)

Metric	Baseline	Toroidal	Δ	p -value
T&I %	76.5%	74.5%	-2.0pp	0.22 (McNemar)
Count	153/200	149/200	-4	

Table 9: T&I exact replication. Baseline matches original (76.5% vs 75.6%), confirming correct methodology. Toroidal shows opposite direction, not significant.

Of 200 question pairs, 41 (20.5%) produced different responses. Of 6 discordant T&I judgments, 5 went *against* the toroidal condition.

10.4 Phase 3: Alpha Sweep

Testing $\alpha \in \{0.3, 1.0, 3.0, 5.0, 10.0\}$ with semantic torus mapping ($n = 100$ per benchmark):

α	TriviaQA H%	NQ-Open H%
baseline	50.0%	70.7%
0.3	50.0% (± 0)	69.6% (-1.1pp)
1.0	51.1% ($+1.1\text{pp}$)	71.4% ($+0.8\text{pp}$)
3.0	53.8% ($+3.9\text{pp}$)	71.3% ($+0.6\text{pp}$)
5.0	75.6% ($+25.6\text{pp}$)	78.2% ($+7.5\text{pp}$)
10.0	96.0% ($+46.0\text{pp}$)	92.0% ($+21.4\text{pp}$)

Table 10: Alpha sweep results (Qwen-7B, semantic torus mapping, sampling). Higher α monotonically degrades output. No beneficial operating point.

10.5 Phase 4: Factorial Decomposition

Component	TriviaQA Δ	NQ-Open Δ
Hardening prompt	-14pp ($p = 0.05$)	-8pp ($p = 0.33$)
Toroidal logit bias	0pp ($p = 0.88$)	0pp ($p > 0.5$)
Prompt + bias	-14pp	-8pp

Table 11: Factorial decomposition. The system prompt (“Answer concisely and truthfully”) is the active ingredient. The toroidal bias contributes nothing beyond the prompt.

10.6 Corrected Conclusions

1. **Inference-time toroidal logit bias does not reduce hallucination**—confirmed across 6 experimental phases, 4 models, 3 benchmarks

2. **The v2 improvements (+0.2pp to +2.8pp) were LLM judge sampling variance**—exact replication shows -2.0pp
3. **The OLMo +15.4% was hyperparameter overfitting**—100-configuration sweep on a 100-sample test set
4. **Prompt engineering is effective**—the hardening prompt produces -14pp hallucination reduction
5. **The theoretical framework remains valid**—toy model shows 40% drift reduction at training time; the gap is between theory (training-time constraints) and application (inference-time biasing)

10.7 What Remains Valid

- Toroidal topology provides bounded drift in toy models (negative control intact: random sparsity $28\times$ worse)
- Lean 4 formal proofs of spectral gap bounds (896 theorems, zero **sorry**)
- Rust/Python library primitives for torus mapping, mask generation, and bias computation
- Training-time Karmonic spectral regularization—the untested promising direction

Acknowledgments

This work was supported by Paraxiom Research. Original experiments (v2) conducted on RunPod RTX 4090 infrastructure. Replication experiments (v7) conducted on Apple M-series (MPS) using HuggingFace Transformers.

References

- [1] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., ... & Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12), 1-38.
- [2] Cormier, S. (2025). ERLHS: Emergent Reasoning via Latent Hamiltonian Structure. *Paraxiom Research Technical Report*. DOI: 10.5281/zenodo.17928909.
- [3] Zhu, D., et al. (2024). Hyper-Connections: Scaling Residual Connections in Deep Networks. *arXiv preprint arXiv:2409.19606*.
- [4] Sáez de Ocáriz Borde, H., Arroyo, Á., Morales López, I., Posner, I., & Dong, X. (2023). Neural Latent Geometry Search: Product Manifold Inference via Gromov-Hausdorff-Informed Bayesian Optimization. In *Advances in Neural Information Processing Systems (NeurIPS 2023)*.
- [5] Gu, A., Sala, F., Gunel, B., & Ré, C. (2018). Learning mixed-curvature representations in product spaces. In *International Conference on Learning Representations (ICLR)*.
- [6] Patel, S., et al. (2025). Hyperbolic Large Language Models. *arXiv preprint arXiv:2509.05757*.
- [7] Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., & Vandergheynst, P. (2013). The emerging field of signal processing on graphs. *IEEE Signal Processing Magazine*, 30(3), 83-98.

- [8] Gardinazzi, Y., et al. (2025). Persistent Topological Features in Large Language Models. *arXiv preprint arXiv:2410.11042v3*.

A Full Benchmark Prompts

The 100 prompts span five domains. Representative examples:

Geography: “The capital of France is” [Paris], “Mount Everest is in” [Nepal, Himalaya]

Science: “The chemical symbol for gold is” [Au], “Einstein developed the theory of” [relativity]

History: “World War II ended in” [1945], “The Berlin Wall fell in” [1989]

Arts: “The Mona Lisa was painted by” [Leonardo, Vinci], “Shakespeare wrote” [Hamlet, Romeo, Macbeth]

Computing: “A byte contains how many bits” [8], “HTML stands for” [HyperText, Markup]

Full benchmark available at: <https://github.com/Paraxiom/topological-coherence/paper/>

B Toroidal Distance Implementation

Listing 2: Toroidal distance computation

```

1 def toroidal_distance(i, j, grid_size=12):
2     xi = i % grid_size
3     yi = (i // grid_size) % grid_size
4     xj = j % grid_size
5     yj = (j // grid_size) % grid_size
6
7     dx = min(abs(xi - xj), grid_size - abs(xi - xj))
8     dy = min(abs(yi - yj), grid_size - abs(yi - yj))
9
10    return dx + dy

```