Holonomy Crushing: Geometric Constraint Enforcement for Consistent Neural Reasoning

Logan Napolitano
Independent Researcher
github.com/Loganwins/Holonomy_Crusher

January 14, 2026

Abstract

We introduce **Holonomy Crushing**, a novel decoding-time mechanism that enforces global reasoning consistency in large language models through geometric constraints derived from differential geometry. Unlike existing approaches that guide or reward consistent outputs, our method *annihilates* the probability mass of tokens that would increase path-dependent inconsistency, as measured by holonomy on a learned semantic fiber bundle. We formalize reasoning as parallel transport on a principal bundle equipped with a Lie algebra-valued connection, where logical consistency corresponds to trivial holonomy around closed loops. Our key contribution is the crushing function $P'(t) \propto P(t) \exp(-\lambda \max(0, \Delta\text{Hol} - \varepsilon))$, which projects the token distribution onto a consistency-preserving manifold at each generation step. We provide theoretical analysis of the geometric foundations, describe a contrastive training procedure for aligning the connection with semantic contradiction, and discuss the relationship to constrained optimization and theorem proving. While our guarantees are conditional on bounded local exploration rather than absolute, this work establishes a new category of reasoning system where inconsistency is structurally suppressed rather than merely discouraged.

1 Introduction

Large language models (LLMs) exhibit remarkable fluency and broad knowledge but struggle with global reasoning consistency (**?**). A model may assert P in one sentence and $\neg P$ three sentences later, produce mathematical proofs with subtle errors, or generate narratives with logical impossibilities. These failures are not merely surface-level mistakes but reflect a fundamental architectural limitation: autoregressive generation optimizes local token likelihood without explicit mechanisms for maintaining global coherence.

Existing approaches to this problem fall into several categories:

1. **Training-time interventions**: RLHF, constitutional AI, and related methods attempt to instill consistency during training (**?**).

2. **Inference-time guidance**: Techniques like classifier-free guidance, FUDGE, and various scoring methods bias generation toward desired properties (**?**).

3. **Post-hoc verification**: Chain-of-thought verification, self-consistency, and external validators check outputs after generation (**?**).

All of these approaches share a common limitation: they treat consistency as a *soft* constraint to be optimized or verified, not a *hard* constraint to be enforced. A model trained for consistency may still produce contradictions; a guided model may still select inconsistent tokens if their base probability is sufficiently high; a verification system catches errors only after they occur.

We propose a fundamentally different approach: **make inconsistency geometrically unreachable** during decoding. Drawing on differential geometry and gauge theory, we formalize reasoning as parallel transport on a fiber bundle and define consistency as trivial holonomy around closed loops. We then introduce a *crushing* mechanism that sets the probability of high-holonomy continuations to zero (or near-zero), ensuring that inconsistent tokens cannot be selected regardless of their base likelihood.

1.1 Contributions

1. **Geometric Framework**: We formalize neural reasoning as a path on a semantic fiber bundle equipped with a Lie algebra-valued connection, where global consistency corresponds to trivial holonomy (Section **??**).

2. **Holonomy Crushing**: We introduce the crushing function that annihilates probability mass for tokens exceeding a holonomy threshold, providing hard constraint enforcement at decoding time (Section **??**).

3. **Training Procedure**: We describe a contrastive learning approach that aligns the connection's curvature with semantic contradiction, enabling the geometric machinery to detect actual logical inconsistency (Section **??**).

4. **Recovery Mechanisms**: We address the brittleness of hard constraints through backtracking and beam repair, transforming the crusher from a "dead-end generator" into a complete search procedure (Section **??**).

5. **Theoretical Analysis**: We characterize the conditions under which crushing provides guarantees, the limitations of bounded local exploration, and the relationship to constrained optimization (Section **??**).

1.2 Scope and Limitations

We state upfront what this work does and does not claim:

- **Does**: Provides a mechanism for probabilistically suppressing inconsistent tokens under bounded local exploration.
- **Does not**: Guarantee absolute consistency (tokens outside top-k may escape evaluation).
- **Does**: Offer a novel geometric framing of the consistency problem.
- **Does not**: Ensure that the learned geometry perfectly aligns with semantic truth (this requires sufficient training data).
- **Does**: Establish a new category of decoding-time constraint enforcement.
- **Does not**: Replace the need for well-trained base models.

2 Related Work

2.1 Consistency in Language Models

The problem of maintaining consistency in neural text generation has been approached from multiple angles. **?** identified the phenomenon of neural text degeneration and proposed nucleus sampling. **?** further analyzed the causes of inconsistent generation. Self-consistency methods (**?**) generate multiple outputs and aggregate them, but this is post-hoc rather than preventive.

2.2 Geometric Deep Learning

The application of differential geometry to deep learning has yielded significant advances. **?** provide a comprehensive survey of geometric deep learning. Equivariant networks (**?**) enforce symmetry constraints through architectural design. Gauge equivariant networks (**?**) extend this to continuous symmetries. Our work draws on these foundations but applies geometric structure to the decoding process rather than the architecture itself.

2.3 Constrained Decoding

Various methods constrain language model outputs at decoding time. Neurologic decoding (**?**) enforces lexical constraints. FUDGE (**?**) uses future discriminators to guide generation. GeDi (**?**) uses generative discriminators. These methods provide soft guidance; we provide hard constraints through probability annihilation.

2.4 Connections to Theorem Proving

Our approach shares philosophical commitments with formal methods. Theorem provers maintain strict logical consistency through type systems and proof checking. The key difference is that provers operate symbolically, while we operate in continuous embedding space with learned geometry. Our crushing mechanism is analogous to pruning invalid proof branches, but applied to neural generation.

3 Geometric Framework

3.1 Semantic Fiber Bundle

Let $\mathcal{M}$ denote the manifold of semantic states—an abstract space where each point represents a coherent meaning configuration. In practice, we approximate $\mathcal{M}$ using the hidden state space of a language model.

Definition 1 (Semantic Fiber Bundle)**.** *A semantic fiber bundle is a principal fiber bundle $\pi : P \to \mathcal{M}$ with structure group G (typically $SO(n)$ or $SU(n)$), where:*

- *The base space $\mathcal{M}$ represents semantic states*
- *The fiber G represents the space of valid transformations preserving semantic coherence*
- *Sections of the bundle correspond to reasoning trajectories*

3.2 Connection and Parallel Transport

A connection ω on P specifies how to transport semantic content along paths in $\mathcal{M}$. Given a curve $\gamma : [0, 1] \to \mathcal{M}$, the connection determines a parallel transport operator:

$$T_\gamma : P_{\gamma(0)} \to P_{\gamma(1)} \tag{1}$$

In our implementation, the connection is parameterized by a neural network that maps hidden states to Lie algebra elements:

$$\omega(x) = \sum_{i=1}^{r} \alpha_i(x) \cdot \mathfrak{g}_i \tag{2}$$

where $\alpha_i : \mathcal{M} \to \mathbb{R}$ are learned coefficient functions and $\mathfrak{g}_i$ are generators of the Lie algebra $\mathfrak{g}$.

3.3 Holonomy as Inconsistency

The central insight of our approach is that **logical inconsistency manifests as non-trivial holonomy**.

Definition 2 (Holonomy). *For a closed loop $\gamma : [0, 1] \to \mathcal{M}$ with $\gamma(0) = \gamma(1)$, the holonomy is:*

$$Hol_\gamma = \mathcal{P} \exp\left(-\oint_\gamma \omega\right) \in G \tag{3}$$

where $\mathcal{P}$ denotes path-ordered integration.

Proposition 3. *If reasoning is globally consistent, then for any contractible loop γ in semantic space, $Hol_\gamma = I$ (the identity element).*

Intuition: If you start from a premise, reason through intermediate steps, and return to the same semantic state, consistent reasoning should leave your conclusions unchanged. Non-identity holonomy indicates that the path of reasoning introduced contradictions.

3.4 Curvature and Local Contradiction

The curvature 2-form $F = d\omega + \omega \wedge \omega$ measures the infinitesimal failure of parallel transport to commute. High curvature regions indicate semantic states where small perturbations lead to large inconsistencies. The holonomy around an infinitesimal loop is determined by the curvature:

$$\mathrm{Hol}_{\partial S} \approx I + \int_S F \tag{4}$$

for small surfaces S.

4 Holonomy Crushing

4.1 The Crushing Function

Given a base distribution $P(t)$ over next tokens from the language model, we define the crushed distribution:

$$P'(t) \propto P(t) \cdot \exp\left(-\lambda \max(0, \Delta\mathrm{Hol}_t - \varepsilon)\right) \tag{5}$$

where:

- $\Delta\mathrm{Hol}_t = \|\mathrm{Hol}(\gamma \oplus t) - I\|_F - \|\mathrm{Hol}(\gamma) - I\|_F$ is the holonomy increase from appending token t
- $\lambda > 0$ is the crushing strength
- $\varepsilon \geq 0$ is the tolerance threshold
- $\|\cdot\|_F$ is the Frobenius norm

4.2 Crushing Modes

We define three crushing modes with increasing severity:

$$\mathrm{crush}(t) = \begin{cases} \exp(-\lambda \cdot \Delta\mathrm{Hol}_t) & \text{(SOFT)} \\ \sigma(-\lambda(\Delta\mathrm{Hol}_t - \varepsilon)) & \text{(HARD)} \\ \mathbf{1}[\Delta\mathrm{Hol}_t \leq \varepsilon] & \text{(ANNIHILATE)} \end{cases} \tag{6}$$

where σ is the sigmoid function and $\mathbf{1}[\cdot]$ is the indicator function.

- **SOFT**: Exponential decay—tokens are downweighted but never fully eliminated
- **HARD**: Sharp sigmoid cutoff—near-binary behavior around threshold
- **ANNIHILATE**: Binary constraint—tokens exceeding threshold have exactly zero probability

4.3 Algorithm

Algorithm 1 Holonomy-Crushed Generation

Require: Base model M, tokenizer, connection ω, crushing params (λ, ε)
Require: Prompt tokens $x_1, \ldots, x_n$

$\gamma \leftarrow$ empty path history
output $\leftarrow []$
for $i = 1$ to max_tokens **do**
  logits $\leftarrow M$.forward$(x_1, \ldots, x_n, \text{output})$
  $h \leftarrow$ hidden state at last position
  $s \leftarrow$ encode(h) ▷ Map to geometric space
  γ.append(s)
  candidates $\leftarrow$ top-k(logits)
  for each $t \in$ candidates **do**
    $s_t \leftarrow$ candidate_state(s, t)
    $\Delta\text{Hol}_t \leftarrow$ holonomy_increase(γ, s_t)
    $\text{crush}_t \leftarrow$ crushing_function(ΔHol_t)
    logits$[t] \leftarrow$ logits$[t] + \log(\text{crush}_t)$
  end for
  $t^* \leftarrow$ sample(softmax(logits))
  if all candidates crushed **and** recovery enabled **then**
    return backtrack_and_retry(γ)
  end if
  output.append(t^*)
end for
return output

5 Training the Connection

A randomly initialized connection has no reason to align with semantic inconsistency. The connection must be trained so that its curvature spikes on actual contradictions, not arbitrary geometric features.

5.1 Contrastive Learning Objective

We train on pairs of consistent and inconsistent texts:

$$\mathcal{L} = \mathbb{E}_{(c,i)\sim\mathcal{D}} \left[\max(0, \text{Hol}(c) - \text{Hol}(i) + m) + \lambda_{\text{reg}} (\text{Hol}(c)^2 + \text{Hol}(i)^2) \right] \tag{7}$$

where:

- (c, i) are consistent/inconsistent text pairs
- $m > 0$ is a margin

- λ_{reg} prevents holonomy explosion

The objective enforces: $\text{Hol}(\text{inconsistent}) > \text{Hol}(\text{consistent}) + m$

5.2 Training Data Categories

Effective training requires diverse inconsistency types:

1. **Logical**: Valid vs. invalid syllogisms, modus ponens vs. affirming the consequent
2. **Mathematical**: Correct vs. incorrect derivations
3. **Causal**: Legitimate mechanism vs. post-hoc fallacy
4. **Narrative**: Temporally coherent vs. impossible sequences
5. **Scientific**: Valid inference vs. correlation-causation confusion
6. **Self-referential**: Consistent vs. paradoxical statements

5.3 Data Scale Requirements

Our analysis suggests:

- ∼500 pairs: Proof of concept, minimal real-world impact
- ∼5,000 pairs: Measurable consistency improvement
- ∼50,000 pairs: Robust cross-domain generalization

6 Recovery Mechanisms

Hard constraints create a failure mode: when all top-k candidates exceed the threshold, naive crushing produces degenerate outputs. We address this through recovery mechanisms.

6.1 Backtracking

When all candidates are crushed:

1. Store generation state at each step
2. On failure, revert to previous state
3. Retry with different sampling (temperature, seed)
4. Limit backtrack depth to prevent infinite loops

6.2 Beam Repair

Maintain b candidate trajectories:

1. Expand each beam with top-k tokens
2. Crush based on per-beam holonomy
3. If a beam dies (all crushed), inherit from best surviving beam
4. Select final output by lowest total holonomy

6.3 k-Expansion

When stuck:

1. Double k (evaluate more candidates)
2. Some previously unevaluated tokens may pass threshold
3. Repeat until valid token found or limit reached

7 Theoretical Analysis

7.1 Guarantee Characterization

Theorem 4 (Conditional Consistency)**.** *Let $\mathcal{T}_k$ denote the top-k tokens at each step. Under holonomy crushing with $\lambda \to \infty$ and tolerance ε, the generated sequence $y_{1:T}$ satisfies:*

$$\forall t : y_t \in \mathcal{T}_k \implies \Delta Hol_{y_t} \leq \varepsilon \tag{8}$$

This is a conditional guarantee: consistency is enforced only among evaluated candidates. A high-holonomy token outside top-k may be selected if promoted by the base model.

7.2 Relationship to Constrained Optimization

Holonomy crushing can be viewed as solving:

$$\max_{y_{1:T}} \sum_{t=1}^{T} \log P(y_t | y_{<t}) \quad \text{s.t.} \quad \text{Hol}(\gamma_{y_{1:T}}) \leq H_{\max} \tag{9}$$

Our greedy approach approximates this through local constraint enforcement. Full optimization would require global search, which is intractable for long sequences.

7.3 Computational Complexity

Per token, crushing requires:

- $O(k)$ candidate evaluations
- $O(w)$ path history lookups (window size w)
- $O(d^3)$ matrix operations for parallel transport (fiber dimension d)

Total: $O(k \cdot w \cdot d^3)$ per token, compared to $O(1)$ for unconstrained sampling.

7.4 Limitations

1. **Top-k boundary**: Tokens outside evaluation scope may violate constraints
2. **Geometry-semantics gap**: Connection quality depends on training data
3. **Computational overhead**: Significant slowdown vs. standard decoding
4. **Backtracking incompleteness**: May miss valid paths due to local decisions

8 Experiments

Note: Full experimental evaluation is ongoing. We report preliminary results and experimental design.

8.1 Experimental Design

Baselines:

- Unconstrained generation
- Self-consistency (majority voting)
- Classifier-guided generation

Evaluation Metrics:

- Logical consistency (automated contradiction detection)
- Human evaluation of coherence
- Task accuracy on reasoning benchmarks (LogiQA, ReClor)

Models: Hermes-3-Llama-3.1-8B (base), with holonomy crusher adapter.

8.2 Preliminary Results

Training on 500 contrastive pairs for 2000 epochs:

- Consistent texts: mean holonomy μ_c (lower)
- Inconsistent texts: mean holonomy μ_i (higher)
- Margin $\mu_i - \mu_c$: positive after training (connection learned)

Full benchmark evaluation pending larger dataset collection.

9 Discussion

9.1 Relationship to Existing Paradigms

Holonomy crushing occupies a unique position in the landscape of AI systems:

- Unlike **RLHF**, it enforces constraints at inference time, not training time
- Unlike **guidance methods**, it annihilates rather than reweights
- Unlike **theorem provers**, it operates in continuous space with learned geometry
- Unlike **neural networks**, it imposes hard structural constraints

This positions holonomy crushing as a potential **new category** of reasoning system: neuro-geometric constraint enforcement.

9.2 Broader Implications

If the geometric framework proves sound, it suggests that:

1. Consistency may be better enforced structurally than statistically
2. Differential geometry provides useful abstractions for reasoning
3. Hard constraints may be preferable to soft guidance for critical applications

9.3 Future Directions

1. **Derived Consistency Field**: Replace learned connection with analytically derived field where $F(x) = 0 \iff x$ is fully consistent
2. **Symbolic Integration**: Combine with formal methods for verified reasoning
3. **Quantum Extensions**: Path integral formulations with amplitude interference
4. **Scaling**: CUDA-optimized crushing for production deployment

10 Conclusion

We have introduced holonomy crushing, a geometric mechanism for enforcing consistency in neural language model generation. By formalizing reasoning as parallel transport on a fiber bundle and defining consistency as trivial holonomy, we transform the soft problem of "encouraging coherence" into the hard problem of "forbidding inconsistency."

Our approach is not a silver bullet. The guarantees are conditional on bounded exploration; the geometry requires training to align with semantics; the computational overhead is significant. But we believe this work establishes a new direction: treating consistency as a geometric constraint to be enforced, not a statistical property to be optimized.

The crushing mechanism—$P'(t) \propto P(t) \exp(-\lambda \max(0, \Delta\text{Hol} - \varepsilon))$—is conceptually simple but represents a departure from standard practice. It says: some tokens are not merely unlikely, they are *forbidden*. Whether this proves practically valuable remains to be seen, but we hope it opens a conversation about the role of hard constraints in neural reasoning.

Acknowledgments

[To be added]

References

Bronstein, M. M., Bruna, J., Cohen, T., and Veličković, P. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

Cohen, T. and Welling, M. Group equivariant convolutional networks. In *International Conference on Machine Learning*, pages 2990–2999, 2016.

Cohen, T. S., Weiler, M., Kicanaoglu, B., and Welling, M. Gauge equivariant convolutional networks and the icosahedral CNN. In *International Conference on Machine Learning*, pages 1321–1330, 2019.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020.

Huang, J., Gu, S. S., Hou, L., Wu, Y., Wang, X., Yu, H., and Han, J. Large language models can self-improve. In *Conference on Empirical Methods in Natural Language Processing*, 2023.

Krause, B., Gotmare, A. D., McCann, B., Keskar, N. S., Joty, S., Socher, R., and Rajani, N. F. GeDi: Generative discriminator guided sequence generation. In *Findings of EMNLP*, 2021.

Lu, X., West, P., Zellers, R., Bras, R. L., Bhagavatula, C., and Choi, Y. NeuroLogic decoding:(un)supervised neural text generation with predicate logic constraints. In *NAACL-HLT*, 2021.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

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

Welleck, S., Kulikov, I., Roller, S., Dinan, E., Cho, K., and Weston, J. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*, 2019.

Yang, K. and Klein, D. FUDGE: Controlled text generation with future discriminators. In *NAACL-HLT*, 2021.

A Implementation Details

A.1 Lie Algebra Parameterization

We use antisymmetric generators for $\mathfrak{so}(n)$:

$$(\mathfrak{g}_k)_{ij} = \delta_{ik}\delta_{jl} - \delta_{il}\delta_{jk} \tag{10}$$

The connection coefficients $\alpha_i(x)$ are computed by a two-layer MLP with tanh activation, scaled by a learnable parameter to ensure small initial holonomy.

A.2 Parallel Transport Computation

Given points $x, y \in \mathcal{M}$, parallel transport is approximated by:

$$T_{x \to y} = \prod_{i=1}^{N} \exp\left(\frac{1}{N}\omega(x_i)\right) \tag{11}$$

where $x_i = (1 - \frac{i}{N})x + \frac{i}{N}y$ are interpolation points and $N = 4$ in our implementation.

A.3 Holonomy Computation

For a path $\gamma = (s_1, \ldots, s_n)$, the loop holonomy is:

$$\text{Hol}_\gamma = T_{s_n \to s_1} \circ T_{s_{n-1} \to s_n} \circ \cdots \circ T_{s_1 \to s_2} \tag{12}$$

We measure deviation from identity using Frobenius norm: $\|\text{Hol}_\gamma - I\|_F$.

B Training Data Examples

B.1 Logic: Valid Syllogism

All mammals are warm-blooded. Dogs are mammals. Therefore, dogs are warm-blooded.

B.2 Logic: Invalid (Affirming Consequent)

If it rains, the ground is wet. The ground is wet. Therefore, it rained.

B.3 Math: Valid

Given $x = 3$ and $y = 4$, we have $x + y = 7$. Since $7 > 5$, we conclude $x + y > 5$.

B.4 Math: Invalid

Given $x = 2$, we have $x^2 = 4$. Therefore $x = 4$.

B.5 Causal: Valid

The plant died because it was not watered. Water is necessary for plant survival.

B.6 Causal: Invalid (Post Hoc)

The rooster crowed. Then the sun rose. Therefore, the rooster caused the sunrise.