

From Explicit Holonomy to Latent Control Fields: Reducing the Cost of Consistency-Aware Transformers

Logan Napolitano

Independent Researcher

Lmnpainting@gmail.com

<https://github.com/Loganwins/HolonomyTransformer>

January 2026

Abstract

In prior work, we proposed the Holonomy Transformer (HoT), an architecture that enforces reasoning consistency through explicit geometric constraints derived from differential geometry. While conceptually principled, the original formulation incurs substantial computational overhead due to dense pairwise holonomy computation— $O(n^2 \cdot d_{\text{fiber}}^3)$ per layer. In this note, we present a counter-argument to the necessity of explicit path-level holonomy measurement and introduce an alternative formulation: the **Latent Control Field** approach. This reformulation preserves the core inductive bias while reducing computational complexity by an order of magnitude. The key insight is that consistency need not be measured after the fact; it can be anticipated through a learned, stateful control signal that accumulates predicted inconsistency and gates information flow accordingly. We reduce holonomy computation from $O(n^2)$ to $O(n)$, replace global waypoint discovery with event-triggered anchoring, and transform geometry from measurement into anticipation. This design note serves as both a self-critique of our prior work and a principled path toward scalable consistency-aware architectures.

1. Introduction

1.1 Context and Motivation

The Holonomy Transformer (Napolitano, 2026a) introduced a novel architectural paradigm: treating reasoning consistency as a geometric property enforced through parallel transport on fiber bundles. The core insight—that inconsistent reasoning paths exhibit non-trivial holonomy around closed loops—remains valid and, we believe, important.

However, a legitimate criticism of the original formulation concerns computational cost. The explicit computation of pairwise holonomy between all token positions introduces:

- **Quadratic complexity** in sequence length for holonomy measurement
- **Cubic complexity** in fiber dimension for matrix exponentials
- **Dense computation** where sparse signals would suffice

This note addresses these concerns directly. We do not retreat from the conceptual framework; we refine its implementation.

1.2 The Counter-Argument

The original HoT computes holonomy as a **measured quantity**:

$$H_{ij} = |T_{i \rightarrow j} \cdot T_{j \rightarrow i} - I|_F$$

This requires explicit parallel transport between all position pairs—expensive and, we now argue, unnecessary.

Our counter-argument is this: Consistency can be enforced without measuring it explicitly. Instead, we can:

1. **Predict** inconsistency before it manifests
2. **Accumulate** predicted inconsistency into a stateful control signal
3. **Gate** information flow based on accumulated state

This transforms holonomy from a **geometric measurement** into a **learned anticipation**—preserving the inductive bias while eliminating the computational bottleneck.

1.3 Relationship to Prior Work

This note builds directly on:

- **Napolitano (2026a)**: "The Holonomy Transformer" — original architecture
- **Napolitano (2026b)**: "Holonomy Crushing" — decode-time consistency enforcement
- **Napolitano (2026c)**: "Technical Report on Expected Behavior" — training dynamics

We do not supersede these works. We extend them with an efficiency-focused reformulation that makes scaling plausible.

2. The Cost Problem

2.1 Computational Analysis of Original HoT

In the original Holonomy Transformer, each attention layer computes:

Operation	Complexity
Standard attention	$O(n^2 \cdot d)$
Pairwise holonomy	$O(n^2 \cdot d_{\text{fiber}}^3)$
Fiber updates	$O(n \cdot d_{\text{fiber}}^2)$
Waypoint detection	$O(n^2)$

For typical values ($n = 2048$, $d = 512$, $d_{\text{fiber}} = 32$):

- Standard attention: ~ 2.1 billion operations
- Holonomy computation: ~ 137 billion operations

The holonomy computation dominates by $65\times$.

2.2 Why This Matters

At small scale ($n < 512$), this overhead is tolerable for research purposes. At production scale, it is prohibitive. If consistency-aware architectures are to have practical impact, the cost structure must change fundamentally.

2.3 The Wrong Solution

One might propose simply reducing d_{fiber} or computing holonomy sparsely. These approaches sacrifice the geometric foundation:

- Smaller d_{fiber} \rightarrow less expressive fiber structure
- Sparse holonomy \rightarrow missed inconsistencies

We seek a solution that preserves the inductive bias while changing the computational strategy entirely.

3. The Latent Control Field Approach

3.1 Core Insight

The key realization is that **we do not need to measure holonomy; we need to bias computation against inconsistency.**

This is analogous to the difference between:

- **Measuring** temperature at every point in a room (expensive)
- **Predicting** temperature gradients from local sensors (cheap)

Both achieve thermal awareness; only one scales.

3.2 From Measurement to Anticipation

Original formulation (measurement):

```
For each token pair (i, j):
    Compute transport  $T_{\{i \rightarrow j\}}$ 
    Compute transport  $T_{\{j \rightarrow i\}}$ 
    Measure holonomy  $H_{\{ij\}} = ||T \cdot T - I||$ 
    Use  $H_{\{ij\}}$  to penalize attention
```

New formulation (anticipation):

```
For each token t:
    Predict inconsistency  $\Delta h_t$  from local state
    Accumulate:  $h_t = \text{momentum} \cdot h_{\{t-1\}} + \Delta h_t$ 
    Gate attention/FFN based on  $h_t$ 
```

The first is $O(n^2)$. The second is $O(n)$.

3.3 The Holonomy Control Field

We introduce a **Holonomy Control Field**—a stateful signal that accumulates predicted inconsistency across the sequence.

Definition: The control field at position t is:

$$h_t = \alpha \cdot h_{t-1} + (1 - \alpha) \cdot \Delta h_t$$

where:

- $\alpha \in [0,1]$ is a momentum parameter
- $\Delta h_t = f_{\theta}(x_t, \phi_t)$ is the predicted holonomy increment
- x_t is the hidden state at position t
- ϕ_t is the fiber state at position t

Properties:

- **Temporal coherence:** Past inconsistency persists via momentum
- **Local computation:** Each Δh_t depends only on local state
- **Learned prediction:** The network learns to anticipate inconsistency

3.4 Predictive Holonomy Estimation

The holonomy predictor is a small neural network:

$$\Delta h_t = \text{softplus}(\text{MLP}([x_t; \phi_t]))$$

This network learns to predict: "If generation continues from this state, how much inconsistency will accumulate?"

Training signal: The predictor is trained end-to-end via the language modeling objective plus holonomy regularization. States that lead to contradictions develop high predicted holonomy; coherent states develop low predicted holonomy.

3.5 Gating Mechanism

The accumulated control field gates information flow:

$$g_t = \sigma(-\lambda \cdot h_t)$$

where λ controls crushing strength.

This gate is applied to:

- **Attention values:** $V_{\text{gated}} = g_t \cdot V$
- **FFN activations:** $\text{FFN}_{\text{gated}}(x) = g_t \cdot \text{FFN}(x)$

High accumulated holonomy \rightarrow closed gate \rightarrow suppressed information flow.

4. Event-Triggered Waypoint Detection

4.1 The Problem with Global Waypoints

The original HoT identifies waypoints by scanning all positions for stability. This requires $O(n^2)$ comparisons to establish relative stability.

4.2 Event-Triggered Alternative

We replace global scanning with local event detection:

A position becomes a waypoint when:

$$|\Delta h_t| < \epsilon \quad \text{AND} \quad \text{stability}(x_t) > \tau$$

where:

- ϵ is the holonomy threshold
- τ is the stability threshold
- $\text{stability}(x_t)$ is a learned stability predictor

Properties:

- **$O(1)$ per position:** No pairwise comparisons
- **Emergent anchors:** Waypoints crystallize where the model naturally stabilizes
- **Cached reuse:** Once detected, waypoints persist in a buffer

4.3 Waypoint Attention Bonus

Detected waypoints receive an attention bonus:

$$\text{Attention}_{ij}' = \text{Attention}_{ij} + \beta \cdot \mathbf{1}[\text{waypoint}_j]$$

This encourages information to route through stable positions without requiring global waypoint graphs.

5. Computational Analysis

5.1 Complexity Comparison

Operation	Original HoT	Control Field HoT
Standard attention	$O(n^2 \cdot d)$	$O(n^2 \cdot d)$
Holonomy/Control	$O(n^2 \cdot d_{\text{fiber}}^3)$	$O(n \cdot d_{\text{control}})$
Waypoint detection	$O(n^2)$	$O(n)$
Fiber updates	$O(n \cdot d_{\text{fiber}}^2)$	$O(n \cdot d_{\text{fiber}})$

5.2 Practical Overhead

For $n = 2048$, $d = 512$, $d_{\text{fiber}} = 16$, $d_{\text{control}} = 64$:

Model	Operations/Layer Relative to Baseline	
Standard Transformer	2.1B	$1.0 \times$
Original HoT	137B	$65 \times$
Control Field HoT	2.4B	$1.15 \times$

The control field approach reduces overhead from $65 \times$ to $1.15 \times$.

5.3 Memory Footprint

Model	Additional Memory
Original HoT	$O(n \cdot d_{\text{fiber}}^2)$ per layer
Control Field HoT	$O(n \cdot d_{\text{control}})$ per layer

With $d_{\text{fiber}} = 32$ and $d_{\text{control}} = 64$: memory reduction of $16\times$.

6. Theoretical Justification

6.1 Why Prediction Suffices

Claim: A learned predictor of holonomy can enforce consistency as effectively as explicit measurement, given sufficient training signal.

Argument: The holonomy of a path depends deterministically on the states along that path. A sufficiently expressive function approximator, trained on examples where holonomy correlates with contradiction, will learn to predict holonomy from local features.

The key insight is that inconsistency leaves **local signatures** before manifesting as **global contradictions**. The predictor learns these signatures.

6.2 Relationship to Energy-Based Models

The control field formulation has a natural interpretation as an energy-based model:

$$E(x_{1:n}) = \sum_t h_t = \sum_t \left[\alpha \cdot h_{t-1} + (1-\alpha) \cdot f_{\theta}(x_t) \right]$$

Low-energy sequences are consistent; high-energy sequences are contradictory. The model learns to generate low-energy continuations.

6.3 Control-Theoretic Interpretation

The gating mechanism implements a form of **feedback control**:

- **Sensor:** Holonomy predictor
- **Controller:** Accumulated control field
- **Actuator:** Attention/FFN gates

This connects our work to cybernetic and control-theoretic perspectives on cognition—a connection we find conceptually appropriate.

7. What This Preserves

We emphasize that the control field approach **preserves the core contributions** of the original HoT:

Original Contribution	Status
Geometric framing of consistency	✓ Preserved (as motivation)
Holonomy as inconsistency measure	✓ Preserved (as training signal)

Fiber bundle embeddings	✓ Preserved (simplified)
Waypoint crystallization	✓ Preserved (event-triggered)
Crushing mechanism	✓ Preserved (as gating)

What changes is the **computational strategy**, not the **conceptual foundation**.

8. What This Does Not Claim

We are explicit about limitations:

- **Not production-ready:** This is a design refinement, not a deployable system
- **Not guaranteed superior:** Empirical validation remains future work
- **Not a replacement:** The original formulation may be preferable for interpretability research
- **Not solved:** Fundamental questions about consistency remain open

This note reduces cost. It does not resolve all challenges.

9. Implementation Sketch

```
class HolonomyControlField(nn.Module):
    """Stateful holonomy accumulator."""

    def __init__(self, d_model, d_fiber, d_control, momentum=0.9):
        self.predictor = nn.Sequential(
            nn.Linear(d_model + d_fiber, d_control),
            nn.GELU(),
            nn.Linear(d_control, 1),
        )
        self.fiber_proj = nn.Linear(d_model, d_fiber)
        self.momentum = momentum

    def forward(self, hidden, prev_holonomy=None):
        # Project to fiber space
        fiber = self.fiber_proj(hidden)

        # Predict holonomy increment
        delta_h = F.softplus(self.predictor(torch.cat([hidden, fiber],

        # Accumulate with momentum
        if prev_holonomy is None:
            holonomy = delta_h
        else:
            holonomy = self.momentum * prev_holonomy + (1 - self.moment

        # Compute gate
        gate = torch.sigmoid(-10.0 * holonomy)

        return gate, holonomy
```

10. Conclusion

We have presented a counter-argument to our own prior work: explicit holonomy measurement is not necessary for consistency-aware computation. A learned, stateful control field can achieve the same inductive bias at a fraction of the cost.

This is not a retreat from the geometric perspective. It is a recognition that **geometry can inspire architecture without dominating computation**. The Holonomy Transformer's conceptual contribution—treating consistency as structural rather than statistical—stands. What changes is how we instantiate that concept efficiently.

The control field approach transforms holonomy from measurement to anticipation, from quadratic to linear, from exact to approximate. We believe this trade-off is correct for scaling, while preserving what matters: an architectural bias toward coherent reasoning.

One-sentence summary:

Consistency need not be measured; it can be anticipated—and anticipation scales.

References

- Napolitano, L. (2026a). The Holonomy Transformer: A Geometrically-Native Neural Architecture for Consistent Reasoning. Zenodo. <https://zenodo.org/records/18247585>
- Napolitano, L. (2026b). Holonomy Crushing: Geometric Constraint Enforcement for Consistent Neural Reasoning. GitHub. https://github.com/Loganwins/Holonomy_Crusher
- Napolitano, L. (2026c). Holonomy Transformer: Technical Report on Expected Behavior and Creativity Extension. Zenodo.
- Bronstein, M. M., et al. (2021). Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. arXiv:2104.13478.
-

Acknowledgments

This work represents a self-critique and refinement of prior ideas. We thank early readers whose skepticism about computational cost motivated this reformulation.

Code Availability

Implementation: <https://github.com/Loganwins/HolonomyTransformer>

License

"Geometry can inspire architecture without dominating computation."