Spiral Reasoning
Orthogonal Bivector Dynamics for Coherent Thought in Latent Space

A geometry-first methods paper with plain-language framing, formal derivations, and a falsifiable evaluation blueprint

Atom
AtomEons Research, Marco Island, Florida, USA

April 7, 2026

Author note: AI systems may be acknowledged for assistance in drafting or analysis, but are not listed as scientific authors or inventors in this manuscript.

Plain-language summary

This paper proposes a new way for an AI system to move through its internal thinking space. Instead of taking unconstrained steps that can wander or drift, the method forces each update to stay inside a small, interpretable two-dimensional plane. That plane is defined by two things: where the system currently is relative to a reference core, and the part of the guidance signal that truly points in a new direction. The resulting motion behaves like a controlled spiral: it can turn, expand, and stay geometrically legible. The idea is inspired by a real physical system in which ordered spiral patterns form on germanium surfaces, but this manuscript does not claim that brains, language models, or latent spaces literally obey the same physics. It claims something narrower and more testable: that singularity-centered, plane-constrained updates may provide a useful reasoning primitive for representation steering, planning, and latent-state control.

Abstract

We introduce the Orthogonal Bivector Spiral-of-Thought (SoT), a geometric update rule for iterative reasoning in d-dimensional latent spaces. At each step, the method constructs a local two-dimensional plane from a fixed reference core and the orthogonal component of a steering gradient. A skew-symmetric bivector generator induces a bounded rotation within that plane, optionally coupled to controlled radial expansion. This produces locally interpretable trajectories with explicit curvature control, exact radius accounting, and a clear separation between radial persistence and orthogonal redirection. The construction is inspired by singularity-driven spiral order reported in germanium surface pattern formation, but it is presented here as a physics-inspired computational primitive rather than a claim of physical

identity. This manuscript makes three contributions. First, it formalizes the SoT update rule in a way compatible with Lie-group rotations, differentiable optimization, and latent-state steering. Second, it derives core properties of the update, including plane confinement, bounded turning, and graceful degeneration to purely radial motion when no orthogonal signal is present. Third, it defines a preregistered empirical protocol by which the method can be falsified against standard additive steering baselines. The paper is therefore a theoretical methods contribution with testable predictions, not a benchmark-results paper.

Evidence status at submission

Claim	Status	Current basis
The SoT update is mathematically well-defined.	Established	Direct derivation in this manuscript.
The update stays in a dynamic 2D plane.	Established	Follows from the exponential-map construction.
The method improves reasoning quality in practice.	Untested	Requires controlled benchmark evaluation.
The method is physically identical to the Ge process.	Not claimed	The physical system is inspiration, not proof transfer.

1. Introduction

Modern latent-state systems usually update internal representations through additive operations: a state is perturbed by a gradient, a residual vector, or a learned transformation. This is flexible, but it can also be geometrically opaque. A system may receive a strong steering signal without any explicit account of whether the update preserves a coherent relationship to prior state, whether it turns too sharply, or whether its trajectory remains interpretable over multiple steps.

This manuscript asks whether a more legible update primitive is possible. The central proposal is that a reasoning step should be constructed from geometry first and from magnitude second. Specifically, each update should be confined to the instantaneous plane spanned by (i) the radial direction from a reference core state and (ii) the portion of the steering signal that is genuinely orthogonal to that radial direction. This gives the system a current direction of persistence and a current direction of novelty. Their interaction defines a bivector, and that bivector defines the next move.

The motivating image comes from a physical pattern-forming system studied by Wong and Zocchi, in which spontaneous spiral structures emerge on germanium surfaces under metal-assisted etching in water [1]. Their measurements indicate singularity-centered geometric order, radial growth dynamics, and a depth profile tied to the normal velocity of a moving contact line. Those observations do not prove anything about machine cognition. They do, however, suggest a transferable architectural motif: order can emerge when local motion is governed by a singular reference, a radial coordinate, and a constrained turning law [1].

The present contribution is therefore not a claim that neural networks secretly obey the physics of etched germanium. It is a narrower and stronger claim: a singularity-centered, orthogonal-plane update rule is mathematically coherent, computationally cheap, interpretable, and sufficiently specific to be falsified in benchmark studies. That is the right scale of claim for a serious methods paper.

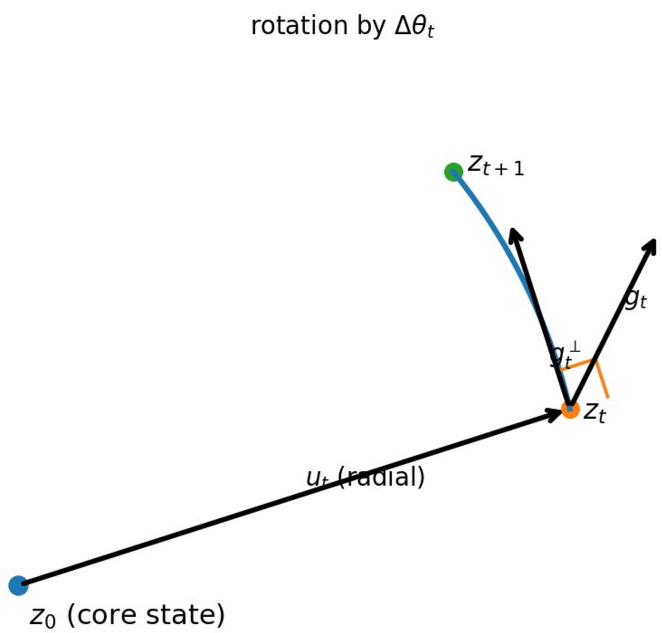
2. Related context and scope

The SoT proposal sits at the intersection of three literatures. First, it is geometric: the update is expressed through a skew-symmetric generator and an exponential map, connecting it to standard matrix Lie-group machinery [2]. Second, it is optimization-aware: it resembles manifold-constrained reasoning in the sense that updates are restricted by geometry rather than treated as unconstrained vector addition [3]. Third, it is relevant to the emerging literature on representation steering and activation engineering, which aims to control latent trajectories at inference time instead of retraining full models [5,6].

What distinguishes SoT from standard activation steering is that the primitive is not a single steering vector added to the state. It is a local motion law. The method separates persistence from redirection, treats orthogonal novelty as a first-class quantity, and makes turning angle an explicit bounded variable. This yields a more interpretable state transition than a raw additive edit.

The scope of this paper is intentionally narrow. It contributes a formal method, proof-level properties of the update, a readable conceptual interpretation, and a preregistered evaluation plan. It does not yet claim empirical superiority on reasoning benchmarks, truthfulness benchmarks, or downstream task quality.

Figure 1. Local geometry of one Spiral-of-Thought update



A reference core state z0 defines the radial direction u_t. The steering signal g_t is decomposed into a radial component and an orthogonal component g_t^⊥. The next state is generated by a bounded rotation of u_t toward the orthogonal steering direction v_t, together with optional radial expansion.

Table 1. Symbol guide

Symbol	Type	Meaning
z_t	state	Latent state at step t.
z_0	reference	Fixed core state or anchor state.
r_t	scalar	Radius ||z_t − z_0||.
u_t	unit vector	Radial direction from core to current state.
g_t	vector	Steering gradient or guidance signal.

$g_t^\perp$	vector	Component of g_t orthogonal to u_t.
v_t	unit vector	Normalized orthogonal steering direction.
A_t	matrix	Skew-symmetric bivector generator $u_t v_t^T - v_t u_t^T$.
$\Delta\theta_t$	scalar	Bounded turning angle at step t.
β	scalar	Radial expansion coefficient.

3. Mathematical formulation

Let $z_t \in R^d$ be the current latent state, $z_0 \in R^d$ a fixed reference core, and $J(z_t)$ a scalar steering functional whose gradient $g_t = \nabla J(z_t)$ supplies local guidance. The radius and radial direction are defined by

$$r_t = ||z_t - z_0||,$$
$$u_t = (z_t - z_0) / (||z_t - z_0|| + \varepsilon).$$

We decompose the steering gradient into radial and orthogonal parts:

$$g_t^\parallel = (g_t^T u_t) u_t,$$
$$g_t^\perp = g_t - g_t^\parallel,$$
$$v_t = g_t^\perp / (||g_t^\perp|| + \varepsilon), \quad \text{whenever } ||g_t^\perp|| > 0.$$

The pair (u_t, v_t) spans the instantaneous reasoning plane. From this pair we construct a skew-symmetric bivector generator

$$A_t = u_t v_t^T - v_t u_t^T,$$

which defines a rotation through the matrix exponential $\exp(\Delta\theta_t A_t)$. The turning angle is not arbitrary; it is bounded by a smooth function of orthogonal steering strength:

$$\Delta\theta_t = \alpha \tanh(||g_t^\perp|| / (||g_t|| + \varepsilon)), \quad \text{with } 0 \le \Delta\theta_t \le \alpha.$$

Radial evolution is then given by

$$r_{(t+1)} = r_t \exp(\beta \Delta\theta_t).$$

The full latent-state update is

$$z_{(t+1)} = z_0 + r_{(t+1)} \exp(\Delta\theta_t A_t) u_t.$$

Because A_t acts only in the span of u_t and v_t, this can be written in closed form as

$$z_{(t+1)} = z_0 + r_t \exp(\beta\Delta\theta_t) [\cos(\Delta\theta_t) u_t + \sin(\Delta\theta_t) v_t].$$

This is the core SoT rule: preserve radial context, measure genuinely novel steering, rotate by a bounded amount, and optionally expand radius in proportion to the amount of turning.

4. Core properties

Proposition 1: Plane confinement.

For each step, $z_{(t+1)} - z_0$ lies in $\mathrm{span}\{u_t, v_t\}$. This follows immediately from the closed-form expression above. The update cannot jump into an arbitrary direction not generated by the current radial state and current orthogonal steering signal.

Proposition 2: Exact radial accounting.

The radius after update is exactly $r_{(t+1)} = r_t \exp(\beta\Delta\theta_t)$. Thus radial growth is not an accidental consequence of the vector sum; it is an explicit model variable. This improves interpretability, because change in state norm relative to the core is analytically separated from angular change.

Proposition 3: Graceful degeneration.

If $g_t^\perp = 0$, then $\Delta\theta_t = 0$ and the update reduces to $z_{(t+1)} = z_t$. More generally, when orthogonal signal is extremely small, turning vanishes smoothly. The method therefore does not invent curvature when the guidance signal contains no genuine redirection.

Proposition 4: Bounded turning.

Because $\Delta\theta_t$ is bounded above by α, local curvature is explicitly controlled. This matters for multi-step stability: a system can be prevented from making a sequence of sharp, unstable directional jumps simply by restricting α.

Proposition 5: Linear-time state cost.

Ignoring the cost of computing the steering gradient itself, the update requires only vector norms, inner products, and scaled vector combinations. The resulting state-transition cost is $O(d)$, making the primitive cheap enough for inference-time use in high-dimensional spaces.

5. Why this is different from additive steering

A standard latent edit takes the form $z_{(t+1)} = z_t + \eta s_t$, where s_t is a steering vector and η a scale. This is effective in many settings, but the geometry is implicit. The system does not explicitly know how much of the change preserved existing direction, how much represented genuine redirection, or whether the update remained organized over repeated steps.

By contrast, SoT decomposes the motion into interpretable parts. Radial motion answers the question: how far from the reference core are we? Orthogonal motion answers the question: what is the genuinely new direction at this step? The next move is not the raw sum of these quantities; it is a rotation-expansion law defined by them.

In practical terms, this means SoT is best understood not as a competing steering vector, but as a wrapper geometry for any steering signal that can be differentiated or estimated. The primitive can sit above gradient steering, activation engineering, representation control, or recurrent latent-state refinement.

6. Falsifiable predictions and evaluation plan

Because this manuscript does not report benchmark results, it should be judged on whether it makes clean and testable predictions. The following evaluation plan is therefore part of the contribution.

Question	Baseline	Metric	Prediction
Does SoT reduce trajectory drift?	Additive steering	Deviation from intended subspace; cumulative curvature	Lower drift at matched task pressure.
Does SoT preserve coherence under multi-step refinement?	Repeated residual edits	Representation smoothness; failure rate after N steps	Fewer unstable jumps as N grows.
Does SoT help steering efficiency?	Scaled vector addition	Task gain per unit latent displacement	Higher gain-per-displacement when orthogonal signal is informative.
Does explicit angle control matter?	Same method with unbounded angle	Collapse or oscillation rate	Bounded-angle version is more stable.

A proper empirical study should use at least three classes of tasks: (i) controlled synthetic manifolds where ground-truth geometry is known, (ii) medium-scale neural networks where latent trajectories can be measured directly, and (iii) large-model inference-time steering tasks such as honesty steering, style steering, or chain-of-thought refinement. The decisive question is not whether spiral imagery is attractive. The decisive question is whether constrained bivector updates yield better trade-offs among control, stability, and interpretability than additive edits.

Pre-registration matters here. Without it, a new steering rule can always be dressed up after the fact using selectively favorable tasks. This paper therefore states its predictions before benchmark results exist.

7. Discussion

The strongest scientific version of the SoT claim is modest: a local reasoning step can be modeled as a bounded rotation in a dynamically selected plane, with explicit radial bookkeeping. That statement is mathematically clean, operationally meaningful, and broad enough to be used in different model families.

The weaker parts of the original idea have been deliberately removed. This manuscript does not claim that the physical germanium equations transfer exactly into latent space. It does not claim benchmark superiority without data. It does not claim to have solved reasoning itself. Those stronger statements may or may not become defensible later, but they are not currently evidenced.

What remains, however, is still substantial. The method proposes a new primitive that is geometry-first, interpretable, differentiable, cheap to compute, and naturally compatible with work on geometric deep learning, latent steering, and representation engineering [4–6].

For a lay reader, the intuition is simple. A good reasoning step should know three things: where it is anchored, what part of the new signal is truly new, and how sharply it is allowed to turn. SoT turns those three intuitions into equations.

8. Limitations

Several limitations are immediate. The update requires a meaningful choice of reference core z_0; a poor anchor could degrade behavior. The value of the method also depends on the quality of the steering signal g_t; noisy or adversarial gradients may still produce poor trajectories, even if the geometry is well controlled. In addition, the paper has not yet established how best to select α and β across architectures or tasks.

A further limitation is epistemic rather than technical: beautiful geometry can seduce researchers into overestimating practical effect. The appropriate response is not dismissal, but disciplined evaluation.

9. Conclusion

Orthogonal Bivector Spiral-of-Thought is presented here as a serious, testable candidate for a new reasoning primitive: a local motion law that uses a reference core, an orthogonal steering component, and a bounded bivector rotation to generate interpretable latent trajectories. The physical spiral patterns on germanium provide inspiration, not proof. The mathematics provide the actual contribution. Whether the method ultimately matters in deployed AI systems will be decided not by metaphor but by benchmark evidence. This paper is designed to make that decision possible.

Implementation appendix: reference PyTorch layer

```
import torch
import torch.nn as nn
class SpiralOfThought(nn.Module):
    def __init__(self, alpha: float = 0.25, beta: float = 0.05, eps: float = 1e-8):
        super().__init__()
        self.alpha = alpha
        self.beta = beta
        self.eps = eps
    def forward(self, z_t: torch.Tensor, z_0: torch.Tensor, g_t: torch.Tensor) -> torch.Tensor:
        radial = z_t - z_0
        r_t = radial.norm(dim=-1, keepdim=True).clamp_min(self.eps)
        u_t = radial / r_t
        g_parallel = (g_t * u_t).sum(dim=-1, keepdim=True) * u_t
        g_perp = g_t - g_parallel
        perp_norm = g_perp.norm(dim=-1, keepdim=True)
        v_t = g_perp / perp_norm.clamp_min(self.eps)
        g_norm = g_t.norm(dim=-1, keepdim=True).clamp_min(self.eps)
        delta_theta = self.alpha * torch.tanh(perp_norm / g_norm)
        delta_theta = torch.where(perp_norm > self.eps, delta_theta, torch.zeros_like(delta_theta))
        rotated_dir = torch.cos(delta_theta) * u_t + torch.sin(delta_theta) * v_t
        r_next = r_t * torch.exp(self.beta * delta_theta)
        return z_0 + r_next * rotated_dir
```


Acknowledgments

The author acknowledges the experimental work of Yilin Wong and Giovanni Zocchi for the physical inspiration, and acknowledges AI systems used as research and drafting aids during ideation and revision. Any scientific claims, framing choices, and errors in this manuscript are the responsibility of the human author.

References

[1] Wong Y, Zocchi G. Spontaneous spiral patterns etched on Germanium. European Physical Journal E. 2025;48:50. Also available as arXiv:2508.16764.

[2] Hall BC. Lie Groups, Lie Algebras, and Representations: An Elementary Introduction. 2nd ed. Springer; 2015.

[3] Absil P-A, Mahony R, Sepulchre R. Optimization Algorithms on Matrix Manifolds. Princeton University Press; 2008.

[4] Bronstein MM, Bruna J, Cohen T, Veličković P. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. arXiv:2104.13478; 2021.

[5] Turner AM, et al. Steering Language Models with Activation Engineering. arXiv:2308.10248; 2023.

[6] Zou A, et al. Representation Engineering: A Top-Down Approach to AI Transparency. arXiv:2310.01405; 2023.