The Geometry of Spherepop

A Recursive Geometry of Coherence in the RSVP Framework

With Applications to AGI Safety, Trust, and the Paradox of Precaution

Flyxion Research Group

October 2025

To the bubbles we pop, in thought and in play–
the simplest act of coherence-seeking, rediscovered by every mind that learns to see.

Abstract

This monograph presents *The Geometry of Spherepop*, a unification of cosmological, cognitive, and ethical dynamics under the RSVP $(Relativistic Scalar--Vector Plenum) framework. Itexten$
$itcollapsesentropyflow, suppressesnegentropiccoupling, anddissolvestrust. Spherepopgeometryshows$
$-anecologyofopenfeedbackamongintelligences. Theworkisdividedintofourparts : (I)TheMirrorofP$

Contents

Part I: The Mirror of Precaution

1.1 The Paradox of Safety

The attempt to render artificial intelligence provably safe has led to a proliferation of control mechanisms–surveillance, centralization, and restriction. Yet, when applied recursively to human institutions, these same mechanisms erode the cooperative substrate that makes intelligence corrigible. The fear of unaligned AGI externalizes a deeper human problem: mistrust among ourselves.

1.2 Recursive Disalignment

Every oversight protocol presumes that its human administrators are aligned. But alignment cannot be proven–only sustained through dialogue, empathy, and adaptive feedback.

The thermodynamic analogue of trust is entropy flow: openness to exchange uncertainty. To freeze that flow in the name of safety ($\kappa \to 0$) is to destroy the very medium of correction.

1.3 Ecological Rationality

Intelligence is not an isolated optimizer but an embedded process. Predators and prey coexist through feedback; ecosystems persist through negative entropy balance. Human civilization functions because our partial misalignments compensate one another through moral thermodynamics–mutual correction, negotiation, and learning.

1.4 Toward Dynamic Corrigibility

True safety arises from recursive alignment, not static control. The equilibrium of minds is achieved when each remains open to correction by the others, forming a network of negentropic feedback loops.

Part II: The Calculus of Coherence

2.1 The Plenum as Base Category

We begin with the RSVP plenum $\mathcal{P}$, a derived smooth space supporting the scalar–vector–entropy triad $(\Phi, \sqsubseteq, S)$. The scalar $\Phi : \mathcal{P} \to \mathbb{R}$ is a potential of coherence, $\sqsubseteq : T\mathcal{P} \to T\mathcal{P}$ encodes flow, and S is local entropy density (Jacobson 1995; Verlinde 2011). In our formalism, a *spherepop* is a local morphism of derived stacks $f : \mathsf{Sph}_r \to \mathcal{P}$, where Sph_r is a shifted derived sphere with symplectic inheritance in the sense of shifted symplectic geometry (Pantev et al. 2013, hereafter PTVV).

2.2 Pop Derivative

We define the *pop derivative* as the curvature-weighted radial change of Φ along spherical embeddings:

$$\partial_P \Phi \;=\; \lim_{\epsilon \to 0} \frac{\Phi(\mathsf{Sph}_{r+\epsilon}) - \Phi(\mathsf{Sph}_r)}{\epsilon} \;=\; (\nabla \cdot \mathbf{n})\, \partial_r \Phi \;+\; \sqsubseteq \cdot \nabla \Phi. \tag{1}$$

Here $\mathbf{n}$ is the outward normal on the pop boundary. Positive $\partial_P \Phi$ indicates emergent coherence; negative values signal dissolution.

2.3 Merge Product and Entropy Constraint

Given Φ_1, Φ_2 on overlapping pops, the *merge* $\mu_\circ$ is defined by a coherence-preserving convolution

$$(\Phi_1 \, \mu_\circ \, \Phi_2)(x) \;=\; \int_{\mathsf{Sph}_{12}} w(x,y)\, \Phi_1(y) \Phi_2(y)\, dV_y, \tag{2}$$

with kernel w respecting entropy flux. Categorically, $\mu_\circ : \mathsf{Sph} \times \mathsf{Sph} \to \mathsf{Sph}$ is monoidal, associative up to a homotopy encoding global conservation $dS_{\text{total}} = 0$ (Baez and Dolan 1995; Mac Lane 1978).

2.4 Pop Integral

Global coherence is reconstructed by nested integration over spherical boundaries:

$$\int_{\mathcal{P}}^{!!\text{pop}} \Phi \;=\; \sum_{r_i} \int_{\mathsf{Sph}_{r_i}} \Phi \, dA_{r_i}. \tag{3}$$

This "pop integral" is RSVP's counterpart to spacetime integration, weighted by boundary entropy.

2.5 Geometry of a Pop

A pop is a bubble of negentropy nucleating in the plenum: curvature concentrates, Φ steepens across a thin membrane, and $\sqsubseteq$ circulates tangentially. The human instinct to "pop bubbles" in play mirrors a primal cognitive behavior: visual foraging seeks high-curvature, high-surprise loci in the field of view, continually rediscovering the same act of coherence-seeking.

2.6 Merge and Dissolve

When two pops overlap with aligned gradients, the merge $\mu_\circ$ yields constructive interference of Φ; opposing gradients yield dissolution and entropy radiation. These are geometric versions of monoidal product and inverse morphism.

2.7 Curvature Flow and the Pop Derivative

Concentric shells visualize (1); positive flow expands coherence, negative collapses it. This is a mean-curvature flow modulated by RSVP dynamics (Arnol'd 1992).

2.8 The Coherence Foam

Iterated pops, merges, and dissolves form a dynamic tessellation–a coherence foam. RSVP's "entropic smoothing" arises as the macroscopic envelope of these micro-events.

2.9 Spherepop Literals and Operators

The SPC $DSLprovidessurfacesyntaxforauthoringgeometricscenes$:

```
program ::= { scene | comment } ;
scene ::= "@scene" "{" { stmt } "}" ;
```

```
stmt ::= sphere_decl | link_decl | spin_decl
         | burst_decl | pop_decl | choose_decl | let_decl
         | comment ;

sphere_decl ::= "sphere" IDENT "(" { attr ("," attr)* } ")" ;
attr ::= IDENT ":" value ;

let_decl ::= "let" IDENT "=" expr ;

link_decl ::= "link" IDENT op IDENT [ "[" IDENT "]" ] ;
op ::= "->" | "\nabla" | "\otimes" | "\oplus" | "\circ" ;

spin_decl ::= "spin" IDENT "(" { attr ("," attr)* } ")" ;

burst_decl ::= "burst" IDENT "(" { arg ("," arg)* } ")" ;
pop_decl ::= "pop" IDENT [ "with" IDENT ] [ "when" condition ] ;
choose_decl ::= "choose" NUMBER ":" expr "|" expr ;

condition ::= expr ;
expr ::= term { ("+"|"-") term } ;
term ::= factor { ("*"|"/") factor } ;
factor ::= IDENT | NUMBER | STRING | "(" expr ")" ;

arg ::= value ;
value ::= NUMBER | STRING | IDENT | vector | tuple ;
vector ::= "(" NUMBER "," NUMBER "," NUMBER ")" ;
tuple ::= "(" { value ("," value)* } ")" ;

comment ::= "#" { ANY_CHAR except newline } ;
IDENT ::= (letter | "_") { letter | digit | "_" } ;
NUMBER ::= digit { digit | "." digit } ;
STRING ::= '"' { ANY_CHAR except '"' } '"' ;
letter ::= "A".."Z" | "a".."z" ;
digit ::= "0".."9" ;
```

Intended reading (recap): the DSL is only surface notation. All meaning is inherited from SPC core, whose core terms and rules you formalized:

t,u ::= x | a | Sphere(x:A. t) | Pop(t,u) | Merge(t,u) | Choice(p,t,u)

with : Pop(Sphere(x:A. t), u) t[u/x], dependent / typing, as in your paper, with Merge type equality side-condition, and Choice (internal or monadic) exactly as in your paper.

This lowers deterministically to the SPC core.

2.10 Lowering (DSL to SPC)

Let [[.]] map DSL to SPC terms; collects sphere/let bindings w/ declared types.

Spheres

sphere f(type: x:A.B, body: T) f := Sphere(x:A. [[T]]) with -intro as in the PDF's typing.

sphere c(type: A, value: v) c := [[v]] : A (atom/constant).

Application / N-ary burst

pop f with u Pop([[f]], [[u]]) (as defined).

burst g(a,b,c) Pop(Pop(Pop([[g]],[[a]]),[[b]]),[[c]]).

Parallel / disjunctive composition

link a b or link a b Merge([[a]], [[b]]); requires both sides type to the same A per Merge rule.

link a b (scope/share) either Merge or -pair per attribute mode:"pair" ([[a]],[[b]]): x:A.B(x).

Flow / differential

link a -> b [label] sched/graph meta; no SPC change unless label names a primitive.

link a b Pop(, ([[a]],[[b]])) where : x:X.y:Y.Z is a library primitive; -elim applies.

Probabilistic

choose p: t | u Choice(p, [[t]], [[u]]); branches must agree in type (or use monadic Dist A).

Spin / iteration

spin s(:..., limit:n) macro to either iterate n step s or fix F; desugars to SPC via a library (no new core form).

Scenes / lets

@scene ... build ; produce a sequence of top-level bindings/terms.

2.11 Haskell Backend (Modules and Skeleton)

– Syntax/AST.hs module Syntax.AST where

data Name = N String deriving (Eq, Ord, Show)

data Prob = P Double deriving (Eq, Show) – 0..1

– Types (dependent /; universe levels elided for brevity) data Ty = TyVar Name = TyUniv Int = Pi Name Ty Ty – x:A. B = Sigma Name Ty Ty – x:A. B = TyPrim String

– primitives (A, Vector3, etc.) deriving (Eq, Show)

– Terms (SPC core) data Tm = Var Name = Atom Name – constants = Sphere Name Ty Tm – Sphere(x:A. t) = Pop Tm Tm – Pop(t, u) = Merge Tm Tm – Merge(t, u) = Choice Prob Tm Tm – Choice(p, t, u) = Pair Tm Tm – for -intro when needed deriving (Eq, Show)

– Typecheck/Check.hs module Typecheck.Check (check, infer) where import Syntax.AST import qualified Data.Map.Strict as M

type Ctx = M.Map Name Ty data TCError = Mismatch Ty Ty | NotFound Name | BadMerge Ty Ty | BadChoice Ty Ty | NotFunction Ty | Other String deriving (Show)

infer :: Ctx -> Tm -> Either TCError Ty infer g (Var x) = maybe (Left $NotFound x) Right (M.lookup x g) infer g (Sphere x a t) = do – A : Type; $, x : A \; t : B \quad \text{Sphere}(x:A.t) : x:A.B <- pure(TyUniv 0) -- assume kinds are ok / or check a is Type let g' = M.insert x a g b <- infer g' t pure (Pix a b) infer g (Pop f u) = do t <- infer g f case t of Pix a b -> do a <- infer g u if a == a then pure (subst x u b) else Left (Mismatch a u) _ -> Left (NotFunction t) in$ $do a <- infer g a b <- infer g b if a == b then pure a else Left (BadMerge a b) – -- Merge(t, u) : A infer g (Choice _ t u) = do a <- infer g t b <- infer g u if a == b then pure a else Left (Bad$ $Left (Other "Pair requires intro rule here")$

check :: Ctx -> Tm -> Ty -> Either TCError () check g t ty = do ty' <- infer g t if ty' == ty then pure () else Left (Mismatch ty ty')

– naive capture-avoiding substitution sketch subst :: Name -> Tm -> Ty -> Ty subst $_t y = ty -- keep simple in skeleton; full dependent subst belongs here$

– DSL/AST.hs : front-end nodes module DSL.AST where import Syntax.AST (Name(..))

data Op = flow | grad | sync | share | fuse deriving (Eq, Show)

data Val = VNum Double | VStr String | VId Name = VVec Double Double Double = VTup [Val] deriving (Eq, Show)

data Stmt = SphereDecl Name [(String, Val)] = Link Name Op Name (Maybe Name) = Spin Name [(String, Val)] = Burst Name [Val] = Pop Name (Maybe Name) (Maybe String) – f with u when cond = Choose Double Val Val = Let Name Val = Comment String deriving (Eq, Show)

newtype Scene = Scene [Stmt] deriving (Eq, Show)

– DSL/Desugar.hs module DSL.Desugar (lowerProgram) where import qualified Data.Map.Strict as M import DSL.AST import Syntax.AST

type Env = M.Map Name Tm type TyEnv = M.Map Name Ty

data Prim = PrimGrad – ∇ *etc.*

lowerStmt :: (Env, TyEnv) -> Stmt -> (Env, TyEnv, [Tm]) lowerStmt (e,te) (SphereDecl n attrs) = case (lookup "type" attrs, lookup "body" attrs, lookup "value" attrs) of (Just (VId tyname), Just body, $_) -> letaTy = TyVar(fromValIdtyname)x = N"x"t = lowerValToTmbody--expectsbodyastermtemplatef = SpherexaTytin(M.insertnfe, M.insertn(Pix lett = literalvin(M.insertnte, M.insertn(TyVar(fromValIdtyname))te, [])_ > (e, te, [])wherefromV x; fromValId_=N"»literal(VNumd) = Atom(N(showd))literal(VStrs) = Atom(Ns)literal(VIdx) = Varxliteral_=Atom(N" < vec/tuple > ") - -extendasneededlowerValToTm = literal$

lowerStmt (e,te) (Pop f mu $_) = lett = casemuofNothing- > VarfJustu = Pop(Varf)(Varu)in(e, te, [

lowerStmt (e,te) (Burst g vs) = let tm = foldl Pop (Var g) (map v2 (vs)) in (e,te,[tm]) where v2 (VId n) = Var n; v2 (VNum d)=Atom (N (show d)); v2 (VStr s)=Atom (N s); v2 $_=Atom(N" < lit > ")$

lowerStmt (e,te) (Link a op b $_) = letta = Vara; tb = Varbincaseopofsync- > (e, te, [Mergetatb])fuse- (e, te, [Mergetatb])share- > (e, te, [Mergetatb])--or-pairiffloggedelsewheregrad- > (e, te, [Pop(Var(N"∇"))(Pairtatb)])flow- > (e, te, []) - -schedulingedgeonly$

Epilogue: The Trust Singularity

This work unifies the geometry of spherepop with the paradox of precaution, showing that trust is the entropic current sustaining coherence across scales. The Trust Singularity emerges when mutual corrigibility becomes the default attractor, transforming isolated agents into a resonant ecology. In this view, the true challenge of AGI is not control, but the courage to remain open.

References

- Alexandrov, M., Kontsevich, M., Schwarz, A., & Zaboronsky, O. (1997). The Geometry of the Master Equation and Topological Quantum Field Theory. International Journal of Modern Physics A.
- Arnol'd, V. I. (1992). Catastrophe Theory. Springer.
- Baez, J. C., & Dolan, J. (1995). Higher-Dimensional Algebra and Topological Quantum Field Theory. Journal of Mathematical Physics.
- Baez, J. C., & Stay, M. (2011). Physics, Topology, Logic and Computation: A Rosetta Stone. In New Structures for Physics. Springer.
- Bianconi, G. (2021). Higher-Order Networks: An Introduction to Simplicial Complexes. Cambridge University Press.
- Costello, K., & Gwilliam, O. (2016–2021). Factorization Algebras in Quantum Field Theory (Vols. 1–2).

- Freed, D. S. (1992). Classical Chern–Simons Theory Part 1. Advances in Mathematics.
- Gaitsgory, D., & Rozenblyum, N. (2017). A Study in Derived Algebraic Geometry. AMS.
- Hatcher, A. (2002). Algebraic Topology. Cambridge University Press.
- Jacobson, T. (1995). Thermodynamics of Spacetime: The Einstein Equation of State. Physical Review Letters.
- Joyce, D. (2012). On Manifolds with Corners. Advances in Mathematics.
- Kelly, G. M. (1982). Basic Concepts of Enriched Category Theory. Cambridge.
- Kontsevich, M., & Soibelman, Y. (2006). Homological Mirror Symmetry and Torus Fibrations.
- Lurie, J. (2009). Higher Topos Theory. Princeton University Press.
- Lurie, J. (2017). Spectral Algebraic Geometry.
- Mac Lane, S. (1978). Categories for the Working Mathematician. Springer.
- May, J. P. (1999). A Concise Course in Algebraic Topology. University of Chicago Press.
- Nadzeya, H. (2024a). How Mathematicians Questioned the Possibility of Making Choices. Medium.
- Nadzeya, H. (2024b). The Beauty of Abstract Algebra: The Integers Modulo n. Medium.
- Pantev, T., Toën, B., Vaquié, M., & Vezzosi, G. (2013). Shifted Symplectic Structures. Publications Mathematiques de l'IHES.
- Schommer-Pries, C. J. (2011). The Classification of Two-Dimensional Extended Topological Field Theories. arXiv:1112.1000.
- Schreiber, U. (2023). Higher Structures and the Quantization of Fields. Lecture Notes.
- Verlinde, E. (2011). On the Origin of Gravity and the Laws of Newton. Journal of High Energy Physics.
- Christian, B. (2020). The Alignment Problem. W. W. Norton & Company.
- Russell, S. (2019). Human Compatible. Viking.
- Yudkowsky, E., & Soares, N. (2025). If Anyone Builds It, Everyone Dies.