ChatGPT

Geometric–Informational Multi-Agent System Framework

We model the **foundational frame** as a time-dependent differentiable manifold $(M,g(t))$. Here M is a smooth spatial domain (possibly with boundary or nontrivial topology), and $g(t)$ is a Riemannian metric (or pseudo-Riemannian if time is included) that endows M with geometry (distances, angles, curvature). The field state includes geometric variables such as the metric $g_{ab}(x,t)$, curvature tensors $R_{abcd}(x,t)$, volume/area forms, and any additional scalar or tensor fields ${\phi_k(x,t)}$ defined on M. These fields can represent local resources, potentials, or “add-variables” that exist in localized regions. For example, one may have a diffusion field $\phi(x,t)$ on M that evolves by a PDE (e.g. $\partial_t \phi = D\,\Delta_g \phi + f(\phi,\rho_{\rm agents})$), where Δ_g is the Laplace–Beltrami operator of (M,g) and $\rho_{\rm agents}(x,t)$ is the agent density. The manifold’s curvature and topology (holes, handles, etc.) can influence agent motion (via geodesic distances) and can evolve in time. One canonical evolution for the metric is the **Ricci flow**, defined by $$\partial_t g_{ab} = -2\,\mathrm{Ric}*{ab}(g)\,, $$ where $ \mathrm{Ric}*(g)$ is the Ricci tensor [1]. This nonlinear PDE “smooths” curvature on M (analogous to a heat equation on geometry) and can be augmented by source terms depending on agent distribution. In summary, we treat the foundational field as a manifold $(M,g(t),{\phi})$ whose state evolves by coupled PDEs (e.g. geometric flows and diffusion equations), influenced by agent presence and intrinsic geometry.

- **Manifold and Geometry:** Let M be a smooth manifold (compact or unbounded) of dimension d. We equip M with a time-dependent metric $g(t)$ whose curvature $\mathrm{Rm}(x,t)$, Ricci tensor $\mathrm{Ric}(x,t)$ and scalar curvature $R(x,t)$ encode “holes” and “indents.” Geometric invariants (e.g. sectional curvatures) govern how agents perceive distance and area. Topologically nontrivial features (handles, holes) may be represented by nonzero homology groups or noncontractible loops.
- **Field Variables:** Beyond geometry, local fields $\phi_k(x,t)$ (scalar or tensor fields on M) model resource distributions or environmental features. These fields satisfy PDEs on M, such as reaction–diffusion or wave equations, possibly driven by agent density $\rho(x,t) = \sum_i \delta(x-x_i(t))$. For instance, the metric evolution could obey a modified Ricci flow $\partial_t g = -2\,\mathrm{Ric}(g) + \alpha\,T_{\rm agents}$, analogous to Einstein’s equation where $T_{\rm agents}$ depends on agent mass-energy. Likewise, $\phi(x,t)$ may diffuse (with Laplacian Δ_g) and couple to agents through source/sink terms.

Citations: Notions from differential geometry extend naturally to metric spaces and networks [2]. In particular, Ricci flow on (M,g) is a well-known geometric PDE $\partial_t g=-2\,\mathrm{Ric}(g)$ [1], and graph analogs of curvature (Ollivier–Ricci, Forman curvature) have been developed in network theory [2]. This justifies treating our foundational frame via Riemannian geometry concepts.

Agent Structure and Dynamics

The **agents** are modeled as a (possibly variable) set $I(t)={1,\dots,N(t)}$, each with a state vector that evolves in time. Each agent $i\in I$ has:

- **Position:** $x_i(t)\in M$, its location on the manifold. Motion is governed by dynamics (continuous or discrete) on M, e.g. $\dot x_i = v_i(s_i,{s_j})$ or by jumps along geodesics.
- **Internal State:** $s_i(t)\in S_i$, including attributes such as cost $C_i(t)$, value $V_i(t)$, expected utility $U_i(t)$, information coefficient $\kappa_i(t)$, strategy variables, etc. Thus s_i may belong to $\mathbb{R}^m$ or a product space. As agents move or join groups, new state coordinates can be appended. For example, entering a region with a special resource may add a local-field variable to s_i.

- **Decision/Control:** Agents select actions or controls $a_i(t)$ (e.g. velocities, strategy updates) to optimize an objective. Each agent has a cost (or utility) function $J_i(s_i,{s_j: j\in \mathcal N_i}, \Phi)$, depending on its state and possibly neighbors $\mathcal N_i$ in the network and on local field values $\Phi(x_i)$. Agents follow dynamics aiming to minimize cost or maximize utility. In a game-theoretic setting, we may write $U_i = V_i - C_i$ and assume agents adjust their s_i (or a_i) by gradient descent/ascent or best-response.

- **Coupling Matrix:** We encode pairwise interactions by a (time-dependent) **coupling matrix** $W(t)=[w_{ij}(t)]$. Here $w_{ij}(t)\ge0$ represents the strength of influence of agent j on agent i. In practice, $w_{ij}>0$ only if i and j are allowed to interact (e.g. neighbors in the information graph or within a distance threshold on M). For instance, one may impose $w_{ij}(t)=0$ whenever the **interact_state** rule is violated (e.g. different teams or non-overlapping strategies). In summary, each agent i has a neighborhood $N_i(t)={j:w_{ij}(t)>0}$ determined by W. The adjacency/weight matrix W can itself depend on agent states (making it dynamic) or on the geometric distance $d_g(x_i,x_j)$. For example: $$ w_{ij}(t) = f\big(d_g(x_i,x_j), s_i(t), s_j(t)\big), $$ with f imposing $w_{ij}=0$ if $d_g>R$ (a range) or if an “interact-state” flag is off.

- **State Dynamics:** Agent states evolve by coupled differential or difference equations. We can posit for each agent i an ODE of the form $$ \dot s_i = F_i\big(s_i,{s_j:j\in N_i}, \Phi(x_i), t\big)\,, $$ where $\Phi(x_i)$ represents the current value of fields (geometry or ϕ) at x_i. In particular, the motion part obeys $\dot x_i = v_i(s_i,\cdots)$ (for some velocity v_i). For example, if agents perform gradient descent on cost, one might set $\dot s_i = -\nabla_{s_i} J_i$ or a replicator equation. Thus the collection of agents defines a multi-agent dynamical system (MAS) [3] .

Importantly, as agents move or form coalitions, their individual states may be augmented. For instance, joining a group of agents might attach a group membership index to s_i, or encountering a new environment might introduce local field variables. These **additional state variables** can be formalized by having s_i lie in a variable state space $S_i(t)$ that depends on location or group membership.

Citations: By definition, a multi-agent system is “a system composed of multiple interacting intelligent agents” [3] . In such systems, agents act to optimize local objectives subject to coupling. Agent interactions mediated by a graph are formalized in **graphical games**, where each agent’s payoff depends on its graph neighbors. As Kearns *et al.* showed, a *graphical game* represents such networked payoffs. In parallel, network theory uses adjacency and Laplacian matrices to capture coupling: for an (unweighted) graph the

Laplacian $L=D-A$ has $L_{ii}=\deg(v_i)$ and $L_{ij}=-1$ if i,j are adjacent [4]. In our setting, $W(t)$ generalizes this by weighting edges based on geometry or state.

Informational Layer (Dynamic Network)

We introduce an **informational network** (multi-layer graph) whose nodes are the agents. This network carries the structure of information flow, history, and strategy sharing among agents.

- **Graph Structure:** Let $G(t)=(V,E(t))$ be a (possibly directed or weighted) graph on the agent set $V=I(t)$. Each edge $e_{ij}\in E$ encodes that agent i receives information from agent j (or they interact socially). Edges may be typed or layered to represent different information channels (e.g. a *communication* layer, a *trust* layer, a *strategy-sharing* layer, etc.). Equivalently, we can consider a *multilayer network* or *multiplex network*, where each layer ℓ is a graph G^ℓ on the same node set. In a multiplex, layers share the same nodes but have different adjacency, and each node has "copies" in each layer. Interslice (interlayer) couplings then link each node to its copies in other layers [5].

- **Dynamics of the Graph:** The edge set $E(t)$ evolves via rules for linking, sharing, and grouping. For example, if agent i chooses to share its cost function with j, we may add a directed edge $i\to j$ annotated with that information. Conversely, edges may disappear if communication halts. Formally one can define a dynamics like $$ \dot a_{ij} = H_{ij}(s_i,s_j, a_{ij},\ldots)\,, $$ or use stochastic link-formation rules. Thus $G(t)$ is a **temporal network**, changing as agents form new connections or break old ones based on state conditions.

- **Edge Attributes:** Each edge can carry rich data: e.g. the weight w_{ij} from above, a history log, a coupling strength, or copies of agent variables (costs, strategies) being shared. If agent i shares its strategy p_i with neighbor j, this could be encoded as edge data that influences j's subsequent decision.

- **Groups and Hypergraphs:** Groups of agents with a common joint objective can be formalized via **hypergraphs** or **hypernetworks**. Mathematically, a hypergraph is defined by a set of vertices V and a family of hyperedges $\mathcal H\subseteq \mathcal P(V)$, each hyperedge $h\in\mathcal H$ being a subset of agents [6]. A group of agents acting in concert corresponds to a hyperedge connecting them. We can then define a **group payoff** $U_h({s_i: i\in h})$ for the entire hyperedge. This is naturally non-separable in general. One can express such a multivariate utility using *copulas*: by Sklar's theorem, any multivariate joint utility function can be decomposed into marginals linked by a copula function. Practically, one may treat each hyperedge as a complete subgraph in a multilayer representation: each group defines a layer in which all its members are interconnected [6]. In a multiplex viewpoint, group membership corresponds to communities across layers.

- **Information Propagation:** The graph $G(t)$ governs how information (costs, strategies, history) propagates. For example, an information diffusion could follow a process like $$ \dot s_i^{\rm info} = \sum_{j\in N_i(t)} a_{ij}(t)\big(s_j^{\rm info}-s_i^{\rm info}\big)\,, $$ analogous to diffusion on a graph (using the Laplacian [4]). Here $s_i^{\rm info}$ could be a strategic variable or belief. More complex models (e.g. rumor spreading, consensus) can be employed.

In summary, the informational layer is a **dynamic graph or multilayer network** that couples agent states and carries group structures. It interacts with the geometric layer: for instance, links may form preferentially between geographically close agents on M, or the existence of an edge may influence agent motion (e.g. i may move toward j if they communicate). Conversely, agent proximity on M may trigger new edges in $G(t)$.

Citations: Multilayer networks and hypergraphs are well-studied abstractions [6] [5]. For example, one can map each group (hyperedge) to a layer in a multilayer graph [6], and represent each layer by its adjacency matrix [5]. This justifies using such structures to encode dynamic coalitions and information channels. Graph Laplacian theory [4] provides tools (eigenvalues, spectral methods) to analyze diffusion and consensus on this network.

Coupled Dynamics of Geometry, Agents, and Network

We now describe how these components interact and how to model their joint evolution.

- **Geometry $\leftrightarrow$ Agents:** The metric $g(t)$ on M shapes local interactions. Agents experience distance $d_g(x,y)$ and curvature-dependent effects. For instance, travel cost between agents at x_i and x_j is a function of geodesic distance $d_g(x_i,x_j)$ and local curvature along the path. Curvature can create "obstacles" (e.g. narrow passageways on M) that affect agent movement. On the other hand, agents influence the geometry: we include coupling terms in the field PDEs that depend on agent locations. Concretely, one may write a coupled system such as $$ \partial_t g = -2\,\mathrm{Ric}(g) + \kappa\,\sum_i \delta_{x_i(t)}\,, \qquad \partial_t \phi = D\,\Delta_g \phi + S(\phi,{x_i})\,, $$ where δ_{x_i} is a Dirac mass at the agent position. Thus, each agent can "source" or "sink" curvature or field values.

- **Geometry $\leftrightarrow$ Network:** The manifold geometry can influence network connectivity. For instance, we may impose a linking rule that $a_{ij}(t)>0$ only if $d_g(x_i,x_j)<R$ (connect only nearby agents), or weight edges by a decreasing function of distance $w_{ij}\propto f(d_g(x_i,x_j))$. Geodesic distances on M thus modulate graph structure. Conversely, the network can affect the effective geometry as seen by agents: for example, if agents communicate to share travel routes, the "effective distance" may be lower along heavily linked paths, mimicking a deformation of the metric.

- **Agents $\leftrightarrow$ Network:** Agent interactions depend on the network, and the network evolves via agents. For each agent i, its dynamics $\dot s_i$ include terms from neighbors in $G(t)$: e.g. strategic update $\dot s_i = F_i(s_i,{s_j:j\in N_i})$. Graph adjacency enters the agent dynamics explicitly. Meanwhile, as agents change state or form coalitions, the graph updates: $\dot a_{ij}$ depends on s_i,s_j. For example, if agents reach a strategic agreement, an edge may be formed to lock that agreement.

- **Group Cooperation (Copula):** When agents form a group h, they optimize a joint utility $U_h(s_h)$ over their combined states. Such a group payoff is typically non-additive. We can model U_h by a multivariate utility function and use copulas to capture dependencies. In practice, one may assign to each hyperedge h a parameterized joint utility or consensus strategy. For instance, if two agents i,j form a coalition, their utility $U_{ij}(s_i,s_j)$ could be specified by a copula linking their marginal utilities, capturing correlation between their objectives.

Modeling Summary (Bulleted): One way to formalize the full system is:

- **Underlying Geometry:** $(M,g(t))$ a time-evolving Riemannian manifold. Define distance d_g, Laplacian Δ_g, and curvature tensors $R_{abcd}(x,t)$. Field variables $\phi_k(x,t)$ satisfy PDEs on M (diffusion, reaction, curvature flow, etc.). Example: Ricci flow $\partial_t g=-2\,\mathrm{Ric}(g) $ [1], and a diffusion equation $\partial_t \phi = D\,\Delta_g\phi + F(\phi,\rho_{\rm agents})$.

- **Agent States:** For each agent i, state $s_i=(x_i,C_i,V_i,U_i,\kappa_i,\ldots)$ with position $x_i\in M$. State evolves by an ODE/algorithm: e.g. $$\dot x_i = v_i(s_i,\Phi),\quad \dot C_i = G_i(s_i,{s_j}_{j\in N_i},\Phi)\,,\ldots$$ possibly derived from optimizing a cost function $J_i(s_i,{s_j})$. In game-theoretic terms, agents play a (differential) game; $\dot s_i$ might represent gradient-play dynamics.

- **Interaction Graph:** A graph $G(t)=(V,E(t))$ or multiplex with adjacency matrices ${A^\ell(t)}$. Edges are created/removed by rules $H(a_{ij},s_i,s_j)$, and carry attributes (weights, shared info). For weighted graph, define degree $D_{ii}=\sum_j A_{ij}$ and Laplacian $L=D-A$ [4].

- **Coupled Equations:** The full system can be written as a **hybrid dynamical system**: $$ \begin{cases} \partial_t g = F_g(g,\mathrm{Ric},{\rho_i}), \ \partial_t \phi = F_\phi(\phi,\Delta_g,{\rho_i}), \ \dot s_i = F_i(s_i,{s_j: j\in N_i(t)},g,\phi), \ \dot a_{ij} = H_{ij}(a_{ij},s_i,s_j). \end{cases} $$ Here ρ_i denotes an agent-induced measure at x_i. For continuum approximations (many agents), one may derive PDEs for the agent density $\rho(x,t)$ on M and HJB/Fokker–Planck equations (as in mean-field games) [7].

These equations are amenable to both analytical and computational treatment. One can analyze equilibria (e.g. Nash equilibria of the game) and stability using dynamical-systems methods (Lyapunov functions, eigenvalue analysis of the coupled ODE/PDE). The graph Laplacian theory [4] and spectral graph methods apply to the network layer. Geometric analysis tools (e.g. curvature bounds, geodesic convexity) apply to the manifold layer. Overall, the model constitutes a **coupled dynamical system** on the product of a manifold, agent state space, and network state space.

Computational Simulation Framework

To simulate this system, one can combine *agent-based modeling* with numerical PDE solvers on manifolds and network evolution algorithms:

1. **Discretize the Manifold:** If M is a smooth surface or volume, represent it by a mesh or coordinate grid. Compute discrete geometry (approximate metric, Laplacian, curvature) via finite elements or finite differences on the mesh.

2. **Represent the Field:** Initialize field variables $\phi_k(x)$ on the mesh. Time-step their PDEs (e.g. using implicit Euler or splitting schemes) to update the field given agent feedback.

3. **Track Agents:** Maintain an array of agent states. At each time step, update each agent's position x_i (e.g. by integrating $\dot x_i=v_i$ along the mesh) and other state components using a chosen ODE integrator (e.g. Runge–Kutta). Agent decisions (cost minimization, strategy update) can be computed by gradient descent or game-theoretic solver at each step.

4. **Update Network:** Given agent states, apply the linking rule to update edges. This may involve adding or removing edges in a dynamic adjacency list. One can then update any edge attributes or propagate information (e.g. perform a network diffusion step using $L=D-A$ [4]).

5. **Loop:** Repeat: at each time step, update (g,ϕ) on the manifold, update agents, then update network. Ensure consistency (e.g. agent movement on M uses the latest geometry).

This hybrid simulation could be implemented in software by coupling a PDE library (for geometry/fields) with a graph library (for dynamic networks) and an agent-based platform. Agent actions at each step solve a local optimization or best-response problem, which can be done by local computation or learning (e.g. evolutionary algorithms). If the number of agents is large, one can also use **continuum approximations**: treat agent density $\rho(x,t)$ as a continuous field and solve a PDE (e.g. a continuity or Fokker–Planck equation) coupled to the geometry, as in mean-field game models on manifolds [7] .

Analytical Extensions and Abstractions

This framework suggests several higher-level mathematical abstractions:

- **Fiber Bundle View:** Conceptually treat the system as a fiber bundle: the base space is the manifold M, and over each point $x\in M$ we attach a fiber encoding local agent/network state. For instance, the fiber could be the set of agents currently at x (or in a neighborhood) together with their connection subgraph. The dynamics then live on this bundle, coupling base and fiber evolution.

- **Manifold of Networks:** One can consider the space of all possible adjacency matrices (or weighted graphs) on N nodes as another manifold or topological space. The network state $A(t)$ evolves on this space. Studying the combined manifold (M,g) and network manifold may involve tools from product manifolds or network geometry [2] .

- **Higher-Order Networks:** We already allow hypergraphs (groups). More generally, one might use simplicial complexes or higher-order networks to model multi-agent interactions beyond pairwise edges.

- **Algebraic Topology:** Tools like persistent homology could track evolving topological features (e.g. agent coverage holes in M). The interplay of the agent network and M's topology could be studied via cohomology or bundle theory.

- **Game-Theoretic Generalizations:** This setup generalizes differential games, replicator dynamics, and evolutionary game theory on networks. For example, one might define a **mean-field game** on the manifold as $N\to\infty$, using PDEs for density and value function [7] . These PDEs naturally incorporate the geometry of M.

- **Information Geometry:** One could define an "information manifold" whose points encode probability distributions of agent states or network configurations. Then geometric objects (Fisher information metric, divergence) could quantify information flows.

- **Optimization Perspective:** The whole system can be cast as an optimization or variational principle in an extended state space. For instance, one could define a global cost functional that includes geometry terms and agent utilities, and study its descent.

Potential Applications

This mathematically rich model has applications wherever space, geometry, and networks co-occur:

- **Social and Economic Systems:** Agents are individuals in a geographic region M (e.g. a city map), with a social or communication network overlay. They trade (minimize cost, maximize profit) under spatial constraints and share information on the network. Multi-layer networks can represent different social ties (family, work, online). (MAS are used in economics and social simulation [8].)

- **Robotics and Swarms:** Autonomous robots or UAVs moving on terrain (M), with an ad-hoc communication network. The geometry (obstacles, terrain curvature) affects motion cost, while a dynamic wireless network carries shared maps or tasks. Robots' objectives can form cooperative games (groups) for tasks like coverage or target search.

- **Epidemiology and Mobility:** Individuals move in a spatial region and also have a contact network for disease spread. The geometry influences physical contact rates (travel time), while the network models social contacts. Control measures (vaccination strategies) can be studied as agent utilities.

- **Ecology:** Animals or agents move over a landscape (M) and form social or predator–prey networks. The environment provides resources ($\phi(x)$) that they compete for (cost) or share (cooperate).

- **Transportation Networks:** Vehicles navigating a road network (spatial graph on M) and communicating (e.g. V2V). Here M could be the road surface, and the informational layer is a traffic information network. Agents (drivers) optimize travel time (cost) and may form car-following coalitions (groups).

- **Computational Biology:** Molecules or cells moving on a curved membrane (M) with signaling network connections. Geometry affects diffusion of chemicals, and cell–cell network transmits signals. Cells optimize resource use (cost) and form clusters (groups).

- **Computer Science:** In distributed computing, nodes have an abstract "location" in a metric space and an overlay network. The geometry could represent latency distances, while the graph is the communication overlay. Nodes aim to optimize latency (cost) and throughput.

In all cases, our framework provides both a precise mathematical model and a simulation platform. It generalizes known structures (Riemannian geometry, dynamic networks, multi-agent games) into a unified setting. In particular, it extends the concept of **multi-agent systems** (multiple interacting agents [3]) to incorporate rich spatial geometry and information layers. This opens avenues for new analytical results (e.g. existence of equilibria on curved spaces, stability under network change) and complex simulations that couple PDEs on manifolds with graph dynamics.

Citations: Multi-agent systems have been applied in diverse domains such as online trading, disaster response, surveillance, and social modeling [8]. Our framework brings together differential geometry and network theory: e.g. recent work on geometric methods for networks [2] and mean-field games on manifolds [7] illustrates how such coupling can be rigorously defined and studied.

---

[1] **Ricci flow - Wikipedia**
https://en.wikipedia.org/wiki/Ricci_flow

[2] **[1910.06258] A Simple Differential Geometry for Networks and its Generalizations**
https://ar5iv.org/pdf/1910.06258

[3] [8] **Multi-agent system - Wikipedia**
https://en.wikipedia.org/wiki/Multi-agent_system

[4] **Laplacian matrix - Wikipedia**
https://en.wikipedia.org/wiki/Laplacian_matrix

[5] [6] **The structure and dynamics of multilayer networks - PMC**
https://pmc.ncbi.nlm.nih.gov/articles/PMC7332224/

[7] **Computational Mean-field Games on Manifolds**
https://arxiv.org/pdf/2206.01622