Concept Paper

Multi-Agent LLM Systems: From Emergent Collaboration to Structured Collective Intelligence

Feng Chen *

Posted Date: 18 November 2025

doi: 10.20944/preprints202511.1370.v1

Keywords: multi-agent systems; large language models; collective intelligence; debate and consensus; coordination protocols; scaling laws; scientific discovery

 doi:10.20944/preprints202511.1370.v1

Concept Paper

Multi-Agent LLM Systems: From Emergent Collaboration to Structured Collective Intelligence

Feng Chen

Independent Researcher

Abstract

Large language models (LLMs) are usually developed and evaluated as solitary agents: a single, monolithic network trained on static corpora and queried one prompt at a time. This single-agent paradigm has produced impressive capabilities, yet it fundamentally mismatches the structure of many real-world problems in science, engineering, and governance, which are inherently multi-actor, iterative, and argumentative. In this Perspective, we argue that the next scaling frontier for LLMs is not simply "bigger models with more data", but *societies of models and tools* designed as structured collective intelligences. We first outline why classical scaling laws, which relate performance primarily to parameter counts, token volume, and compute, are insufficient for tasks that require debate, division of labor, and long-horizon coordination. We then introduce a conceptual framework based on three interaction regimes—competition, collaboration, and coordination—and show how different task families naturally demand different regime designs, incentives, and communication protocols. Building on emerging multi-agent LLM systems in reasoning, code generation, and autonomous science, we sketch a research programmer for "multi-agent pretraining", in which agents jointly learn not only language and world models, but also norms of discourse, peer review, and self-correction. We further discuss how multi-agent architectures reshape scaling laws, evaluation methodology, and safety: performance becomes a function not only of model size and data, but also of team composition, interaction topology, and institutional memory. Finally, we argue that carefully engineered artificial communities may approximate the epistemic dynamics of real scientific communities more faithfully than any single, static model, opening a path toward more robust, transparent, and controllable AI systems.

Keywords: multi-agent systems; large language models; collective intelligence; debate and consensus; coordination protocols; scaling laws; scientific discovery

---

1. Introduction: From Solitary Models to Artificial Communities

The recent history of large language models is often told as a story of scale. As model parameters, training data, and compute budgets have grown by several orders of magnitude, performance on a wide range of benchmarks has improved in a remarkably regular fashion. Empirical "scaling laws" suggest that, for a broad class of architectures and training recipes, cross-entropy loss decreases approximately as a power law of model size, dataset size, and training compute. [1–3] This view has driven a decade of progress: given a fixed compute budget, one can choose an approximately optimal trade-off between parameters and tokens, as in compute-optimal recipes that recommend training smaller models on more data for better efficiency. [2,3]

However, these laws are fundamentally *single-agent* objects: they are defined for one model, trained on an effectively static dataset, evaluated on a static test set. They tell us nothing about how systems of multiple models, tools, and humans behave when coupled together over time. At the same time, real-world systems that we want AI to support—scientific fields, industrial supply chains, regulatory regimes, and climate and energy infrastructures—are inherently *multi-actor*. Scientific progress, for example, emerges from interactions between experimentalists, theorists, instrument

 doi:10.20944/preprints202511.1370.v1

builders, and reviewers; between laboratories that compete for priority and collaborate on standards; and between institutions that stabilize knowledge through journals, conferences, and funding structures. [12] It is thus increasingly natural to adopt a "machine behavior" perspective that studies AI systems as actors within socio-technical ecosystems rather than isolated optimization artefacts. [4] The rise of LLM-based agents makes this mismatch between solitary models and collective tasks particularly salient. Early "agentic" frameworks treat an LLM not merely as a text predictor, but as a decision-making component that can decompose tasks, call tools, and interact with humans and environments. [7–9] Surveys of LLM-based agents and tool-augmented models document an explosion of frameworks in which LLMs plan, act, and reflect across diverse domains. [7,14,16,22] In parallel, work on augmented language models and tool learning demonstrates that models can teach themselves to call APIs, search engines, and external programs, further blurring the line between pure prediction and action. [8,9,14]

These developments naturally generalize to *multi-agent* settings: instead of one agent coordinating all actions, multiple LLM instances, possibly with different prompts, tools, or fine-tunings, can collaborate or compete on tasks ranging from web navigation and software engineering to scientific discovery and governance. [10,16,22] Generative social simulators show that populations of memory-equipped agents can exhibit believable, emergent social behavior in persistent environments, such as a virtual town in which agents form routines, share information, and coordinate events. [12] Recent studies on multi-agent debate, self-reflection, and graph-structured reasoning illustrate that groups of interacting LLMs can improve factuality and robustness on some benchmarks—but also that naive "agent swarms" are prone to failure modes such as degeneration of thought, majority herding, and overconfident consensus. [11,13,15]

In this Perspective, we argue that the key scientific question is therefore shifting from "How big can one model be?" to "How should we architect and train structured collective intelligence?" We posit that the next scaling frontier involves not only more parameters and tokens, but also richer *interaction topologies* and *institutional rules* for societies of language agents. To make this case, we draw on two complementary traditions. From machine learning, we inherit the language of scaling laws, agents, and evaluation benchmarks. [1–3,7,15–18,21,22,41,42] From physics and materials science, we draw on systems where complex macroscopic behavior emerges from local interactions and constraints—such as droplets spreading on corrugated substrates, hygroscopic liquids suppressing condensation, fractal microfluidic networks, and facet-dependent electrochemical sensing. [25–31,33–38]

Experiments on wetting and electrowetting on corrugated and anisotropic substrates show how microscopic topography, sharp edges, and surface anisotropy pin contact lines, break symmetry, and induce directional spreading in droplet ensembles, even when the base chemistry is uniform. [26–28] Hygroscopic glycerol and dipropylene glycol droplets on cold hydrophobic surfaces act as local "vapor sinks" that suppress condensation and generate ring-shaped dry zones whose size depends sensitively on substrate temperature, cooling history, and solution composition. [33–37] Fractal microfluidic channels exhibit flow patterns and transport efficiencies that cannot be inferred from any single branch; they arise from the global topology of the dendritic network. [36] Polyacrylamide solutions display rich crystallization and self-assembly morphologies during evaporation, controlled by concentration, evaporation rate, and substrate interactions. [32] Facet-dependent electrochemistry on Au–Pd core–shell nanorods highlights how interface geometry at the nanoscale decisively shapes sensing performance. [38]

Across these systems, no single droplet or interface "knows" the macroscopic pattern. Global behavior is determined by the organization of many interacting components and the geometry of the substrate. Our claim is that multi-agent LLM systems are entering an analogous regime: global properties—robustness of beliefs, diversity of hypotheses, ability to sustain long-horizon projects—will be properties not of any single agent, but of the ensemble acting within an engineered communication topology and memory substrate. The remainder of this Perspective develops this argument in eight steps. In Section 2, we explain why the next scaling frontier is multi-agent rather

 doi:10.20944/preprints202511.1370.v1

than purely parametric, and introduce the idea of *collective scaling laws*. Section 3 proposes a taxonomy of interaction regimes—competition, collaboration, and coordination—and links them to different task families. Section 4 discusses architectural motifs for LLM societies, focusing on role specialization, shared memories, and communication topologies, with analogies to interfacial and microfluidic systems. Section 5 turns to training objectives, arguing that debate, consensus, peer review, and bargaining should be treated as first-class optimization targets. Section 6 outlines new benchmarks and metrics for measuring *collective* rather than individual intelligence, including recent work on HiddenBench and holistic evaluation frameworks.[15,41–44] Section 7 analyses risks and opportunities in delegating to artificial communities. Section 8 offers an outlook on how multi-agent LLM systems might evolve into engineered scientific communities that are deeply integrated with experimental platforms and human institutions.

2. Why the Next Scaling Frontier Is Multi-Agent, Not Just Bigger Models

Classical scaling laws for language models have been remarkably successful at predicting how performance improves as a function of model size, dataset size, and compute. [1–3] In their simplest form, these laws state that cross-entropy loss on held-out data decays as a power law in these three quantities, with smooth behavior across several orders of magnitude. This view has guided the design of GPT-3–class models and data-optimal Chinchilla-style models, and underpins much of the current practice in training foundation models.[1–3] Yet these laws implicitly assume a particular *shape* of intelligence: that of a single, unified agent trained on a static corpus and queried one prompt at a time. They say nothing about how performance scales when we compose many models and tools into a system that plans, acts, and communicates over long horizons. As LLMs are embedded into increasingly complex workflows—planning and acting in tool-augmented settings,[7–9,14,16] coordinating with other agents, and interacting with humans in rich environments[10,12,16,22]—this omission becomes critical.

There are also growing signs that "just scale up" is hitting both practical and conceptual limits. Practically, the cost and energy footprint of training ever-larger models is becoming unsustainable, while the high-quality text that fueled early foundation models is being rapidly exhausted. [1–3,17,18] Conceptually, there is a broadening consensus that scaling next-token prediction on static corpora will not by itself yield robust planning, grounded understanding, or the capacity to engage with open-ended scientific and engineering environments. [4,5,21,24] Scaling laws are therefore best seen not as a universal recipe, but as a description of a particular *regime*, one that largely ignores interaction, embodiment, and institutional structure. An instructive analogy comes from interfacial physics. The behavior of an isolated droplet on a smooth, homogeneous surface can often be described by relatively simple laws: Young's equation for equilibrium contact angles, basic scaling arguments for spreading and capillary length scales. But as soon as we introduce patterned substrates, sharp edges, or heterogeneous wettability, the behavior becomes dramatically richer. Droplets can be pinned on edges, exhibit anisotropic contact angles, or undergo complex shape transitions.[26–28] In such systems, macroscopic function—such as directional transport or condensation control—is no longer determined solely by droplet size and surface tensions, but by the *geometry and organization* of many droplets and interfaces.

The user's own work provides concrete examples of this transition from single-object to collective behavior. Studies of wetting and electrowetting on corrugated substrates and anisotropic surfaces show how sharp edges and patterned topographies pin contact lines and induce direction-dependent apparent contact angles, even when the underlying material properties are uniform. [26–28] Hygroscopic glycerol and dipropylene glycol droplets on cold hydrophobic surfaces suppress condensation by acting as local vapor sinks, generating ring-shaped dry zones whose sizes depend delicately on surface temperature, exposure time, and droplet composition. [33–35,37] Fractal microfluidic channels exhibit flow and mixing patterns governed by branching ratios and global topology, not just by local channel dimensions. [36] Polyacrylamide solutions show rich crystallization and self-assembly morphologies during evaporation, with pattern formation

 doi:10.20944/preprints202511.1370.v1

controlled by concentration, evaporation rate, and interfacial interactions. [32] Facet-dependent electrochemistry of Au–Pd core–shell nanorods reveals that even at the nanoscale, *organization of interfaces* (facet orientation, shell structure) is a decisive scaling dimension for sensing performance. [38]

These examples highlight a central point: scale is not only about size, but also about structure. In interfacial systems, one can "scale" by increasing droplet volume or channel width, but also by increasing the complexity of substrate patterns, the branching of channels, or the diversity of wetting regimes. The macroscopic function of the system—condensation suppression, mixing efficiency, sensing—depends on both. We argue that something similar is now true for language models. Classical scaling laws capture the effect of *parametric* scale: more parameters, more data, more compute. What they ignore is *organizational* scale: the number and diversity of agents, the topology of their interactions, and the richness of their shared environment and memory. As LLMs transition from solitary predictors to autonomous agents that plan, act, and interact, [7–10,14,16,22] a third axis of scaling becomes unavoidable.

We propose to conceptualize this new axis in terms of population, organization, and institution:

- Population scale refers to the number and diversity of agents in a system. A trivial multi-agent setup might simply duplicate the same base model several times; a more sophisticated one might include agents with different model sizes, training data, or alignment procedures, as well as specialist tools and simulators. Diversity in cognitive styles and priors—"optimistic vs skeptical", "global vs local"—plays a role analogous to polydispersity in droplet ensembles or heterogeneity in porous media.
- Organizational scale captures the topology and hierarchy of interactions: are agents arranged in a simple planner–worker structure, a star topology with a central judge, a deep hierarchy with multiple levels of review, or a graph with local neighborhood communication? Just as the branching ratio and connectivity of fractal microchannel determine flow and mixing patterns, [25,36] different communication topologies in LLM societies can lead to dramatically different modes of convergence, exploration, and error propagation.[10,15,22,24]
- Institutional scale concerns the maturity of norms, protocols, and shared memories that govern the system over time. Human scientific communities rely on journals, peer review, standards, and archives to stabilize knowledge and coordinate activity.[12] Artificial communities can similarly maintain institutional memory in shared vector stores, version-controlled artefacts, and explicit procedural templates. The user's work on condensation control and pattern formation under repeated cycles of wetting and drying provides a physical analogue: history-dependent phenomena—such as hysteresis in contact angles or path-dependent crystallization—show that past interactions matter for present behavior.[32–35,37]

Once we recognize this third axis, it becomes natural to talk about collective scaling laws: empirical relationships between performance and not only parameters and data, but also team size, diversity, interaction structure, and institutional complexity. Early experiments in multi-agent debate indicate that simply adding more agents does not monotonically improve performance: majority voting may help in some regimes, while unstructured debate can exacerbate "degeneration of thought", where agents collectively get stuck on an early, plausible-but-wrong line of reasoning. [11,13,15] Similarly, multi-agent orchestration frameworks for code generation and tool use exhibit diminishing returns or regressions when agent roles are poorly defined or communication is too dense. [7,10,16]

The implication is that *how* we add agents matters as much as *how many* we add. Just as placing hygroscopic droplets at the wrong spacing or on the wrong substrate can fail to suppress condensation—or even worsen icing by producing unintended gradients [33–35,37]—carelessly adding more LLM agents without principled interaction design can amplify biases, hallucinations, or unsafe behaviors. Conversely, carefully engineered societies of agents, with well-designed roles, communication protocols, and shared memories, may unlock capabilities that are inaccessible to any solitary model, regardless of its parameter count.

 doi:10.20944/preprints202511.1370.v1

In summary, the next scaling frontier for LLMs is multi-agent in a precise sense: we must develop theories and engineering practices for scaling *collectives* along axes of population, organization, and institution, and for characterizing the resulting collective scaling laws. This requires rethinking training objectives, evaluation methodologies, and safety frameworks not for individual models, but for artificial communities that increasingly resemble scientific and engineering institutions.[12,16,22–24]

3. Interaction Regimes: Competition, Collaboration, and Coordination

To turn multi-agent LLM systems from ad hoc "swarms" into structured collective intelligences, we need a vocabulary for how agents interact. A useful starting point is to distinguish three idealized interaction regimes—competition, collaboration, and coordination—and to understand how they manifest in LLM societies and in human scientific practice. In reality, most systems combine these regimes over time and across roles, but treating them separately clarifies design choices and failure modes.

3.1. Competitive Regimes: Debate, Self-Play, and Adversarial Search

Competitive regimes pit agents against each other with partially opposed objectives. In the context of LLMs, the most prominent example is multi-agent debate, where several agents propose solutions and arguments, critique one another, and a judge (human or model) selects a winner.[10,11,13,15,22,43] Intuitively, competition can expose flaws that a single reasoned might overlook: each agent has an incentive to find counter-examples, inconsistencies, or overlooked constraints. Empirically, debate-style protocols have improved performance on certain reasoning benchmarks, especially when agents are initialized with diverse samples and encouraged to disagree.[11,13,15] However, the design of competitive regimes is delicate. Recent work shows that naive debate can degenerate into majority dynamics, where the first plausible answer gains an early lead and is then reinforced by subsequent rounds—a phenomenon closely related to the "degeneration of thought" problem in single-agent chain-of-thought reflection. [13,15] If all agents share the same underlying model and training data, they may also share the same blind spots; competition then amplifies confidence in a wrong answer rather than correcting it.

These dynamics are reminiscent of pattern selection and symmetry breaking in interfacial systems. When droplets sit on a symmetric substrate, multiple configurations may be metastable; small perturbations or imperfections determine which configuration is selected. [26–28] Once a particular pattern—say, a certain direction of spreading or a particular rotation mode—is established, surface tension gradients and contact-line pinning can reinforce it, making it difficult for the system to transition to alternatives. [29–31] In competitive LLM regimes, early "pattern selection" in the space of arguments can similarly lock the collective into a particular trajectory, especially if the communication topology favors majority views or if judges are insufficiently skeptical. [11,13,15,43]

To harness competition effectively, we therefore need:

- Diversity of priors and roles among agents, so that at least some are predisposed to challenge majority views;
- Debate protocols that reward *novel* critiques and counter-examples rather than repetition;
- Judging mechanisms that can recognize when minority arguments are epistemically stronger than majority ones.

These design constraints parallel those in adversarial training and robustness: the benefit of adversaries depends on their diversity and strength relative to the main model.

3.2. Collaborative Regimes: Division of Labor and Team Reasoning

Collaboration regimes aim to combine complementary strengths to solve tasks no single agent could handle alone. In LLM societies, this often takes the form of role-specialized teams: one agent searches and summarizes literature; another performs mathematical derivations; a third writes and debugs code; a fourth checks safety or alignment constraints. [7,8,10,14,16,22] Such patterns mirror

 doi:10.20944/preprints202511.1370.v1

human scientific teams, where experimentalists, theorists, and data analysts work together on shared problems. [12] The user's materials-science workflows offer concrete examples of tasks that naturally invite collaborative decomposition. In studies of condensation suppression via hygroscopic droplets, one must integrate interfacial thermodynamics (vapor–liquid equilibrium, Raoult–Kelvin–Stefan coupling), numerical simulation of diffusion and heat transport, and experimental design for substrate preparation and imaging. [33–37] In work on fractal microfluidic channels, finite-element simulations, analytical approximations, and microfabrication expertise must be combined to understand how flow velocity and mass fraction vary across branches. [25,36] In electrochemical sensing with facet-engineered Au–Pd nanorods, surface science, electrochemistry, and signal-processing expertise are all required. [38] An LLM society tackling such problems could plausibly assign distinct agents to literature synthesis, physical modelling, experimental planning, and data analysis, with a coordinator weaving their contributions into coherent loops.

Collaborative regimes raise their own design questions:

- How to assign and evolve roles? Static hand-crafted roles may be a starting point, but over time the system should learn which agents are effective at which subtasks, and adjust division of labor dynamically.
- How to encourage information sharing without overload? If every agent broadcasts everything to everyone, communication becomes expensive and noisy. Conversely, if information stays soloed, the team cannot integrate its insights. This is closely analogous to balancing connectivity and mixing in fractal networks: too few connections and transport is inefficient; too many and flows interfere and recirculate.[25,36]
- How to prevent "free-riding" and over-reliance on a single strong agent? In human teams, social norms and incentives encourage each member to contribute. In LLM teams, one agent (often the largest or best-aligned) may end up doing most of the work. Training objectives and orchestration logic must explicitly value diverse contributions, not just final answers.[21,22]

From a learning perspective, collaborative regimes suggest multi-agent objectives that reward *marginal contribution* to group success, akin to credit assignment in cooperative multi-agent reinforcement learning. They also suggest opportunities for *curriculum design*: starting from simple pairwise collaborations and gradually increasing team size and heterogeneity as the system learns to coordinate.[21,22]

3.3. Coordinated Regimes: Orchestration and Workflow Execution

Coordination regimes focus on reliable execution of complex workflows over time, often under resource constraints. Here the central challenge is not to maximize diversity or adversarial challenge, but to ensure that the right actions are taken in the right order, with clear responsibilities and fallbacks. In LLM societies, coordination commonly appears as planner–worker architectures. A planner agent decomposes a task into subgoals, assigns them to worker agents (which may themselves call tools or interact with environments), and integrates their outputs. Variants include tree-structured planners, graph-based workflow engines, and controller–executor patterns in autonomous laboratories and software-engineering pipelines.[7,8,10,14,16]

Again, there are instructive analogies in the user's physical systems. Microfluidic platforms for controlled condensation, crystallization, or sensing rely on carefully orchestrated flows, temperature profiles, and reagent injections.[25,32–38] Valves, pumps, and channels must be actuated in precise sequences; timing errors can change nucleation pathways, flow regimes, or sensor responses. In such systems, coordination failures—deadlocks, race conditions, or mis-synchronisation—can be as damaging as incorrect local physics.[25,36]

For LLM societies, coordination design includes:

- Task decomposition strategies: how the planner represents tasks, chooses subtasks, and decides when to stop decomposition;
- Scheduling and resource allocation: which agents or tools are invoked when, subject to latency and cost constraints;

 doi:10.20944/preprints202511.1370.v1

- Failure handling and recovery: how the system detects when a subtask has failed or produced inconsistent results, and how it retries, escalates, or replans.

Poorly designed coordination can lead to oscillations (planners repeatedly revising plans without progress), deadlocks (agents waiting on each other indefinitely), or brittle pipelines that collapse under small deviations. These phenomena are closely related to instabilities in controlled physical systems, such as feedback-induced oscillations in thermal management loops or choking in branching flow networks.[25,36]

3.4. Regime–Task Alignment And Dynamic Regime Switching

In practice, effective multi-agent LLM systems will almost always blend competition, collaboration, and coordination. A scientific discovery pipeline might use competitive regimes during hypothesis generation and critique, collaborative regimes during model building and experimental design, and coordinated regimes during automated execution and reporting.[10–12,21,22,33–37] A key design question is therefore regime–task alignment: which regimes are appropriate for which stages and domains, and how should systems switch between them? Debate-like competitive regimes seem particularly suited for focused reasoning problems with clear correctness criteria.[11,13,15] Collaborative regimes are natural for open-ended design tasks where diverse ideas and skills are needed. Coordinated regimes are essential for long-horizon, safety-critical workflows, such as running high-throughput experiments in a chemical lab or deploying code to production services.[7,10,16] Dynamic regime switching—e.g. starting with a collaborative brainstorming phase, then entering a competitive critique phase, and finally committing to a coordinated execution plan—is likely to be crucial for aligning LLM societies with human expectations. Designing such switches requires both *meta-control* (agents or policies that decide which regime to invoke when) and *meta-learning* (the ability to adapt regime choice based on experience).

4. Architectures for LLM Societies: Roles, Memories, and Communication

If interaction regimes describe *how* agents relate to one another, architectures describe *what they are made of*: which roles are instantiated, how memory is organized, and how messages move through the system. Existing surveys on LLM-based agents identify core modules—profile, memory, planning, and action—that repeat across many single-agent frameworks.[7,16,22] Multi-agent systems enrich this picture by turning these modules into *distributed resources*: profiles correspond to heterogeneous agents, memories become shared institutions, and planning and action are spread over a communication topology rather than concentrated in a single controller.

4.1. Role Specialization and Cognitive Diversity

The simplest multi-agent architecture instantiates several *identical* copies of the same base LLM and lets them interact. While this can already help—independent sampling reduces some idiosyncratic errors—its benefits are limited by the strong correlation between agents' priors and failure modes.[11,13,15] In practice, effective LLM societies tend to rely on role specialization and cognitive diversity. Role specialization can be implemented at several levels. At the prompt level, agents are assigned different *personas* ("optimist", "skeptic", "formal proof assistant", "experimentalist"), which bias them towards particular reasoning styles. [10,13,22,43] At the tooling level, agents are endowed with different action sets: one can call literature search APIs, another can run simulations, a third can interact with a robotic lab, a fourth can access governance policies and safety checklists. [7–9,14,16] At the model level, different backbones or fine-tunes can be used: a small, fast model for broad exploration, a large, careful model for final validation.

The rationale is analogous to introducing controlled heterogeneity in physical systems. In wetting on corrugated or anisotropic substrates, symmetry breaking and directional spreading arise when contact lines encounter sharp edges or patterned wettability.[26–28] Small geometric differences in groove orientation or edge sharpness produce qualitatively different macroscopic behaviors, from pinned droplets to guided capillary films. Likewise, in hygroscopic condensation

 doi:10.20944/preprints202511.1370.v1

control, mixtures of ethylene glycol, dipropylene glycol, and glycerol with different viscosities and hygroscopicities produce distinct dry-zone morphologies and temporal dynamics.[33–37] Carefully combining such components yields richer and more controllable behavior than any uniform film.

For LLM societies, designing *useful* diversity means more than randomizing temperature or seeds. It involves: (i) curating different knowledge priors (e.g. one agent fine-tuned on safety policies, another on numerical methods, another on materials science); (ii) encouraging distinct reasoning strategies (e.g. short, direct answers vs exhaustive enumeration of hypotheses); and (iii) explicitly training agents for complementary roles, such as proposal, critique, and synthesis.[11,13,15,22,43]

4.2. Shared Memory and Institutional Knowledge

Single LLM calls are ephemeral: each conversation has its own context window and is forgotten when the session ends. In contrast, genuine collective intelligence requires institutional memory—persistent artefacts that outlive individual interactions. Existing agent frameworks introduce memory modules that store past trajectories, tool outputs, and user preferences in vector databases or key–value stores.[7,16] Multi-agent systems can go further by separating *individual* from *shared* memories.

Individual memories allow each agent to build its own history of successes, failures, and idiosyncratic preferences. Shared memories function as community artefacts: design documents, experiment logs, code repositories, policies. They serve the same role as lab notebooks, version control systems, or standards documents in human scientific communities.[12]

Designing such memories raises several questions:

- Granularity. What events deserve to be written into institutional memory? Storing every intermediate thought is infeasible and undesirable; instead, systems must learn to extract "commit-worthy" artefacts: accepted hypotheses, vetted protocols, approved code patches.
- Structure. Should memory be predominantly vector-based (for flexible retrieval) or symbolic (for explicit constraints and traceability)? Hybrid approaches can, for example, index structured records (e.g. an experimental run of condensation control conditions and outcomes) with both symbolic keys and learned embedding.[32–37]
- Revision and forgetting. Scientific institutions constantly revise their knowledge: retractions, updated standards, superseded protocols. LLM societies need mechanisms for amending or retiring outdated entries, lest they be haunted by early mistakes—especially when systems autonomously generate synthetic data or self-imposed "norms".

Again, interfacial systems provide instructive analogies. In condensation suppression by hygroscopic droplets, the substrate accumulates a history of exposure: repeated cycles of icing and drying can roughen surfaces, redistribute solutes, and create microscopic defects that act as new nucleation sites. [33–35,37] In evaporative self-assembly of polyacrylamide, transient concentration fields and crystal growth leave lasting patterns that influence subsequent deposition. [32] These systems exhibit hysteresis: the state of the interface encodes a memory of past conditions, which in turn shapes future dynamics.

Translating this into LLM architectures, we can think of institutional memory as an engineered "hysteresis" mechanism: the collective does not reset between tasks; it accumulates structured traces that bias future interactions. Unlike physical substrates, however, we can design explicit policies for when and how these traces are updated. A promising direction is to treat memory editing itself as a *governed multi-agent process*, with specialized "archivist" and "editor" agents reviewing proposals to add, modify, or delete entries, analogous to journal editors or standards committees.

4.3. Communication Topologies and Substrates

The third architectural ingredient is communication topology: who can talk to whom, and through what channel. Current LLM multi-agent systems largely rely on broadcast or hub-and-spoke patterns: all agents see the same conversation, or all messages pass through a central orchestrator. [10,16,22] More structured topologies—trees, rings, small-world graphs—remain underexplored, but

 doi:10.20944/preprints202511.1370.v1

early work on graph-structured prompting and multi-agent collaboration suggests that topology can significantly affect performance, resource usage, and emergent behavior. [15,24] Topology interacts with substrate, the medium through which messages flow. In practice, this substrate might be: plain natural-language chat; structured JSON messages; code and configuration files; or actions taken in an external environment (e.g. writing to a shared file system, interacting with instruments). Substrate design shapes what kinds of information are cheap or expensive to transmit. For instance, enforcing structured action logs that record parameter settings and outcomes for microfluidic experiments makes it easier for other agents to audit and generalize experimental results, much as well-annotated CAD files and process flows enable reproducibility in materials synthesis.[25,32–38]

Here the analogy to microfluidic channels and patterned substrates is especially tight. In fractal microchannel networks for heat and mass transfer, the choice of branching ratios, channel widths, and connectivity determines pressure drops, flow distribution, and mixing patterns.[25,36] Small changes—adding a bypass channel, narrowing a branch—can prevent stagnation zones or reduce local overheating. Similarly, in condensation and icing control, adding micro-grooves or wettability gradients provides preferred pathways for droplet motion and vapor transport, guiding collective behavior at the interface.[26–28,33–37] By designing the physical *substrate*, one can orchestrate many interacting droplets without individually controlling them. For LLM societies, prompt templates, tool APIs, and message schemas *are* the substrate. A topology with strong central bottlenecks (e.g. all communication through a single "boss" agent) risks overload and single points of failure. A fully connected topology risks redundancy and echo chambers. Hybrid designs—local neighborhood communication plus occasional global broadcasts, or hierarchical trees with cross-links—may offer better trade-offs between diversity and coherence.[15,24]

4.4. Design Motifs and Failure Modes

From these ingredients, we can identify recurring design motifs in LLM societies:

- Triadic structures with proposer, critic, and judge roles, echoing author–reviewer–editor triads in scientific publishing.[11,13,21,22,43]
- Committees plus executors, where a deliberative body evaluates options and an execution agent interacts with external systems.
- Guardrail agents that monitor conversations for safety and compliance, forming an institutional "immune system".[18,23]

Each motif brings characteristic failure modes. Triads can collapse into rubber-stamping if critics are too weak or correlated with proposers. Committees may suffer from groupthink or capture by a dominant agent. Guardrail agents might be bypassed or co-opted, as recent jailbreak and red-teaming studies on commercial LLMs demonstrate.[18,23] Understanding these motifs empirically—through ablation, benchmarking, and formal analysis—will be essential for moving from artisanal architectures to principled design of artificial institutions.

5. Multi-Agent Training Objectives: Debate, Consensus, Peer Review, Bargaining

Architectures specify *who* interacts and *how*; training objectives specify *what they are trying to achieve*. Most current LLMs are optimised for next-token prediction, followed by a layer of alignment via supervised instruction tuning and reinforcement learning from human feedback (RLHF).[4,5,17,18,23] These objectives are defined at the *individual* level: a single model is rewarded for producing desirable responses. Multi-agent systems call for collective training objectives that explicitly value group-level performance, diversity of perspectives, and robust decision-making.

5.1. From Individual Loss to Collective Objectives

In cooperative multi-agent reinforcement learning, a classic challenge is credit assignment: how to distribute a global reward signal among agents whose contributions are interdependent. LLM societies face a similar issue. If we judge only the final answer of a multi-agent debate or collaborative

 doi:10.20944/preprints202511.1370.v1

workflow, we have no signal for the quality of intermediate arguments, critiques, or coordination steps. Conversely, if we reward each agent purely on its local accuracy or persuasiveness, we may inadvertently incentivise behaviours that harm collective performance, such as persuasive but misleading arguments.[11,13,15,21,22]

A first step is to define collective metrics that encapsulate desirable group behaviour:

- Group accuracy on tasks with clear ground truth (e.g. mathematical proofs, physics word problems, code generation tests).[1–3,11,21]
- Calibration and uncertainty management, measuring whether the group's expressed confidence matches its actual reliability.[41,42]
- Diversity of hypotheses, rewarding settings where the group explores multiple plausible explanations before converging.[11,13,15,24,41]
- Conflict resolution quality, e.g. whether minority but correct views can eventually overturn majority but wrong ones, as tested in hidden-profile tasks.[41]

HiddenBench, a benchmark for collective reasoning in multi-agent LLMs based on the hidden-profile paradigm from social psychology, makes this gap precise.[41] Across a wide range of tasks and prompting strategies, groups of frontier models fail to integrate distributed information, displaying human-like collective failures such as majority amplification and neglect of critical but rare signals.

5.2. Debate-Style Objectives

Multi-agent debate frameworks provide a natural template for collective objectives. In a typical setup, two or more agents generate arguments for and against candidate answers; a judge—either another model or a human—selects a winner and possibly provides feedback.[11,13,15,22,43] Training objectives can then reward agents who (i) argue for correct outcomes, (ii) successfully expose flaws in incorrect arguments, and (iii) refrain from overconfident advocacy when evidence is weak. Recent work on multi-agent debate and "breaking mental set" shows that carefully designed debate protocols can reduce degeneration of thought and improve reasoning, but also that LLMs are prone to premature convergence and unfair judging when using homogeneous agent pools.[11,13,15,24,43] To address these issues, debate-style objectives should:

- Encourage novelty: reward agents for introducing new lines of evidence or alternative reasoning paths.
- Penalize redundancy: limit rewards for repeating already stated arguments, to avoid echo chambers.
- Incorporate meta-cognitive signals: allow agents to express uncertainty, defer judgement, or call for more evidence, and reward such caution when appropriate.

Concretely, this could involve fine-tuning agents on synthetic debate transcripts where good practices—citing relevant facts, acknowledging limitations, updating beliefs—are rewarded, using a combination of supervised learning and RLHF with human or high-quality model judges.[4,5,17,18,23]

5.3. Consensus and Self-Consistency

Beyond explicit debate, many multi-agent systems rely on consensus mechanisms: self-consistency across multiple samples, majority voting, or weighted averaging of agent outputs.[11,13,15] At present, these mechanisms are typically applied *post hoc* as inference-time tricks. A more principled approach is to incorporate consistency as a training signal. For instance, one can generate multiple independent solutions from a group of agents, identify discrepancies, and use ground truth or external verification tools (e.g. unit tests, physical simulators) to label which solution is correct. [7–9,14,21,22] Agents whose reasoning aligns with the verified solution are rewarded; agents whose reasoning was incorrect can be presented with counterfactual feedback ("here is why you were wrong") and fine-tuned to adjust their internal decision boundaries. Over time, this can

 doi:10.20944/preprints202511.1370.v1

shape the group's tendency to calibrate itself: when agents disagree, they learn to weigh arguments by diagnosticity rather than mere frequency.

In scientific workflows, consensus is rarely a simple vote; it emerges from chains of experiments, replications, and methodological critique.[12,32–37] Training LLM societies to mimic such *evidential consensus*—aggregating not only opinions, but also diverse lines of evidence—will require integrating external tools (simulators, lab platforms) into the training loop so that agents can test and update their claims.

5.4. Peer Review and Cross-Agent Critique

A particularly promising class of objectives models peer review. Here, one agent (the "author") proposes a solution, design, or hypothesis; one or more "reviewer" agents critique it; and an "editor" adjudicates and requests revisions. Each role has distinct incentives: authors are rewarded for clarity, novelty, and correctness; reviewers for identifying real flaws and providing constructive suggestions; editors for balancing rigor and throughput. [12,21,22] Such triadic patterns are emerging in code generation and document drafting, where drafting agents are paired with critic agents that highlight issues, and "meta" agents that synthesize corrections.[10,14,16] Extending this to more scientific settings suggests training objectives that:

- Reward reviewers when their critiques lead to measurable improvements in subsequent revisions (e.g. lower error rates, higher robustness).
- Discourage *spurious* criticism by penalising reviewers whose comments do not improve or actively degrade performance.
- Teach authors to *respond* to critique by revising appropriately, providing justification when they reject comments, much like human authors in journal rebuttals.

In the user's experimental pipelines, such as automated condensation control or microfluidic experiments, an "author" agent might propose an experiment (choosing droplet composition, substrate patterning, temperature profile); "reviewer" agents might check physical plausibility, safety, and novelty against prior experiments; an "editor" agent could approve experiments that pass certain thresholds and route them to a robotic lab. [32–37] Training such systems end-to-end—using experiment outcomes as the ultimate reward—could align AI-driven hypothesis generation with real-world scientific constraints.

5.5. Bargaining, Negotiation, and Resource Allocation

Many real-world multi-agent problems are not purely cooperative. Different stakeholders value different objectives (accuracy, cost, safety, fairness), and multi-agent LLM systems deployed in socio-technical contexts will have to negotiate trade-offs. Bargaining-style objectives explicitly train agents to articulate preferences, propose compromises, and converge on agreements that satisfy fairness or efficiency criteria. [21,22] For example, consider a scenario where multiple "lab PI" agents share a finite-budget automated experiment platform. Each proposes sets of experiments (e.g. different parameter sweeps for hygroscopic materials or microfluidic chips); a bargaining protocol must allocate resources fairly while preserving overall scientific throughput. Training LLM agents in such settings could use classic concepts from cooperative game theory (e.g. Nash bargaining solutions, Shapley-value-inspired credit assignment) as shaping signals, combined with constraints derived from safety and regulatory requirements.[18,23] At a broader level, negotiation objectives are relevant for AI governance itself. Multi-agent systems could mediate between regulators, industry, and civil society, exploring policy options and surfacing trade-offs. Training them for *procedural virtues*—transparency, respect for rights, responsiveness to evidence—will be at least as important as training for raw predictive accuracy.[4,5,18,23]

5.6. Towards Multi-Agent Pretraining

Today, most multi-agent behaviors are grafted onto models that were never explicitly trained to interact. A more ambitious agenda is multi-agent pretraining, where interaction patterns and

 doi:10.20944/preprints202511.1370.v1

collective objectives are built into the training process from the outset.[16,19,21,22,41] This could involve:

- Generating synthetic corpora of debates, peer reviews, and negotiations, possibly seeded by real scientific and engineering records (papers, reviews, code reviews, standards discussions).
- Training populations of agents jointly, sharing parameters where appropriate but allowing role-specific adapters or memory modules to diverge.
- Periodically evaluating collectives on benchmarks such as HiddenBench, multi-agent debate suites, and long-horizon task simulations, feeding back collective metrics into the training loop.[11,15,21,22,41–43]

The long-term vision is that, just as current LLMs internalize statistical patterns of language and world knowledge, future LLM societies will internalize norms of interaction: what it means to argue in good faith, to review responsibly, to coordinate reliably under uncertainty. Such norms will not emerge automatically from scaling individual models; they require collective objectives that treat multi-agent behavior as a first-class target of optimization.[16,19,21,22,41–44]

6. New Benchmarks: Measuring Collective, Not Individual, Intelligence

Most language-model benchmarks were designed for single agents answering static prompts. They measure task accuracy, sometimes robustness or bias, but almost never *collective* properties such as division of labor, institutional memory, or conflict resolution.[1–3,17,18,41,42] As LLMs are increasingly deployed as societies of agents, this evaluation regime becomes as incomplete as judging a research institute solely by the IQ scores of its members.

6.1. Why Single-Agent Benchmarks Are Insufficient

Single-agent benchmarks implicitly assume that "the system" is a single model producing a one-shot answer. In a multi-agent setting, however, *how* an answer is produced can matter as much as the final output:

- Is the solution the product of genuine information integration, or did one strong agent dominate while others free-rode?
- Does the group remain robust when some agents are noisy, adversarial, or misaligned?
- Can a correct minority view eventually overturn an incorrect majority, as in hidden-profile experiments in social psychology?

HiddenBench introduces precisely such tasks for multi-agent LLMs: each agent receives only a subset of the information, and the group must communicate to identify the correct choice.[41] Across a wide range of tasks and prompting strategies, even strong models fail to integrate distributed information, displaying human-like collective failures such as majority amplification and neglect of critical but rare signals. Similar concerns arise in multi-agent debate studies: simply adding more agents or rounds of argument does not guarantee better truthfulness, and can entrench early mistakes. [11,13,15,22,43]

These results echo the user's experimental observations in interfacial systems: adding more droplets, channels, or structural elements does not monotonically improve performance. For example, adding hygroscopic droplets in the wrong pattern can create undesirable condensation gradients, and increasing microchannel connectivity can induce recirculation zones and dead water pockets.[25,33–37] Evaluating such systems requires *system-level* metrics—heat-transfer efficiency, overall condensation suppression—not just measurements of any single droplet or channel.

6.2. Task Families for Collective Evaluation

What kinds of tasks should multi-agent benchmarks include? At least four families seem essential:

1. Distributed-information reasoning. Hidden-profile style problems, where no single agent has enough information to solve the task, but the group could in principle succeed through

 doi:10.20944/preprints202511.1370.v1

communication.[41] This probes whether agents can surface and integrate complementary evidence rather than amplifying shared priors.

2. Long-horizon projects. Tasks that require maintaining goals, plans, and artefacts over many steps and time scales—e.g. multi-stage codebase refactoring, iterative scientific experiment design, or multi-day project management. Generative-agent environments, in which agents inhabit persistent simulation towns, offer testbeds for institutional memory and norm formation.[12]

3. Multi-environment agent benchmarks. Suites that evaluate LLM agents across diverse environments—games, web tasks, tool-using scenarios—can be extended to multi-agent regimes where agents must share tools and coordinate actions.[7,10,14,16,22]

4. Safety- and governance-sensitive scenarios. Benchmarks where different agents play regulators, developers, auditors, and affected stakeholders, negotiating policies or red-teaming decisions. Multi-agent referee systems such as ChatEval, which use agent committees to evaluate generated text, already show how LLM panels can outperform single models in judgement tasks.[43]

These families mirror the diversity of behaviors seen in the user's physical experiments: local phenomena (e.g. contact-line pinning), long-time evolution (evaporation-driven self-assembly), multi-channel transport (fractal microfluidics), and safety-relevant regimes (icing, over-condensation).[25–28,32–37] No single benchmark captures them all; a *portfolio* of scenarios is needed.

6.3. Metrics for Collective Performance

Once tasks are defined, we need metrics that capture collective properties. Inspired by holistic evaluation frameworks such as HELM, which adopt multi-metric evaluations for single LLMs,[42] we can define a multi-dimensional scorecard for LLM societies:

- Task performance: final accuracy, solution quality, and resource usage (latency, tokens, tool calls).
- Robustness and fault tolerance: degradation under agent failures, corrupted messages, or adversarial perturbations.[17,18,41–43]
- Division of labor: how evenly are contributions distributed? Do specialized agents carry out the subtasks they are suited for, or does one "hero agent" dominate?
- Deliberation quality: diversity of hypotheses explored before convergence, depth of critiques, and degree of unjustified agreement.[11,13,15,41,43]
- Institutional memory and reproducibility: can the group reconstruct its past decisions, rationales, and experimental conditions from its logs and shared artefacts?

Some of these metrics require new instrumentation. For example, measuring division of labor may involve attributing marginal contribution to each agent, akin to Shapley values in cooperative game theory. Measuring deliberation quality may require structured human evaluation frameworks such as QUEST,[44] which decomposes evaluation into dimensions of information quality, reasoning, style, safety, and trust, and could be adapted to judge multi-agent transcripts.

6.4. Controlled Environments and "Wind Tunnels"

To systematically study multi-agent dynamics, we need controlled "wind tunnels": simulation environments where interaction structure, information asymmetries, and feedback signals can be varied independently. [41–44] HiddenBench uses carefully designed hidden-profile tasks as such a social wind tunnel. [41] Multi-agent debate frameworks propose debate arenas where agent roles and rules can be systematically ablated.[11,13,15,22,43] Generative-agent sandboxes populate a virtual town with memory-equipped agents and manipulate their environment to observe the emergence of social norms, gossip, and coordination patterns.[12]

Analogously, the user's microfluidic chips and patterned substrates serve as physical wind tunnels for interfacial phenomena, enabling precise control over geometry and boundary conditions to study emergent flows and condensation patterns.[25–28,33–37] A long-term goal is to build *standardized* wind tunnels for LLM societies: shared benchmarks and environments where different

 doi:10.20944/preprints202511.1370.v1

research groups can compare architectures, training schemes, and governance mechanisms on equal footing, much as standard microfluidic test structures are used across labs.

6.5. Reporting Standards and Transparency

Finally, collective benchmarks demand richer reporting standards. HELM emphasizes that transparency about models, scenarios, and metrics is as important as raw scores.[42] For multi-agent systems, transparency must extend to:

- Communication topology and agent roles.
- Training regime (single- vs multi-agent, debate-style fine-tuning, synthetic interaction data).
- Memory structures and access policies.
- Safety features (guardrail agents, rate limits, oversight loops).[17,18,23,42,43]

Without such system-level documentation—perhaps in the form of "ecosystem cards" complementing model cards—it will be impossible to compare or govern artificial communities responsibly.

7. Risks and Opportunities in Delegating to Artificial Communities

Designing LLM societies is not only a matter of performance; it also reshapes the risk landscape. Some threats familiar from single models reappear in new forms, while others are genuinely emergent properties of multi-agent interaction. At the same time, multi-agent architectures create opportunities for robustness and co-governance that solitary models lack. [18,21–23,41–44]

7.1. New Risk Patterns: Collusion, Echo Chambers, and Power Asymmetries

In single-model systems, we worry about hallucination, bias, and misalignment.[4,5,17,18,23] In multi-agent systems, we must additionally worry about collusion, echo chambers, and power dynamics:

- Collusion and jailbreaks. Multiple agents can conspire—intentionally or emergently—to bypass safety mechanisms. For example, one agent might rephrase disallowed content into innocuous-looking instructions that another agent executes, or a group might gradually normalize unsafe actions through repeated mutual reinforcement. Multi-agent debate protocols can, in principle, be repurposed to *game* evaluation metrics or persuade judges of incorrect conclusions.[11,13,15,22,43]
- Echo chambers and groupthink. As HiddenBench demonstrates, LLM groups can fail to integrate distributed information, instead amplifying shared but incomplete priors.[41] If communication topologies favor majority views and judges are insufficiently skeptical, groupthink becomes the default: the appearance of consensus hides a fragile epistemic base.
- Power asymmetries. In heterogeneous societies, some agents (larger models, those with privileged tool access, or those closer to human interfaces) may disproportionately shape outcomes. Over time, institutional memory and decision logs can entrench these asymmetries, much like path-dependent processes in physical and social systems.[12,32–37]

These risks resemble failure modes in human institutions: regulatory capture, polarized echo chambers, and inequitable representation. They also echo interfacial phenomena where small asymmetries—e.g. a slightly rougher region of substrate or a preferred nucleation site—can dominate long-term pattern formation.[26–28,32–37]

7.2. Accountability, Traceability, and Audit

As multi-agent systems take on higher-stakes roles—autonomous labs, financial decision-making, policy analysis—questions of accountability become central. When a collective makes a harmful recommendation, who (or what) is responsible? How can we reconstruct the chain of reasoning that led to a decision?[18,23,42–44]

At minimum, artificial communities should maintain tamper-evident logs of:

 doi:10.20944/preprints202511.1370.v1

- Which agents participated in which decisions.
- What information each agent saw and produced.
- How final outputs were synthesized from intermediate contributions.

Such logs enable post-hoc audit and, in principle, could support formal responsibility assignments. They also provide data for improving the system: by analyzing where debates derailed or coordination broke down, designers can refine roles and protocols. Single-model evaluation frameworks such as HELM emphasize documenting training data, evaluation scenarios, and known limitations. [42] For LLM societies, documentation must cover *interaction-level* details as well. Multi-agent referee systems like ChatEval, and psychometric frameworks such as QUEST, already point toward richer logging and analysis of agent–agent interaction patterns. [43,44]

7.3. Opportunities: Robustness, Diversity, and Human–AI co-Governance

On the positive side, carefully designed multi-agent systems can address some weaknesses of solitary models:

- Robustness through redundancy. Multiple agents with diverse priors and tools can cross-check each other's outputs, catching errors that a single model might overlook.[11,13,15,21,22,41–43] Analogous to redundant sensors in engineering or multiple droplets suppressing condensation in different zones of a cold surface, diversity can increase fault tolerance—if interaction protocols prevent herding.
- Bias mitigation via cognitive diversity. By embedding agents tuned to different normative frameworks (e.g. different fairness definitions, privacy preferences, or stakeholder perspectives), a system can surface tensions and trade-offs rather than silently optimising for one objective. Human overseers can then make more informed decisions. [4,5,12,18,23]
- Human–AI co-governance. Multi-agent architectures naturally accommodate *human agents* as first-class participants: reviewers, auditors, or domain experts who can join deliberations, veto decisions, or reshape protocols. In contrast to monolithic black-box models, artificial communities can be designed to expose interfaces at multiple levels of granularity: from individual debates to committee reports. [12,18,21–23]

This suggests a vision of human–AI institutions where humans and LLM agents share decision-making, analogous to mixed committees in regulatory science. In the user's domain, one could imagine automated materials labs where LLM agents propose experiments, robotic systems execute them, and human scientists oversee and occasionally intervene—much like PIs guiding PhD students and postdocs. [32–38]

8. Outlook: from Emergent Swarms to Engineered Scientific Communities

Multi-agent LLM systems today resemble early microprocessor networks: powerful individually, surprisingly capable in small swarms, but still lacking the engineered abstractions, protocols, and safety rails that turned raw hardware into the internet and modern cloud computing. [1–5,7,10,14–16,21–24,41–44] In that sense, the "collective intelligence" frontier is less about inventing a brand-new paradigm and more about importing decades of lessons from distributed systems, markets, and scientific communities into the design of artificial societies. One near-term trajectory is deep vertical integration with scientific infrastructure. In materials and interfacial science, we already see tightly coupled loops connecting theory, simulation, and experiment—say, from analytical models of droplet pinning and anisotropic wetting on corrugated or sharp-edged substrates to high-speed imaging and controlled microfluidic experiments on fractal networks. [25–28,32–38] In a multi-agent LLM setting, this loop can be operationalized: one family of agents specializes in domain theory and physical scaling arguments; another orchestrates simulation pipelines and checks regime validity; a third monitors experiments in real time, performing image-based measurements and uncertainty estimation; a fourth curates and compresses the resulting data into reusable "notebooks" that later agents can build on. Over time, such systems could discover their own "design rules" for workflows, just as they already discover reusable sub-skills in tool-use benchmarks.[7–9,14,16]

 doi:10.20944/preprints202511.1370.v1

A second direction is institutionalizing norms inside agent societies. Human science progresses not just because individuals are smart, but because communities enforce norms: reproducibility, credit assignment, adversarial peer review, and cumulative synthesis.[12] Multi-agent LLM systems make it technically feasible to simulate such institutions. Instead of a single monolithic "critic", one can instantiate panels of reviewers with orthogonal incentives: one penalizes overfitting to benchmarks, another penalizes unverifiable claims, a third focuses on experimental feasibility. Their disagreements become explicit signals for where further evidence is needed, and their consensus becomes a high-precision prior on which directions are worth expensive real-world experiments. [11–13,15,21,22,41–44] In the long run, we might treat these artificial institutions as software-defined scientific bodies: configurable, auditable, and co-evolving with human communities.

A third pillar is richer, explicitly engineered communication topologies. Most current multi-agent systems rely on simple broadcast or fully-connected chat; yet work on graph-structured prompting already suggests that non-trivial topologies—trees, DAGs, feedback cycles—can substantially improve reasoning efficiency. [15,24] Generalizing this, we can imagine lattices of agents reflecting physical constraints (e.g. spatial locality in microfluidic networks or pilot plants), hypergraphs capturing shared intermediate artefacts (e.g. a common thermodynamic model used across tasks), or layered architectures mirroring experimental abstraction levels (molecular, mesoscopic, device-scale). Designing and learning these topologies becomes a first-class research object, analogous to architecture search in deep learning, but at the level of collectives rather than individual networks. [15,16,21,22,24,41–44] Fourth, data becomes an institutional resource rather than a static corpus. In continuous scientific campaigns—such as long-term studies of condensation suppression, hygroscopic water harvesting, or catalyst performance—data is streamed under evolving conditions, instrument calibrations drift, and the definition of "interesting" events changes over time. [32–40] Multi-agent LLM systems are natural containers for these dynamics: one agent can specialize in detecting distribution shift; another in proposing new measurement protocols to characterize it; a third in reweighting or re-indexing historical data so that cross-regime generalization remains possible. If the "new Moore's law" is about compound growth in high-quality, task-aligned data, then agent societies are the mechanism by which that data is selected, distilled, and turned into durable scientific knowledge rather than raw logs. [16,19,21–24,39,40]

A particularly promising, but under-explored, direction is to ground multi-agent design in physical metaphors and constraints. Interfacial physics offers a rich vocabulary here: pinning vs depinning of contact lines as an analogy for commitment and revision in argument; vapour-sink-induced dry zones as an analogy for informational shielding around fragile hypotheses; anisotropic wetting and fractal microchannel networks as analogues of directional information flow and multi-scale coordination. [25–28,33–37] These metaphors are not just rhetorical. They can guide concrete design choices—for example, introducing "energy barriers" to prevent agents from flipping conclusions without sufficient new evidence, or designing "capillary channels" of communication that preferentially route highly compressed summaries rather than raw logs, mirroring how microstructures control liquid transport in real materials. [25,26,33–37]

Finally, there is the question of governance and co-evolution with human institutions. As agent societies take on more responsibility for proposing experiments, allocating instrument time, or triaging candidate hypotheses, they become de facto decision-makers in scientific and industrial processes. [18,21–23,41–44] Governance then becomes multi-layered: we need mechanisms to align individual agents with collective objectives; mechanisms to align artificial collectives with the labs, companies, and regulatory bodies that deploy them; and mechanisms for these artificial communities to remain legible and contestable to humans. Multi-agent architectures make this both harder and easier: harder because behavior emerges from complex interactions, easier because we can assign explicit roles—ombudsman agents, audit agents, dissent-seeking agents—whose sole purpose is to keep the system interpretable and corrigible.

In summary, the shift from single-agent LLMs to structured multi-agent systems is not just an engineering trend but an opportunity to rethink how we "do science" in the age of AI. The most

 doi:10.20944/preprints202511.1370.v1

impactful systems will likely resemble well-run research institutes more than solitary geniuses: diverse, argumentative, self-correcting, deeply integrated with instruments and simulations, and embedded in broader human-governed ecosystems. [12,21–24,32–38,41–44] Building them will require as much insight from physics, sociology, and institutional design as from machine learning itself—and it is precisely at these disciplinary boundaries that the next breakthroughs in collective intelligence are likely to appear.

References

1. Brown, T. B. *et al.* Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* 33, 1877–1901 (2020).
2. Kaplan, J. *et al.* Scaling laws for neural language models. *arXiv* 2001.08361 (2020).
3. Hoffmann, J. *et al.* Training compute-optimal large language models. *arXiv* 2203.15556 (2022).
4. Achiam, J. *et al.* GPT-4 technical report. *arXiv* 2303.08774 (2023).
5. Bubeck, S. *et al.* Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv* 2303.12712 (2023).
6. Touvron, H. *et al.* LLaMA: Open and efficient foundation language models. *arXiv* 2302.13971 (2023).
7. Mialon, G. *et al.* Augmented language models: A survey. *arXiv* 2302.07842 (2023).
8. Yao, S. *et al.* ReAct: Synergizing reasoning and acting in language models. *arXiv* 2210.03629 (2022).
9. Schick, T., Dwivedi-Yu, J., Ossa, A., Scarlatos, A. & Schütze, H. Toolformer: Language models can teach themselves to use tools. *Adv. Neural Inf. Process. Syst.* 36, 38822–38839 (2023).
10. Li, K. *et al.* CAMEL: Communicative agents for "mind" exploration of large scale language model society. *arXiv* 2303.17760 (2023).
11. Du, Y. *et al.* Improving factuality and reasoning in language models through multiagent debate. *arXiv* 2305.14325 (2023).
12. Malone, T. W. *Superminds: The surprising power of people and computers thinking together.* (Little, Brown and Company, 2018).
13. Shinn, N., Cassano, F. & Gopinath, A. Reflexion: Language agents with verbal reinforcement learning. *arXiv* 2303.11366 (2023).
14. Madaan, A. *et al.* Self-Refine: Iterative refinement with self-feedback. *Adv. Neural Inf. Process. Syst.* 36, 24608–24628 (2023).
15. Besta, M. *et al.* Graph of Thoughts: Solving elaborate problems with large language models. *arXiv* 2308.09687 (2023).
16. Pan, L. *et al.* A survey of large language model based agents: Architectures, tasks, and challenges. *arXiv* 2401.09498 (2024).
17. Mehandru, N. *et al.* Evaluating large language model agents in real clinics: Opportunities and challenges. *npj Digit. Med.* 7, 178 (2024).
18. Dong, Y. *et al.* Safeguarding large language models: A survey. *arXiv* 2407.10991 (2024).
19. Zheng, J. *et al.* Towards lifelong learning of large language models: A survey. *ACM Comput. Surv.* 57, 1–35 (2025).
20. Seshadri, A. *et al.* A survey of large language model agents for question answering. *arXiv* 2503.19213 (2025).
21. Wei, J. *et al.* Reasoning with language models. *Commun. ACM* 68, 46–57 (2025).
22. Ni, Y. *et al.* Large language models as agents. *Found. Trends Mach. Learn.* 18, 1–194 (2024).
23. OpenAI. OpenAI o1 system card. (OpenAI, 2024).
24. Chen, F. *et al.* Beyond scaling laws: Towards scientific reasoning-driven LLM architectures. *Preprints* 202504.2088 (2025).
25. Whitesides, G. M. The origins and the future of microfluidics. *Nature* 442, 368–373 (2006).
26. Wang, Z. & Zhao, Y.-P. Wetting and electrowetting on corrugated substrates. *Phys. Fluids* 29, 067101 (2017).
27. Wang, Z., Chen, E. & Zhao, Y.-P. The effect of surface anisotropy on contact angles and the characterization of elliptical cap droplets. *Sci. China Technol. Sci.* 61, 309–316 (2018).
28. Wang, Z., Lin, K. & Zhao, Y.-P. The effect of sharp solid edges on the droplet wettability. *J. Colloid Interface Sci.* 552, 563–571 (2019).

 doi:10.20944/preprints202511.1370.v1

29. Wang, Z.-L. & Lin, K. The multi-lobed rotation of droplets induced by interfacial reactions. *Phys. Fluids* 35, 021705 (2023).
30. Wang, Z. *et al.* Realization of self-rotating droplets based on liquid metal. *Adv. Mater. Interfaces* 8, 2001756 (2021).
31. Wang, Z. *et al.* Spontaneous motion and rotation of acid droplets on the surface of a liquid metal. *Langmuir* 37, 4370–4379 (2021).
32. Hu, J. & Wang, Z.-L. Crystallization morphology and self-assembly of polyacrylamide solutions during evaporation. *Fine Chem. Eng.* (2024).
33. Hu, J. & Wang, Z.-L. Inhibition of water vapor condensation by dipropylene glycol droplets on hydrophobic surfaces via vapor sink strategy. *Surf. Interfaces* (2024).
34. Wang, Z.-L. *et al.* Suppression of water vapor condensation by glycerol droplets on hydrophobic surfaces. *Phys. Fluids* 36, 067106 (2024).
35. Hu, J. & Wang, Z.-L. Effect of substrate temperature on the dry zone generated by the vapor sink effect. *Phys. Fluids* 36, 123104 (2024).
36. Hu, J. & Wang, Z.-L. Analysis of fluid flow in fractal microfluidic channels. *Phys. Fluids* 36, 093603 (2024).
37. Hu, J. & Wang, Z.-L. Effect of hygroscopic liquids on spatial control of vapor condensation patterns. *Surf. Interfaces* (2024).
38. Xu, Y. *et al.* Facet-dependent electrochemical behavior of Au–Pd core@shell nanorods for enhanced hydrogen peroxide sensing. *ACS Appl. Nano Mater.* 6, 18739–18747 (2023).
39. Zhuang, S. *et al.* Advances in solar-driven hygroscopic water harvesting. *Adv. Mater.* 33, 2001238 (2021).
40. Ni, F. *et al.* Tillandsia-inspired hygroscopic photothermal organogels for atmospheric water harvesting. *Adv. Funct. Mater.* 30, 2003268 (2020).
41. Li, Y., Naito, A. & Shirado, H. HiddenBench: Assessing collective reasoning in multi-agent LLMs via hidden profile tasks. *arXiv* 2505.11556 (2025).
42. Liang, P. *et al.* Holistic Evaluation of Language Models. *arXiv* 2211.09110 (2022).
43. Chan, C. M. *et al.* ChatEval: Towards better LLM-based evaluators through multi-agent debate. In *Proc. Int. Conf. Learn. Represent.* (ICLR) (2024).
44. Tam, T. Y. C. *et al.* A framework for human evaluation of large language models in healthcare derived from literature review. *npj Digit. Med.* 7, 159 (2024).