Jinyeop Song
PhD Candidate in Physics, MIT
yeopjin@mit.edu · (+1) 617-949-1042

Research Proposal : A Systems Biology Approach to Multi-LLM Agent Systems

Multi-LLM agent systems excel in real-world tasks—e.g., solver–refiner pipelines for math reasoning and coding [3], multi-agent retrieval-augmented generation (RAG) [6], and collaborative scientific tools [4]. In these systems, role-specialized agents tackle partial subtasks and, through orchestration, collectively solve the end task. For example, in a solver–refiner for math reasoning [3], a solver proposes candidate solutions, while a refiner selects and improves the most reliable answer.

However, current implementations rely heavily on large foundation models and static prompts, which lead to cost inefficiency and prompt-dependent capabilities that limit practical, scalable deployment. A natural alternative is multi-agent systems built with cost-efficient models, with each agent fine-tuned via SFT or RL for their roles. Training agents independently with verifiable or human-crafted rewards can work in simple settings, but it is brittle and does not scale to complex tasks or many agents. A more scalable approach is to jointly train the *network of agents* for end-to-end performance (e.g., quality of final answers). In this regime, however, the agents' learning is coupled and non-stationary, creating multi-body dynamics that make optimization challenging—an important yet underexplored problem.

Fortunately, **Systems Biology** offers a productive lens for addressing these challenges. It studies how interacting components—genes, neurons, or species—give rise to stable, functional behavior at the system level. I believe that knowledge and ideas in systems biology can transfer to multi-LLM-agent systems to address challenges and yield impactful research. In this spirit, I propose two projects: (1) applying a *network motif* lens to predict and control stability in multi-agent systems, and (2) *Pool-of-Agents*, a methodology inspired by population evolution that leverages agent pools to mitigate training instability and variance.

Project 1: Modeling Multi-Agent Training Dynamics via Network Motifs

Network motifs with predictable dynamics Feed-forward loops (FFLs) are recurring three-node regulatory patterns (termed *motifs*) found throughout bacterial and eukaryotic transcription networks [5]. In an FFL, three biological nodes—X (a gene or protein), Y, and Z—form two paths from X to Z: X directly regulates both Y and Z, and Y also regulates Z (Figure 1). Each node has a time-varying *expression level* (its activity), and regulation captures how changes in the expression levels of X and Y influence the expression level of Z. In *Coherent FFLs*, where both direct and indirect paths have the same net sign (e.g., C1-FFL: all activating with an AND gate at Z), the motif exhibits sign-sensitive delays and persistence detection, filtering transient noise and stabilizing the output signal. In *Incoherent FFLs*, where the paths have opposite signs (e.g., I1-FFL: X activates both Y and Z, but Y represses Z), the motif generates pulses and adaptation: Z responds rapidly via the direct path, then accumulation of Y suppresses it, creating transient responses that enable fold-change detection and dynamic-range compression. These architectural differences produce fundamentally distinct, yet predictable, dynamical regimes that are tunable via hyperparameters.

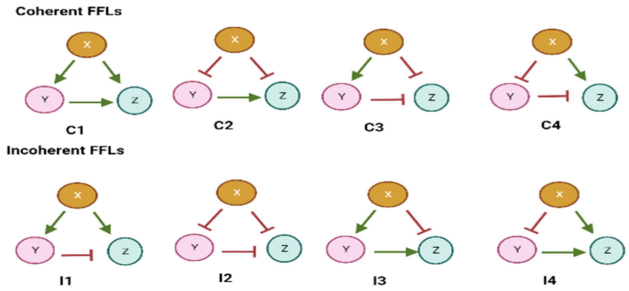

Figure 1: The eight types of feed-forward loops (FFLs). Coherent FFLs (C1–C4) have matching regulatory signs on both paths, enabling persistence detection. Incoherent FFLs (I1–I4) have opposite signs, generating pulse responses and adaptation. Green/red arrows denote activation/repression; Image from [12].

Multi-LLM-agent systems as regulatory networks. We map multi-LLM-agent systems to the dynamic regulatory network by treating each agent as a node whose ”expression level” is its task competence $A_i(t)$—the agent's learned performance for its assigned role at training step t. Edges between agents represent learning influences. A positive (activating) edge $X \to Y$ means that improving agent X's competence accelerates Y's learning—for example, a better retriever provides richer context that sharpens the reasoner's learning signal. A negative (repressing) edge $X \dashv Y$ means that improving X throttles Y's learning activity—for instance, a stricter critic gates the solver's exploration, reducing its update frequency. These influences can be implemented and tuned via dynamic budgets (token/tool quotas), shaped auxiliary credit assignment (interaction-based rewards), and gradient coupling (inter-agent regularization). Importantly, by treating multi-agent systems as regulatory networks, we can apply established motif-level principles (e.g., stability, adaptation, oscillation) to systematically predict, diagnose, and steer multi-agent training across different topologies and hyperparameters.

Research questions and investigation plan. The central question is: Can the *motif framework* explain the multi-agent training dynamics—specifically, stability, convergence speed, and adaptation? We hypothesize that C1-FFLs stabilize training by filtering spurious reward signals, whereas I1-FFLs accelerate early exploration via rapid transient responses and improve out-of-distribution adaptation, though at higher risk of oscillatory behavior under certain hyperparameter regimes. To test this, we explore simple tasks (e.g., RAG or math QA) where each agent's performance is reliably quantifiable, examining how motif types and hyperparameters predictably alter training dynamics. We then scale to open-ended tasks (e.g., SWE-bench) to assess whether these patterns hold in more realistic settings. Our goal is to derive design principles for architecting multi-agent systems with predictable and stable learning dynamics.

Project 2) Population-based Methods for Stable Multi-Agent Training

Instability of RL finetuning One major challenge of current RL-based LLM finetuning (e.g., PPO or GRPO) is inherent instability [7, 1]. Because policy updates continually shift the training distribution, RL finetuning becomes a moving target with high variance and poor convergence, leading to high costs in finding optimal performance. In multi-agent setup, this instability compounds dramatically. Each agent's policy updates simultaneously affect the training signals of all other agents, creating multiple interacting sources of instability. This can trigger cascading failures: one agent's improvement might destabilize another, leading to oscillating dynamics where agents trade off improvements. Ultimately, the system can become trapped in poor local maxima, failing to achieve its potential performance.

Pool-of-Agents training To address this instability, we draw inspiration from how population size shapes drift–selection dynamics in evolution. In large populations, selection is more efficient and genetic drift (stochastic noise) is weaker, producing more reproducible and convergent evolutionary dynamics [10, 8, 13] (Fig. 2). Adopting this principle, we propose a training method that maintains a *Pool-of-Agents* for each role, rather than training a single agent per role. Pool members perform identical roles but are initialized with slightly different policies via random perturbations. During each rollout, we sample one member from each pool. Individual agents can then be updated using standard gradient-based methods, or the pool can be optimized with evolution–strategy-based approaches [9]. We expect the following advantages: (1) reduced variance through ensemble averaging, (2) more robust exploration of the strategy space, and (3) resistance to cascading failures from individual agent updates. While maintaining agent pools increases memory and compute costs, these can be mitigated with LoRA [2] or parameter parallelism [9] for efficient deployment.

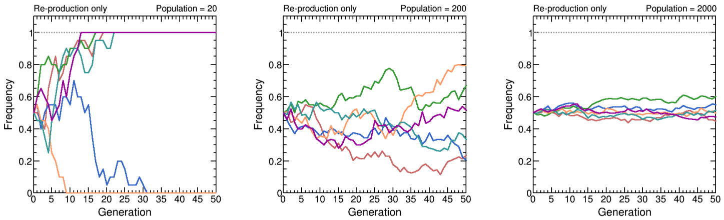

Figure 2: Effect of population size on evolutionary dynamics. With small populations (N=20), genetic drift dominates, causing high variance and rapid fixation. As population size increases (N=200, N=2000), drift weakens and allele frequencies stabilize, demonstrating more predictable evolutionary trajectories. Image from [11].

Research questions and investigation plan. The key question is whether *Pool-of-Agents* training yields more stable and robust optimization in end-to-end joint training of LLM agents. Specifically, we investigate (1) whether maintaining agent pools reduces training variance compared with single-agent baselines and (2) how population size affects the stability–compute trade-off. To address these questions, we will implement the *Pool-of-Agents* training framework and evaluate it across diverse multi-agent tasks (e.g., collaborative dialogue and compositional reasoning). We will compare against standard RL fine-tuning baselines (PPO, GRPO, etc.) and measure (a) variance across random seeds and (b) convergence rate and stability. We will also conduct ablation studies on population size to characterize the stability–compute trade-off. The end goal is to establish population-based training as a principled method for stable multi-LLM-agent optimization.

References

[1] Juntao Dai, Taiye Chen, Yaodong Yang, Qian Zheng, and Gang Pan. Mitigating reward over-optimization in rlhf via behavior-supported regularization. *arXiv preprint arXiv:2503.18130*, 2025.

[2] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[3] Yichen Huang and Lin F Yang. Winning gold at imo 2025 with a model-agnostic verification-and-refinement pipeline. *arXiv preprint arXiv:2507.15855*, 2025.

[4] Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

[5] Shmoolik Mangan and Uri Alon. Structure and function of the feed-forward loop network motif. *Proceedings of the National Academy of Sciences*, 100(21):11980–11985, 2003.

[6] Thang Nguyen, Peter Chin, and Yu-Wing Tai. Ma-rag: Multi-agent retrieval-augmented generation via collaborative chain-of-thought reasoning, 2025.

[7] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[8] Andrei Papkou, Chaitanya S Gokhale, Arne Traulsen, and Hinrich Schulenburg. Host–parasite coevolution: why changing population size matters. *Zoology*, 119(4):330–338, 2016.

[9] Xin Qiu, Yulu Gan, Conor F Hayes, Qiyao Liang, Elliot Meyerson, Babak Hodjat, and Risto Miikkulainen. Evolution strategies at scale: Llm fine-tuning beyond reinforcement learning. *arXiv preprint arXiv:2509.24372*, 2025.

[10] Christine Taylor, Drew Fudenberg, Akira Sasaki, and Martin A Nowak. Evolutionary game dynamics in finite populations. *Bulletin of mathematical biology*, 66(6):1621–1644, 2004.

[11] Jing Wang. Genetic drift. `https://boundino.github.io/S188592web/drift.html`, 2018. Final project for MIT Physics 8.592 (Statistical Physics in Biology), Spring 2018.

[12] Tsigereda Weldemichael, Michael Dare Asemoloye, and Mario Andrea Marchisio. Feedforward loops: evolutionary conserved network motifs redesigned for synthetic biology applications. *Applied Sciences*, 12(16):8292, 2022.

[13] Xin-Feng Zhao, Yi-Qi Hao, and Quan-Guo Zhang. Stability of a coevolving host-parasite system peaks at intermediate productivity. *PLoS One*, 12(1):e0168560, 2017.