

Peer Review

Review of: "AgentNet: Decentralized Evolutionary Coordination for LLM-based Multi-Agent Systems"

Evgenii Arsentev¹

1. Independent researcher

Summary

The paper proposes AgentNet, a multi-agent framework in which every LLM agent carries both a router and an executor, so that task allocation happens through local peer-to-peer decisions instead of a central orchestrator. Three mechanisms carry the design: a weighted directed graph whose edges are updated after each task and pruned below a threshold (Section 3.2, Eqs. 2 and 3); a per-agent RAG memory of local trajectory fragments, retrieved top- k at inference and pruned when the pool hits capacity (Section 3.3, Eq. 4); and a routing step where an agent may forward a task, split it, or execute it, with capability vectors updated from observed performance (Section 3.4). The empirical case rests on Table 1 (three backbones — DeepSeek-V3, GPT-4o-mini, Qwen-turbo — across MATH, BBH, and API-Bank, with nine baselines), Table 2 (heterogeneity on BBH), Table 3 (evolution ablation), and Figures 5, 8, 9, 10. The framing questions — can specialization emerge without a controller, and does decentralized routing buy anything — are good ones, and the routing/memory design is described in enough detail to be argued with. My problem is with what the experiments actually establish. The headline claims that the abstract makes loudest — fault tolerance, privacy preservation, scalability, efficiency — are the ones with no measurement behind them at all, and several accuracy numbers in the tables do not survive arithmetic checking.

Does the claim hold on the data

Partly, and not for the claims the paper leads with. Take them one at a time.

Accuracy. Section 4.2 (line following Table 1) says AgentNet "consistently outperforms centralized frameworks." On the GPT-4o-mini block of Table 1, it does not outperform them: AgentNet's MATH column reads 85.00, the same value as AFLOW and as GPTSwarm in the same block; AgentNet's BBH

column reads 86.00, the same value as GPTSwarm. That is a three-way tie and a two-way tie, not consistent superiority. Elsewhere, the margins are one or two test items. In the DeepSeek-V3 block, MATH: MetaGPT 92.14 vs. AgentNet 92.86 — on the 140-problem MATH test set defined in Section 4.1, that is 129 vs. 130 problems, a one-item difference. Same block, API-Bank: React 29.00 and Self-Consistency 29.00 vs. AgentNet 30.00 — one task out of 100. With a single pass at a single configuration and no confidence intervals, a one-item gap is not evidence of anything. The word "significantly" appears in the abstract, in Section 1 (final paragraph), and in Section 4.4 ("AgentNet significantly improves performance"), with no test anywhere in the paper.

Fault tolerance. This is claimed in the abstract, in the three bullets of Section 1, in Section 3.1, and again in the conclusion of Section 5.1 ("a high degree of fault tolerance"). There is no experiment in which an agent fails. Section 5.1 varies the agent count and executor pool size and reports accuracy; that measures neither failure recovery nor the absence of a single point of failure. A kill-one-agent (and kill-the-highest-degree-agent) experiment would be cheap and would directly support the paper's central architectural argument. As the paper stands, its most distinctive claim is entirely unmeasured.

Privacy. Claimed in the abstract, in Section 1 (the paragraph beginning "Moreover, each of these three innovations inherently enhances data privacy"), in Section 3.1, and in the conclusion. There is no threat model, no adversary, no leakage metric, no experiment. Section 3.4 offers one concrete mechanism — when an agent splits a task, it forwards only completed subtask results, not the decomposition reasoning — but subtask content itself is forwarded verbatim, and nothing quantifies what does or does not cross an organizational boundary. "Privacy-preserving" is doing a lot of work in the title's implicit promise and in the abstract; either measure it or downgrade it to a design intention.

Specialization. Section 5.2 concludes that agents specialize autonomously, reading off the "final ability scores of the agents after training" in Figure 9. Those ability scores are the system's own capability vectors, updated by the mechanism described at the end of Section 3.4 — the capability contribution is "calculated by analyzing the types of operations successfully executed by the agent and the quality of results," i.e., an LLM-driven self-assessment. Using the system's self-reported competence as the evidence that the system became competent is circular. The honest measurement is external: hold out per-category BBH items and report each agent's accuracy per category. That is a small amount of extra work and would turn Section 5.2 from an illustration into a result.

Compute budget. Table 1's caption says three agents were used for every multi-agent method "to ensure a fair comparison," which equalizes agent count but nothing else. AgentNet runs a multi-hop routing

pipeline with retrieval on top of a prior evolution phase over the training sets of Section 4.1 (700 MATH problems, 627 BBH problems, 100 API-Bank tasks). Direct is one call. Self-Consistency samples several paths, and Self-Refinement iterates. These sit in one table under one accuracy column with no token counts, no call counts, and no wall-clock time. Since the abstract claims improved "efficiency," and Section 3.2 claims the pruning mechanism optimizes "communication overhead," the absence of any cost measurement is a gap in the paper's own terms. Please add tokens-in/tokens-out and the number of LLM calls per solved task for every row of Table 1.

Table 2 does not support the reversal it is read as supporting. Section 4.3 concludes that heterogeneity hurts in small teams and helps in larger ones. The whole basis is eight numbers: 3 agents give 0.86 / 0.84 / 0.81 / 0.81, and 5 agents give 0.79 / 0.86 / 0.85 / 0.85 across Fully Homogeneous, Skill Hetero., LLM Hetero., Both Hetero. On a 100-item BBH test set, that is a 2–5 item spread in the 3-agent row, and the "reversal" hinges on a single cell — 5 agents / Fully Homogeneous at 0.79 — being 7 items below its 3-agent counterpart. One cell, one run, no interval. The conclusion in the paragraph after Table 2 is stated far more confidently than eight single-run points allow.

Reproducibility

Section 4.1 ("Parameter Configuration") does better than most: temperature 0.0, max tokens 2048, top-p 1.0, and BAAI/bge-large-en-v1.5 for memory similarity. That is a real start. What is missing:

No code or data release statement appears anywhere in the paper, including the Statements and Declarations section. No model snapshots — "GPT-4o-mini," "DeepSeek-V3," and "Qwen-turbo" in Table 1 are product names, not versions, and all three moved during 2024–2025; without dates, the table cannot be reproduced. No seeds and no number of repetitions: temperature 0.0 is not determinism for hosted MoE endpoints, so every number in Tables 1, 2, and 3 should be a mean over repeated runs with a spread. None of the framework's own hyperparameters have values: the decay factor α (Eq. 2), the pruning threshold θ_w (Eq. 3), the memory capacity $C_{\max}$, and retrieval depth k (Section 3.3), and the decay factor in the capability update (Section 3.4) are all defined symbolically and never instantiated. Section 3.4 refers to "an instruction set containing carefully crafted prompts" for capability inference; those prompts are not in the paper. Nothing states whether baselines received the same training sets as AgentNet's evolution phase — this matters most for Synapse, which needs exemplars, and for AFLOW, GPTSwarm, and MorphAgent, which optimize on training data. Nothing states that the training and test splits are disjoint; for API-Bank in particular, Section 4.1 says both the 100 training tasks and the 100 test tasks

were "randomly sampled from the full API-Bank corpus," which as written does not exclude overlap. One sentence fixes this, and without it, a reader has to wonder about leakage into the memory pools.

On the retrieved version: Table 1's bold and underline formatting described in its caption is lost in the copy I was given, so I have not verified which cells the authors marked as best and second-best; my reading of ties above is from the numeric values only. Figures 5, 8, 9, and 10 did not come through as readable images in this version, so I have assessed them only through the text of Sections 4.4, 5.1, 5.2, and 5.3 and their captions.

Specific points

1. **Table 3 contradicts Table 1 on the same configuration.** Table 3 is captioned "3 Agents ... Backbone: gpt-4o-mini" and gives AgentNet, API-Bank column, 32.00; Section 4.4 repeats it in prose ("performance improves notably from 23.00 to 32.00"). Table 1, GPT-4o-mini block, AgentNet row, API-Bank(Acc/%) column gives 29.00. Table 1's caption fixes three agents for all multi-agent methods, so these are the same setting and should be the same number. Note that 32.00 is exactly the value in Table 1's Qwen-turbo block, AgentNet row, API-Bank column — which makes a transcription slip the likely explanation. The other two columns are consistent (MATH 85.00 and BBH 86.00 match Table 1's GPT-4o-mini AgentNet row). Please recheck and correct; the evolution ablation's API-Bank gain is currently 9 points in Table 3 versus 6 points if Table 1 is the correct value.
2. **Two MATH percentages in Table 1 are not achievable on the stated test set.** Section 4.1 defines the MATH test set as 20 examples per type over 7 types, total 140 problems. Accuracy on 140 items can only take values $k/140$, i.e., steps of 0.714 — 87.86, 88.57, 91.43, 92.14, and so on. In the DeepSeek-V3 block of Table 1, MATH(Acc/%) Self-Consistency reads 88.00 and AFLOW reads 91.67. Neither is a multiple of $1/140$. (91.67 is $11/12$ exactly, which suggests a different denominator entirely.) Every other MATH cell in Table 1 that I checked does map cleanly onto $k/140$. Two possibilities, and the paper needs to say which: a transcription error, or those two rows were scored over a reduced set because some runs failed to produce a parseable answer and were dropped. If it is the latter, the drop rate has to be reported for every row — silently averaging over the runs that survived would change how the whole table reads.
3. **The MorphAgent baseline reports identical numbers across three different backbones.** In Table 1, the MorphAgent row (spelled that way in all three blocks; Section 4.1 and reference [27] spell it

MorphAgent) reads BBH 56.00 under DeepSeek-V3, 56.00 under GPT-4o-mini, and 56.00 under Qwen-turbo. The API-Bank column reads 16.00 under both DeepSeek-V3 and GPT-4o-mini. Three backbones whose Direct baselines differ by 12 BBH points (69.00, 59.00, 57.00 in the same table) landing on the same MorphAgent score to the hundredth is not plausible as an independent result. Either the baseline was run once and copied across blocks, or values were pasted in error. Since MorphAgent is the closest prior system to AgentNet — reference [27] is titled "Empowering Agents Through Self-Evolving Profiles and Decentralized Collaboration" — this is the single most important baseline in the paper to get right.

4. Self-Consistency at temperature 0.0 is a broken baseline. Section 4.1 describes Self-Consistency as sampling "multiple reasoning paths" and selecting by majority vote, citing Wang et al. [38], whose method requires sampling at non-zero temperature. Section 4.1's Parameter Configuration sets temperature 0.0 and top-p 1.0. If that setting applied to the baselines, the sampled paths were identical and the majority vote was a no-op, which would explain why Self-Consistency underperforms plain React in the GPT-4o-mini block of Table 1 on MATH (54.28 vs 55.71). If it did not apply to the baselines, then AgentNet and its baselines ran under different decoding settings and the comparison is unmatched. State explicitly which decoding parameters each method used; if Self-Consistency ran at $T=0$, it has to be rerun.

5. The "fully decentralized" claim is contradicted by the paper's own allocation procedure. Section 3.2 states that decision-making is distributed "without relying on a central authority or coordinator." But Section 3.4 says that for compound tasks the system "ranks all agents according to capability matching scores and selects the highest-scoring agent as the initial agent," and the forward operation is described as "evaluating the capability vectors of other agents in the network." Ranking all agents requires a global view of all capability vectors. Likewise, the weight matrix w_m in Eqs. 2 and 3 is described as maintained across tasks with a global pruning threshold, and the paper never says where it lives or how it is kept consistent under concurrent updates. This matters twice over: it weakens the decentralization claim, and it undercuts the privacy claim, since globally visible capability vectors are exactly the kind of proprietary-expertise metadata Section 1 says organizations want to protect. Please describe what each agent actually knows about the others, and how the weight matrix is replicated or partitioned.

6. The acyclicity argument in Section 3.4 runs backwards. The closing paragraph before Section 4 says that while a task "is only being forwarded from one agent to another, its state remains

unchanged, preserving the Directed Acyclic Graph (DAG) structure of the task routing path. This ensures that the task's progression avoids being trapped in an infinite loop." Unchanged state is precisely the condition under which a router can make the same decision again: if agent A forwards to B on the basis of the task state and B evaluates the same unchanged state, B forwarding back to A is not excluded by anything stated. The paper needs either a visited-set / hop-limit mechanism made explicit, or a short proof that the routing relation is acyclic. Reporting the observed path-length distribution and the rate of repeated visits from the BBH runs would also settle it empirically.

7. API-Bank accuracies are low enough across the board to suggest a scoring problem, and scoring is never defined. In Table 1's API-Bank(Acc/%) column, every one of the 30 cells falls between 9.00 and 32.00 — including Direct at 26.00 on DeepSeek-V3 and every strong single-agent baseline. The ordering is also strange: in the Qwen-turbo block, Direct (27.00) beats React (23.00), Synapse (24.00), and Self-Refinement (23.00). When a benchmark compresses every method into a 20-point band near the floor and inverts the usual ordering, the most likely explanation is the evaluation harness, not the methods. The paper never says what is being scored for API-Bank — exact match on the API call, correctness of arguments, or the final natural-language response. Define the metric and show a couple of scored examples.

8. Section 5.1's scalability conclusion is smaller than its own noise. The text reports training performance rising "from 80.38 for 3 agents and 30 executors to 81.18 for 9 agents and 40 executors" — 0.8 points over a threefold increase in agents — and then says testing "fluctuated between 80 and 86." The fluctuation band is roughly seven times the size of the trend, and on a 100-item test set spans six items. The paragraph concludes that AgentNet "can scale effectively"; what the numbers show is that tripling the agent count changes nothing measurable. Related: Section 4.4 reports 82.14% training accuracy for AgentNet's router on BBH, which sits above the entire 80.38–81.18 training range quoted in Section 5.1. The configurations may differ — Figure 8's caption notes "routers without pool limit" — but as written, a reader cannot reconcile the two, and the figure is not legible in the version I received. Please state the configuration behind the 82.14 figure.

9. Table 2's units contradict its caption. The caption reads "BBH Accuracy under Different Heterogeneity Settings (Acc%)" while the cells hold 0.79 through 0.86. Table 1 uses the percentage form (86.00) for the same benchmark. Pick one. Table 2 also never names the backbones used: Section 4.3 describes "LLM heterogeneity (different language models, same capabilities)" without saying which models were mixed, which makes that row impossible to interpret or reproduce.

10. **The React baseline is described and cited as Chain-of-Thought.** Section 4.1 defines React as "a prompting technique that elicits step-by-step reasoning from language models," citing Wei et al. [36] — which the reference list confirms is "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models" — alongside Yao et al. [32]. ReAct interleaves reasoning with tool-calling actions; CoT does not act at all. The distinction is decisive on API-Bank, where the whole task is tool use. State which one was actually run.
11. **Section 2.2's characterization of prior work conflicts with the paper's own baseline choice.** Section 2.2 says evolutionary approaches "still tend to operate within centralized control structures" and that existing systems "lack mechanisms for decentralized specialization and coordination." MorphAgent, reference [27], is titled "Empowering Agents Through Self-Evolving Profiles and Decentralized Collaboration" and is the paper's headline multi-agent baseline. The novelty claim needs to be stated against that system specifically — what AgentNet does that MorphAgent's decentralized collaboration does not — rather than against a general characterization that MorphAgent contradicts.
12. **Task labels and difficulty levels for API-Bank come from an unvalidated LLM annotator that is also a backbone under test.** Section 4.1 says the seven task types were assigned with GPT-4o-mini and that difficulty was "determined by prompt complexity and required toolchain length." No agreement rate against human labels is reported, and GPT-4o-mini is one of the three backbones in Table 1. Report inter-annotator agreement on a sample, or at minimum have a second model or a human check a subset.
13. **Minor, textual.** "MorphAgent" appears in all three blocks of Table 1 for what Section 4.1 and reference [27] call MorphAgent. The Figure 2 caption reads "Illutration" and ends with a double period. Section 3.3, second paragraph, has "reasoning and acing" for "acting." AFlow is "AFlow" in Section 2.2 and "AFLOW" in Section 4.1 and Table 1. Section 4.2's prose says MetaGPT reaches "only 53.00% accuracy on Logical QA," which is the GPT-4o-mini value; MetaGPT's BBH column reads 64.00 under DeepSeek-V3 and 51.00 under Qwen-turbo, so the sentence should name the backbone it is quoting.

Recommendation

Major revisions. The architecture is a legitimate contribution, and the routing/memory design is specified well enough to be built on, but three of the four properties the abstract advertises — fault tolerance, privacy preservation, efficiency — have no measurement behind them, the specialization

result is measured by the system's own self-assigned scores, and the accuracy tables contain a direct contradiction between Table 3 and Table 1 (API-Bank, GPT-4o-mini, AgentNet: 32.00 vs 29.00), two MATH values that cannot be produced on the stated 140-problem test set, and a baseline row repeating identical values across three different backbones. None of this is fatal to the idea; the numbers need auditing, the cost of each method needs reporting alongside its accuracy, the claims with no experiments behind them need either experiments or softer wording, and the reproducibility gaps (model snapshots, seeds, repeated runs, hyperparameter values, code) need closing before the paper can be read as evidence rather than as a proposal.

Declarations

Potential competing interests: No potential competing interests to declare.