EvoMAS: Evolutionary Generation of Multi-Agent Systems

ICML'26

https://github.com/amazon-science/EvoMAS

Yuntong Hu[1], Yuting Zhang[2], Matthew Trager[2], Yi Zhang[2], Shuo Yang[2], Wei Xia[2], Stefano Soatto[2]

[1]Emory University
Work done at AWS

[2]AWS

Background and Motivation

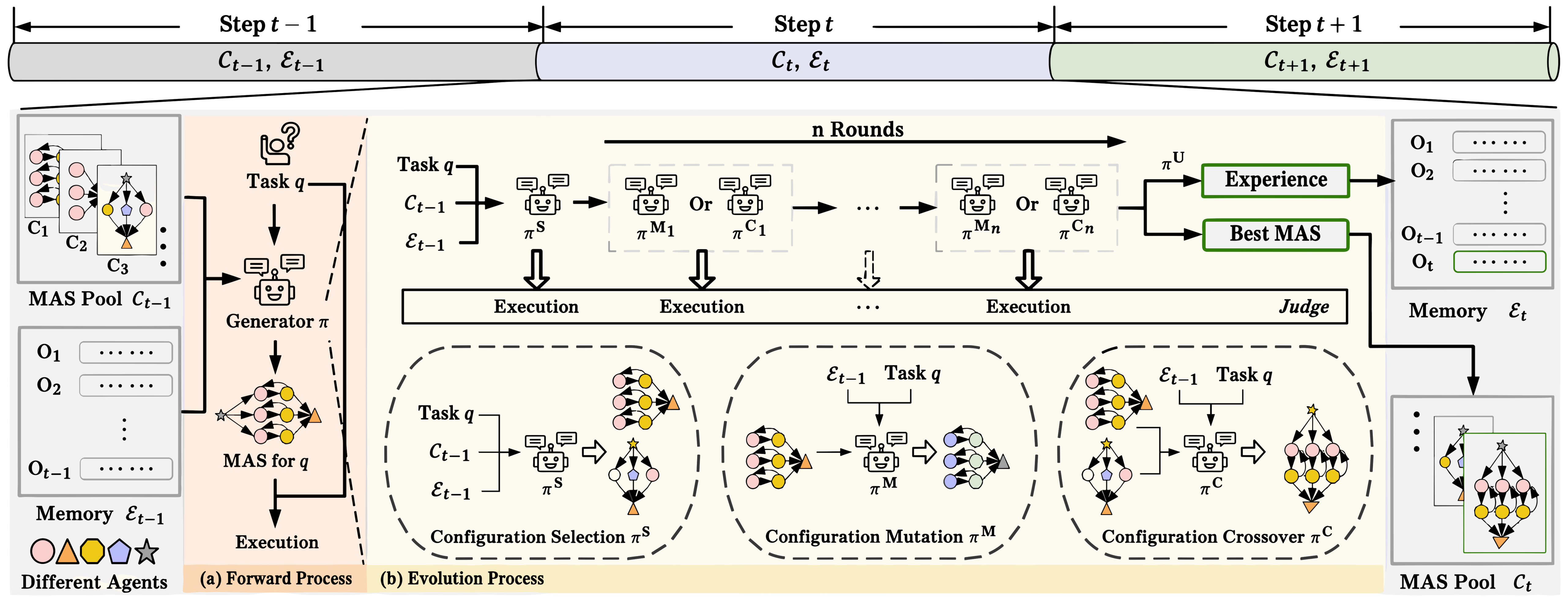
- **Problem**: MAS are effective but designing them is labor-intensive, brittle, and task-specific.
- **Gap**: Code-gen approaches are expressive but unreliable (29%-99% crash); template methods are reliable but suboptimal in performance.

✓ ***Tensions between expressive MAS generation vs. reliable execution.***

Configuration-based Structured MAS

- **Agents, their roles, prompts, models, tools**, and **topology** (Configurations are represented as declarative specifications and executed by a runtime interpreter, enabling expressive yet reliable MAS generation w/o code generation).
- Genetic algorithm inspired **structured evolution** is enabled with the configurations.

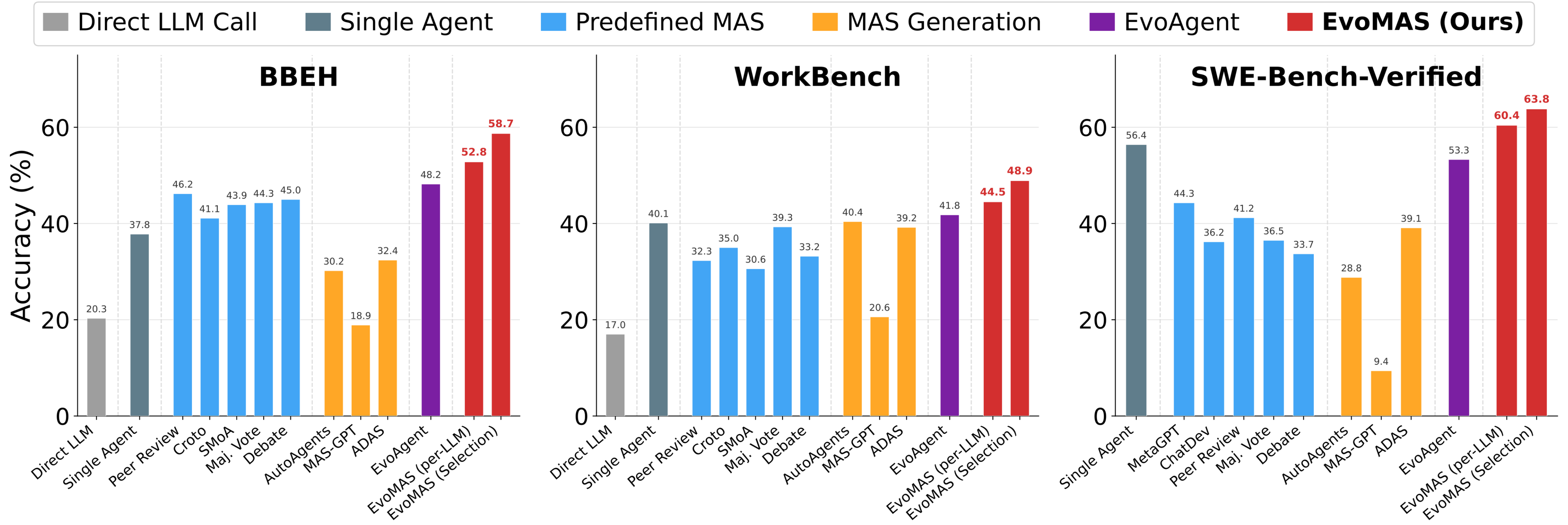
EvoMAS: An evolutionary framework for multi-agent system generation



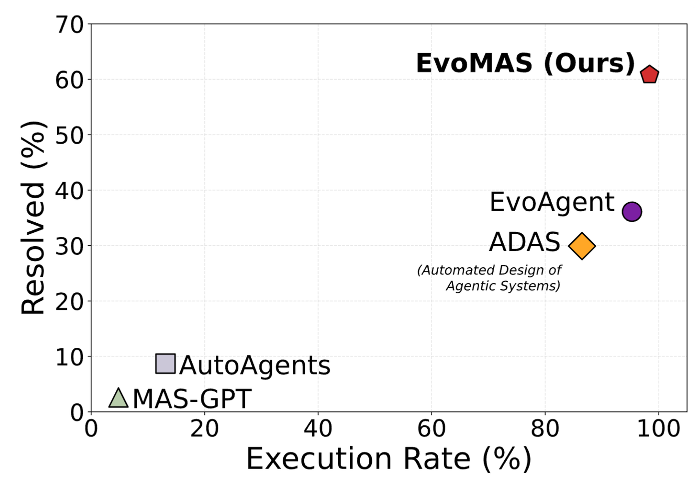
- ❖ **Pool:** seeded from general purpose MAS in literature (Peer Review, Majority Vote, Debate, etc.)
- ❖ **Meta-model:** an LLM that reads configs + execution traces; drives all evolutionary operators
- ❖ **Per-query evolution loop (T steps):**
  - ❖ **Select:** retrieve task-relevant configs from pool
  - ❖ **Mutate (80%):** analyze trace, identify bottleneck, modify one component per step
  - ❖ **Crossover (20%):** inherit topology from one parent; recombine agents from both
  - ❖ **Evaluate:** keep offspring if reward improves
- ❖ **After task: add best config to pool; consolidate traces into memory for cross-task transfer**
- ❖ **Reward:** LLM-judge scores correctness, completeness, reasoning quality (no ground-truth). Cost = tokens + latency. R = Quality – β·Cost

Main Experiments

- ❖ **Cross-Task Learning Paradigm**: Online testing setting; evolution happens in tandem with testing. Evolutionary gains benefit future tasks.
- ❖ **Accuracy:** EvoMAS outperforms all across reasoning (BBEH), tool-use (WorkBench), and software engineering benchmarks (SWE-Bench).
  - ☐ The figure shows the best result over three LLM models for agents: Qwen3-235B-A22B, Qwen3-Coder-480B, Claude-3.5-Sonnet.
  - ☐ Meta-model: Claude-4-Sonnet. EvoMAS (Selection) can select LLM for each agent via evolution.

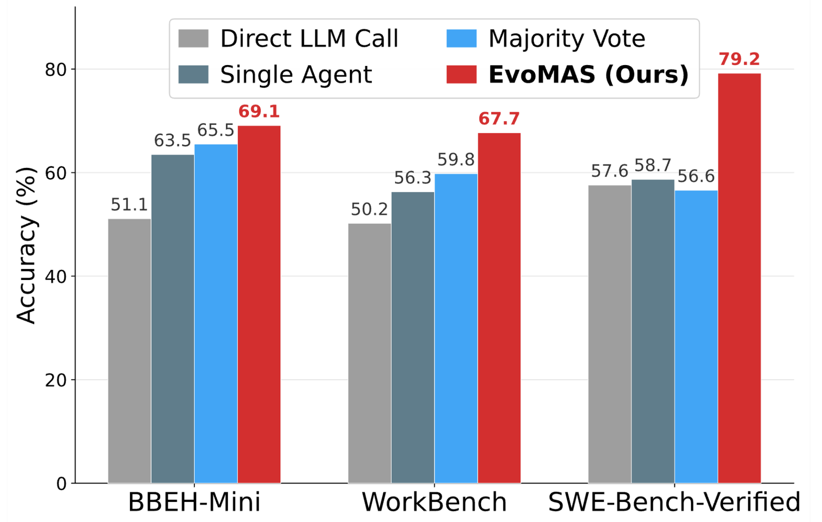


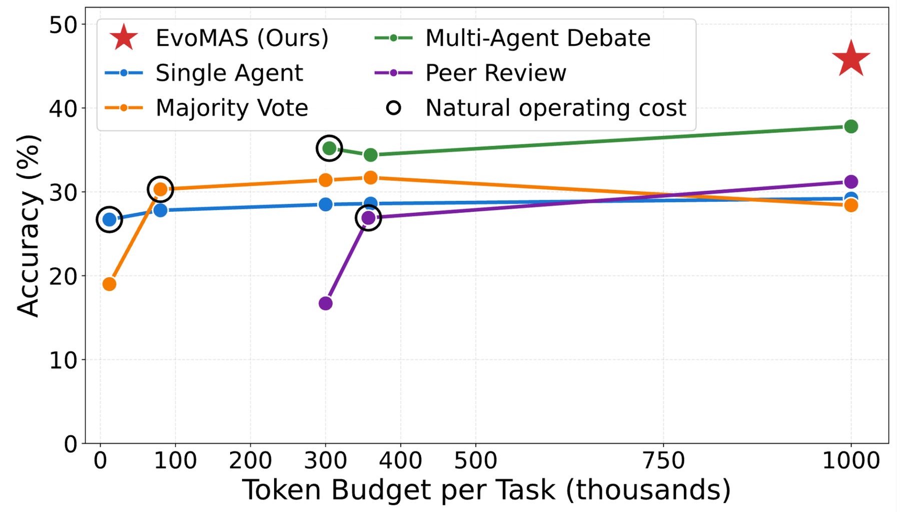
- ❖ **>98% MAS execution success**, >50% improvement over other MAS generation methods on SWE-Bench



- ❖ With Claude-Sonnet-4.5, EvoMAS achieves **SWE-Bench-Verified SOTA at 79.2%** resolved rate

Claude-Sonnet-4.5 for both the meta-model and agents



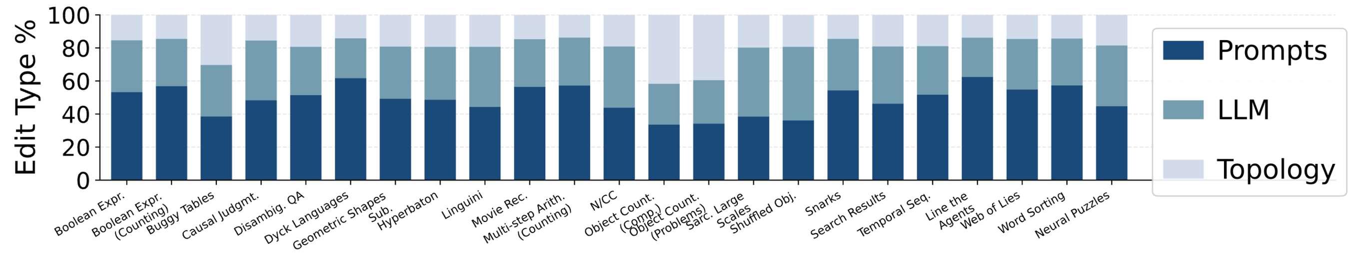


← **EvoMAS performs effective test-time compute scaling** when budgets allow.

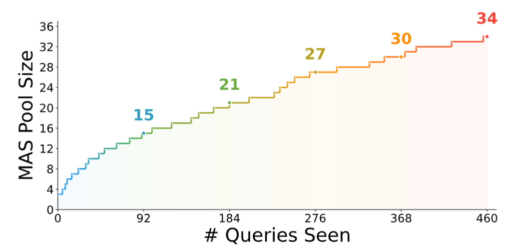
- ☐ Human-designed workflows quickly saturate, wasting available budgets.
- ☐ EvoMAS continues to improve with additional evolutionary search.

EvoMAS Behaviors and Robustness

- ❖ EvoMAS decides which configuration components to evolve.



- ❖ **Evolution trajectory patterns:** (BBEH-Mini)
  - ☐ Prompt refinement dominates early, followed by model selection and topology optimization, adapting to task requirements.
  - ☐ Mutation drives exploration; crossover becomes effective once specialized MASs emerge.



- ❖ **Consistent improving over different initial pools:** The proposed initial pool based on literature works the best; meanwhile, EvoMAS consistently improves over different initializations (+1.2~7.6pp) – curated, agent-generated, or empty.
- ❖ **Flexibility in selecting the meta-model:** Weakest meta-model (open-weight Qwen3-235B) achieves 58.2%, beating all baselines (best: 50.2%) by +8.0pp; 5.6pp below default meta-model (Claude-4-Sonnet: 63.8%).
- ❖ **LLM-judge is close to oracle:** LLM-judge near-oracle; oracle improves only +1.5 to +3.4pp across benchmarks; open-weight judge drops only 1.4-1.8pp.
- ❖ **Task-order insensitivity:** Accuracy std is only 1.1% across 11 random shuffles of the evaluation order on BBEH-Mini.
- ❖ **Accuracy breakdown:** On BBEH-Mini with Qwen3-235B, per-task evolution alone (40.4% vs Single Agent 36.6% and Peer Review 39.8%) -> cross-task memory (+3.3pp) -> cross-task MAS pool (+2.2pp = 49.1%).
- ❖ **Cost breakdown:** Meta-model overhead modest (7.7% tokens, 10.6% time); MAS execution dominates (55.4% tokens, 69.8% time).
- ❖ **Amortization & Transferability:** Evolution conducted on a limited number of tasks; later tasks amortize cost via direct pool selection without evolution (~65% tokens saved), enabling offline evolution and **transfer to new samples** with improvement.