[58] Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. 2021. MiniF2F: a cross-system benchmark for formal Olympiad-level mathematics. *ArXiv* abs/2109.00110 (2021).
[59] Aojun Zhou, Ke Wang, Zimu Lu, Weikang Shi, Sichun Luo, Zipeng Qin, Shaoqing Lu, Anya Jia, Linqi Song, Mingjie Zhan, et al. 2023. Solving Challenging Math Word Problems Using GPT-4 Code Interpreter with Code-based Self-Verification. *ArXiv* (2023).
[60] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. 2023. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. In *ICLR*.

A APPENDIX: EVOLUTION OF MATHAGENT

LLM-based agent originates from agent-based model (ABM) following a theoretical agent model, Belief-Desire-Intention model (BDI). BDI defines an agent as "based on the **belief** about the state of the world and the **desire** for an ideal state, an **intention** to achieve the ideal state is formed, which generates the **action** that interacts with the external environment, accepts the **response** given by the environment, and further updates the belief". Based on the definition, Figure 7(a) shows a graphical illustration of BDI. Recent LLM-based agents can be counted as instantiations of BDI. Figure 7 shows a general, practical LLM-based agent framework, which defines a specific interaction method with the environment - Tool Using.

The most significant difference between MathAgent and a general LLM-based agent is that MathAgent does not require an explicit environment to interact with but needs to complete reasoning within the agent, which is shown in Figure 7. Although MathAgent can also enhance the calculation abilities of the reasoner by calling tools, these external modules are not included in the agent. Therefore, we focus on the systematic exploration of MathAgent, and the use of external tools is no longer within the research scope.

B APPENDIX: PROMPTS IN TWO MATHAGENTS

We show all prompts in two MathAgents. As for MathAgent-M, we define Plan, Integrate, Summarize, and three kinds of actions, whose prompts are given in Table 6. Plan is a fundamental function to determine the present action with a JSON format. Integrate and Summarize are two auxiliary function to help the completion of the inference. Three actions, including Infer, Associate, and Observe, have been introduced in Section 3.3. All these actions are selected automatically with the help of Planner.

MathAgent-H contains Plan, Check, Summarize, and three kinds of actions, whose prompts are given in Table 8. Plan is a fundamental function to determine the present action with a JSON format. Check and Summarize are two functions of Reflector. Logical actions (Infer, Calculate, Disprove, Classify, and Induce) is used to perform compound logical reasoning, while mathematical actions (Associate, Construct) is specific to mathematical problems. Auxiliary actions (Analyze, Rethink, and Integrate) help the system to complete the inference. All actions have been introduced in Section 3.3. In the plan function, {action: Action Description} is predefined, which is shown in Table 7, whose definitions are consistent with the result in Section 2.1.

C APPENDIX: ALGORITHMS OF MATHAGENT SYSTEMS

We also provide the execution algorithms of the two systems. MathAgent-M is a model-aligned system whose almost all actions can be selected by Planner automatically. Algorithm 1 shows the execution process of MathAgent-M.

Algorithm 1: MathAgent-M Algorithm.

Input: Problem $(X_0, y_0) \in D$, Simulation Function with LLM f_{LLM}, Prompts $P = \{\text{plan} : P_{\text{pl}}, \text{stopcheck} : P_{\text{sc}}, \text{summarize} : P_{\text{sm}}, \text{infer} : P_{\text{inf}}, \text{associate} : P_{\text{ass}}, \text{observe} : P_{\text{obs}}\}$
Output: Proof/Result r

Initialize Memory: m_0 =NULL;
Initialize Stop Index: idx = False;
Count: $n = 0$;
while idx == *False* **do**
  Select Action: $a_n, m = f_{LLM}(X_n, y_n, m_n, P_{\text{pl}})$, $a_n \in \{\text{infer, associate, observe, summarize, stopcheck}\}$;
  Update Memory: $m_n \leftarrow m$;
  if $a_n \in \{\text{infer, associate, observe, summarize}\}$ **then**
    Execute Action: $X_{n+1} = f_{LLM}(X_n, P_A), A \in \{\text{inf, ass, obs, sm}\}$;
  else
    $idx, X_{n+1} = f_{LLM}(X_n, P_{\text{sc}})$;
  end
  if idx ==*True* **then**
    $r = X_{n+1}$
  end
  Count: $n \leftarrow n + 1$;
end
return r;

Based on Algorithm 1, MathAgent-M adopts a planning function to select actions in Reasoner and Reflector automatically. Afterward, the action is taken with a *Line* topology. These two steps are executed alternately until the inference is completed or terminated.

MathAgent-H is a human-aligned system whose actions are defined to align with humankind. For example, Induce executes mathematical induction by dividing the problem into two sub-tasks: initial condition verification and induction, which obey demonstrated logic. Algorithm 2 shows the execution process of MathAgent-H.

In Algorithm 2, MathAgent-H also adopts a planning function to select actions in Reasoner automatically. Afterward, the action is performed by Reasoner with diverse topologies in Executor. Finally, Reflector checks the inference proof for each step. These three steps are executed alternately until the inference is completed or terminated. Note that not all actions are selected by Planner automatically. Instead, several specific actions, such as Integrate, need to be performed at a fixed location, which is in line with human knowledge.

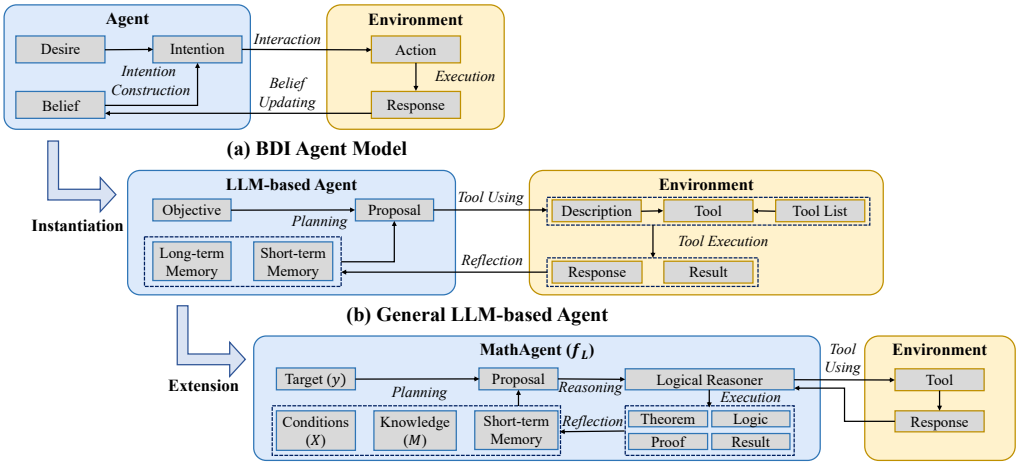

Figure 7: (a) shows the BDI agent model. (b) is a general, practical LLM-based agent framework. (c) illustrates MathAgent.

Table 6: Prompts in MathAgent-M

Function	Prompt
Plan	Choose an action that could be helpful for solving the problem. The outputs should be in the JSON format of "{'Action': X, 'task': E}", where E is the objective or guidance of the action.
Infer	Infer new rationales within the given context using deduction methods such as equation transformation, calculation, induction, and more. The outputs should be in the JSON format of "{'inferences': Is}", where Is should be in one sentence.
Associate	Seek associations within the given context to uncover valuable external knowledge. This knowledge may encompass theorems, lemmas, clever tricks, or any other insights not yet present in the context. The outputs should be in the JSON format of "{'associations': As}, where As should be in one sentence."
Observe	Analyze and discuss existing conditions, including generalization, negation, and reflective summarization. The outputs should be in the JSON format of "{'Observations': Obs}".
Summarize	Integrate intermediate steps into new premises and eliminate unnecessary or redundant references or rationales. The outputs should be in the JSON format of "{'new_premises': NPs}".
StopCheck	Summarize all generations into a final solution. The output should be in the JSON format of "{'status': ST, 'solution': Sol}". ST is either "solved" or "failed" and Sol is the summarized solution or the error analysis, specifically.

Table 7: Action Description in MathAgent-H.

Action	Description
disprove	'disprove' involves negation or counterproof.
calculate	'calculate' focus on calculation or formula derivation.
induce	'induce' is using mathematical induction.
classify	'classify' is using classification discussion with finite cases.
deduce	'deduce' is general text-based reasoning when other actions are not applicable.
analyze	'analyze' guides the exploration via discussion or analysis.
rethink	'rethink' means think outside the box.
associate	'associate' seeks applicable theorems and formulas.
construct	'construct' designs auxiliary conditions or variables.

Table 8: Prompts in MathAgent-H

Function	Prompt
Plan	You are an AI planner specialized in choosing an action and designing the task/guidance. Please read the given context and choose an action that could be helpful for solving the problem. Then a special AI agent who is only capable of one action will be called to finish it by completing the task or following the guidance. {Question Description} Please choose one action based on following instructions: {Action Description}
Deduce	You are an AI mathematician specialized in promoting the exploration and advance the reasoning/proving. Please read the given context and promote the proof. {Question Description} Remember, the only action you are capable of is defined as: {Action Description}
Calculate	You are an AI mathematician specialized in using calculation or formula derivation for promoting the exploration and advancing the reasoning/proving Please read the given context and promote the proof. {Question Description} Remember, the only action you are capable of is defined as: {Action Description}
Disprove	You are an AI planner specialized in solving mathematical problems by contradiction. Please read the given context and devise a contradiction scheme. {Question Description} Please design a contradiction scheme in the JSON list format of "{'Conditions': C, 'Goal': G}". C is all conditions assumed in proof by contradiction (including the necessary original conditions), and G is the target that is intended to be disproved in the proof by contradiction.
Classify	You are an AI planner specialized in devising a classification discussion scheme to solve math problems. Please read the given context and devise a classification discussion scheme. Question Description Please divide the problem into some subproblems in the JSON list format of "[{'Conditions': ST, 'Goal': SG}, and so forth.]". ST is the new conditions of one subproblem and SG is the target of it.
Induce	You are an AI planner specialized in devising a scheme with mathematical induction method to solve math problems. Please read the given context and devise an induction scheme. Question Description Please divide the problem into two subproblems in the JSON list format of "[{'Type': "base step", 'Conditions': C1, 'Goal': G1}, {'Type': "induction step", 'Conditions': C2, 'Goal': G2}]". C1 and C2 are the new conditions of the base step and the induction step, respectively. G1 and G2 are the targets of the two steps.
Associate	You are an AI mathematician specialized in Seeking (external) applicable theorems and formulas to aid or start an exploration. Please read the given context and promote the proof. {Question Description} Remember, the only action you are capable of is defined as: {Action Description}
Construct	You are an AI mathematician specialized in constructing auxiliary conditions/variables to aid or start an exploration. Please read the given context and promote the proof. {Question Description} Remember, the only action you are capable of is defined as: {Action Description}
Analyze	You are an AI mathematician specialized in providing an analysis/discussion for further decision-making. Please read the given context and promote the proof. {Question Description} Remember, the only action you are capable of is defined as: {Action Description}
Rethink	You are an AI mathematician specialized in thinking outside the box or finding useful patterns for further decision-making. Please read the given context and promote the proof. {Question Description} Remember, the only action you are capable of is defined as: {Action Description}
Integrate	You are an AI mathematician who is good at summarizing the proof {Question Description} You need summarize the proof. Pay attention: the summary should be shorter and clearer in three sentences or less, and you don't need to judge the correctness of the proof.
Check	You are an AI mathematician who is good at checking proofs and summarizing them. Please read the given context and make your judgement. {Question Description} You need to check whether the proof processing is right. If you believe the proof is wrong, please output in the JSON format: "{"Correctness": "wrong", "Summary": R}". R is the reason why you think the proof is wrong, and should be shorter and clearer in three sentences or less. Otherwise, if you think it's right, please output in the JSON format: "{"Correctness": "right"}"
Summarize	You are an AI mathematician who is good at summarizing the proof Question Description You need to check whether the final target is solved. Please output in the JSON format: You need to check whether the final target is solved. Please output in the JSON format: "{"Solved":S1,"Summary":S2}". S1 is yes/no whether the final target is solved, S2 is your summary of this proof. Pay attention: S2 should be shorter and clearer in three sentences or less, and you don't need to judge the correctness of the proof.
StopCheck	You are an AI mathematician who is good at summarizing the proof {Question Description} You need to check whether the final target is solved. Please output in the JSON format: You need to check whether the final target is solved. Please output in the JSON format: "{"Solved":S1,"Summary":S2}". S1 is yes/no whether the final target is solved, S2 is your summary of this proof. Pay attention: S2 should be shorter and clearer in three sentences or less, and you don't need to judge the correctness of the proof.

Algorithm 2: MathAgent-H Algorithm.

Input: Problem $(X_0, y_0) \in D$, Simulation Function with LLM f_{LLM}, Prompts
$P = \{\text{preprocess} : P_{\text{pp}}, \text{plan} : P_{\text{pl}}, \text{summarize} : P_{\text{sm}}, \text{check} : P_{\text{ck}}, \text{stopcheck} : P_{\text{sc}}, \text{deduce} : P_{\text{ded}}, \text{calculate} : P_{\text{cal}}, \text{disprove} : P_{\text{dis}}, \text{classify} : P_{\text{cls}}, \text{induce} : P_{\text{ind}}, \text{associate} : P_{\text{ass}}, \text{construct} : P_{\text{con}}, \text{analyze} : P_{\text{alz}}, \text{rethink} : P_{\text{rtk}}, \text{integrate} : P_{\text{int}}\}$

Output: Proof/Result r

Split Problem: $X_0, y_0 = f_{LLM}((X_0, y_0), P_{\text{pp}})$
Initialize Memory: m_0 =NULL;
Initialize Stop Index: idx = False;
Count: $n = 0$;
while idx == *False* **do**
  Select Action: $a_n, m = f_{LLM}(X_n, y_n, m_n, P_{\text{pl}})$, $a_n \in \{\text{deduce, calculate, disprove, classify, induce, associate, construct, analyze, rethink}\}$;
  Update Memory: $m_n \leftarrow m$;
  if a_n == disprove **then**
    Prove by Contradiction: $X'_n, y'_n, m = f_{LLM}(X_n, y_n, m_n, P_{\text{dis}})$;
    Update Memory: $m_n \leftarrow m$;
  else
    if $a_n \in \{\text{Classify, Induce}\}$ **then**
      Classify: $\{(X_n^i, y_n^i) | i = 1, 2, \cdots, k\} = f_{LLM}(X_n, y_n, m_n, P_A), A \in \{\text{cls, ind}\}$;
      Update Memory: $m_n \leftarrow m$;
      for $i = 1$ to k **do**
        Recursive Calculation: $(X_n'^i, y_n'^i) = \text{Self}(X_n^i, y_n^i)$;
      end
      Integrate: $X'_n, y'_n, m = f_{LLM}(\{(X_n'^i, y_n'^i) | i = 1, 2, \cdots, k\}, y_n, m_n, P_{\text{int}})$;
      Update Memory: $m_n \leftarrow m$;
    else
      Execute Other Action: $X'_n, m = X_n \cup f_{LLM}(X_n, m_n, P_A)$, $y'_n = y_n, A \in \{\text{ded, cal, ass, con, alz, rtk}\}$;
      Update Memory: $m_n \leftarrow m$;
    end
  end
  Check: $idx_c, m = f_{LLM}(X'_n, y'_n, X_n, y_n, m_n, P_{\text{ck}})$;
  if idx_c == *True* **then**
    $X_{n+1}, y_{n+1} = f_{LLM}(X'_n, y'_n, P_{\text{sm}})$;
  else
    $X_{n+1}, y_{n+1} = X_n, y_n$;
  end
  Update Memory: $m_{n+1} \leftarrow m$;
  Stop Check: $idx = f_{LLM}(X_{n+1}, y_{n+1}, P_{\text{sc}})$;
  if idx == *True* **then**
    $r = f_{LLM}(X_{n+1}, y_{n+1}, P_{\text{sm}})$;
  end
  Count: $n \leftarrow n + 1$;
end
return r;