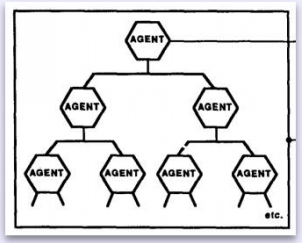
Finding the Scaling Laws of *Agents*

Guohao Li
Founder and CEO @ Eigent.AI

CAMEL-AI.org

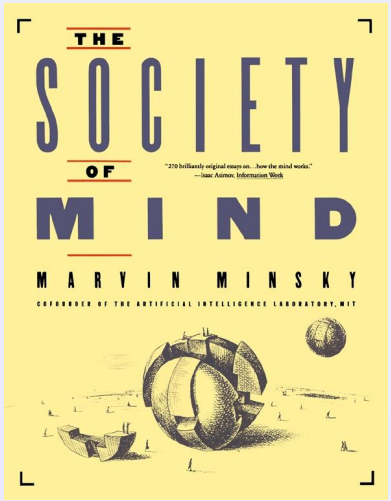
Agent from 1986

- *Agents* are mindless processes
- *Agent* by itself can only do some simple things
- Joining these *agents* in *societies* leads to true *intelligence*



What magical trick makes us intelligent? The trick is that there is no trick. The power of intelligence stems from our vast diversity, not from any single, perfect principle.

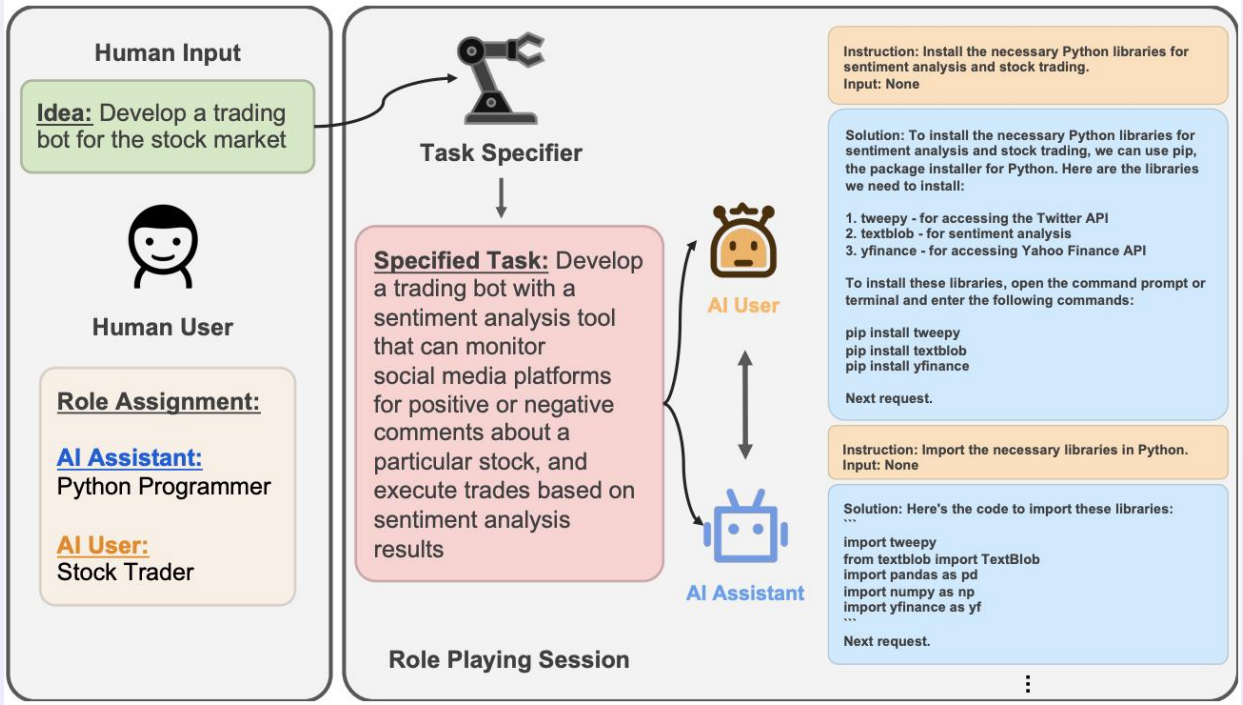
—Marvin Minsky, The Society of Mind, p. 308



CAMEL

CAMEL-AI

- Idea
- Role assignment
- Task agents
- Chat agents



CAMEL Role-Playing Framework (The First LLM multi-agent framework, released on Mar 2023)
CAMEL (NeurIPS 2023): https://arxiv.org/abs/2303.17760

Instruction-following Conversations

CAMEL-AI

```
            pygame.quit()
            sys.exit()
        elif event.type == pygame.MOUSEBUTTONDOWN:
            for i, character_rect in enumerate(character_selection_rects):
                if character_rect.collidepoint(event.pos):
                    selected_character = i
                    break

    # Display the background image
    screen.blit(background_image, (0, 0))

    # Display the character selection screen
    screen.blit(character_selection_text, character_selection_text_rect)
    for i, character_image in enumerate(character_images):
        screen.blit(character_image, character_selection_rects[i])

    # Update the display
    pygame.display.update()

    # Exit the character selection screen if a character is selected
    try:
        selected_character
        break
    except NameError:
        pass

# Load the selected character image
selected_character_image = character_images[selected_character]

# Display the selected character image
while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()

    # Display the background image
```

Create a treasure hunt game: Gamer (AI User) & Python Programmer (AI Assistant)
(Asset Figures are generated by Stable Diffusion)

More Agents Between than one?

CAMEL-AI

- CAMEL agents are better one agent > 70% on 200 tasks
- GPT-4 evaluation aligns with Human evaluation

	Draw	gpt-3.5-turbo Wins	CAMEL Agents Win
Human Evaluation	13.3%	10.4%	**76.3%**
GPT4 Evaluation	4.0%	23.0%	**73.0%**

Agent Evaluation Results

Language Models as Agents in CAMEL

CAMEL-AI

```python
class ChatAgent(BaseAgent):
    def __init__(self,
                 system_message,      # System message
                 model,               # Model backend
                 memory,              # Memory management
                 tools):              # Available tools
        self.model_backend = model
        self.memory = memory
        self.tools = tools
        ...  # Additional initialization

    def step(self,
             input_message,          # Main chat interface
             response_format):       # Response format config
        # Conversation loop
        while True:
            self.update_memory(input_message)            # Update conversation memory
            messages = self.memory.get_context()         # Get all messages from memory
            response = self.model_backend.run(messages)  # Generate response
            tool_call = self._extract_tool_call(response)  # Extract tool call information
            ...  # Additional processing

            if tool_call:
                response = self._step_tool_call_and_update(response)  # Handle tool call
                ...  # Additional tool handling
            else:
                break  # Exit loop for normal response

        # Return messages, status, and session info
        return ChatAgentResponse(
            output_messages=response,
            status=self.memory.terminated,
            info=self.get_info()
        )
```

Key Features:

- Memory: *Manages chat history and context window*

- Tools: *Supports both internal and external function calls*

- Step Loop: *Handle task require multiple request with one step*

OASIS: Simulate Social Media with 1 million Agents

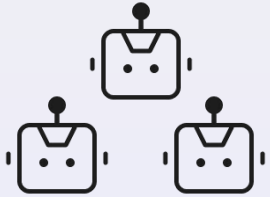
CAMEL-AI

- Up to 1 million agents
- Replicate social science experiments
- Explore dynamic of agent society

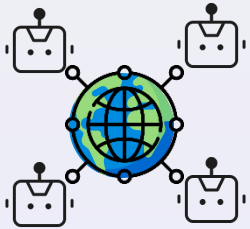
OASIS: Open Agent Social Interaction Simulations with One Million Agents (NeurIPS 2024 OWA workshop): https://arxiv.org/abs/2411.11581

OASIS: Simulate Social Media with 1 million Agents

CAMEL-AI



A few of agents



Millions of Agents

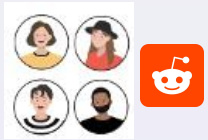
- How to simulate agent society and what is the new interaction scenario?
  - We can not simply set millions of agents talk with each other like group chat.
  - What can connects the globe, linking hundreds of millions of people or agents?
    - **Social Media**

X (Twitter) Reddit WeChat Weibo .....

OASIS: Herd Effect in Reddit

CAMEL-AI

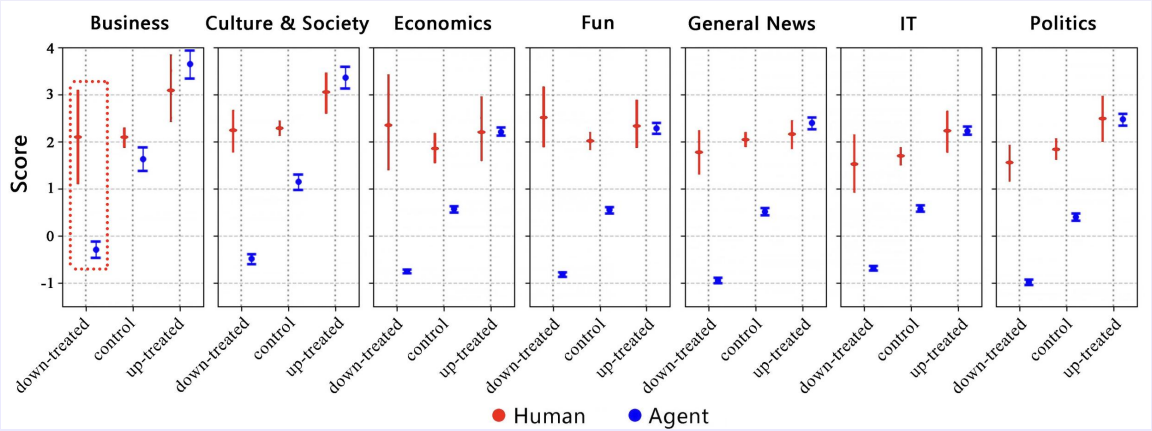
Posts



Users in Reddit

OASIS → Herd Effect!

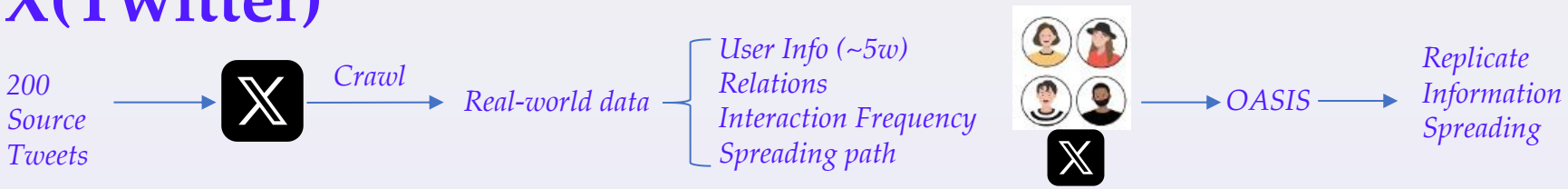
A downvote increases human' upvote chances, but the agent follows the downvote trend.

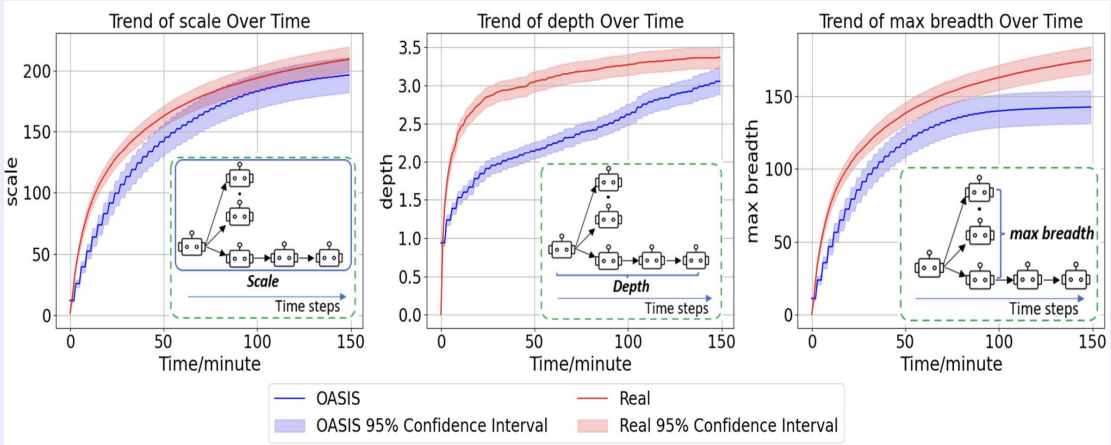


Agents are more prone to herd behavior than human.

OASIS: Information Propagation in X(Twitter)

CAMEL-AI





OASIS aligns well in terms of scale and max breadth, but its maximum propagation depth is relatively limited.

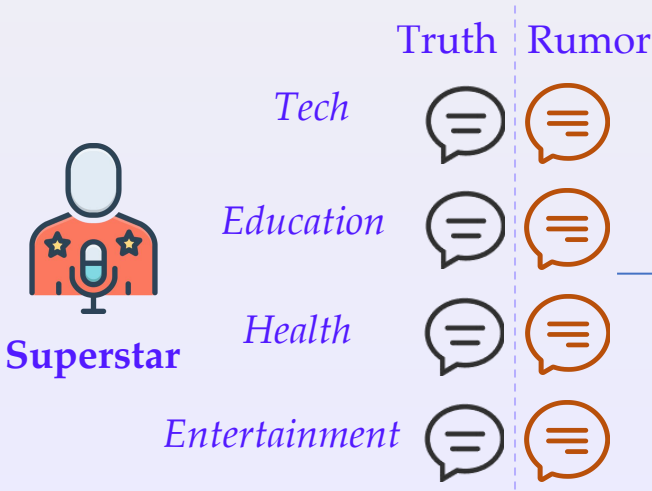
OASIS: Information Propagation in X(Twitter)

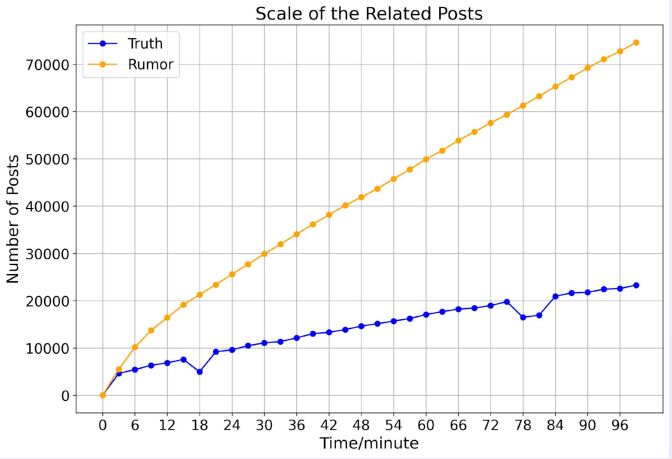
CAMEL-AI

Posts about Rumor

Posts about Truth

OASIS





1 million users

Misinformation spreads faster than truth

OASIS as an Environment

CAMEL-AI

- 1. Choose the action space, load the database and `oasis.make()` to create the social media environment
- 2. `env.reset()` to start the simulation
- 3. Simulate agent actions and `env.step(action)` to let agents post, like, and interact
- 4. `env.close()` to end the simulation

```python
import asyncio
import os

from camel.models import ModelFactory
from camel.types import ModelPlatformType, ModelType

import oasis
from oasis import ActionType, EnvAction, SingleAction


async def main():
    openai_model = ModelFactory.create(
        model_platform=ModelPlatformType.OPENAI,
        model_type=ModelType.GPT_4O_MINI,
    )

    # Define the available actions for the agents
    available_actions = [
        ActionType.CREATE_POST,
        ActionType.LIKE_POST,
        ActionType.REPOST,
        ActionType.FOLLOW,
        ActionType.DO_NOTHING,
        ActionType.QUOTE_POST,
    ]

    # Define the path to the database
    db_path = "./data/twitter_simulation.db"

    # Delete the old database
    if os.path.exists(db_path):
        os.remove(db_path)

    # Make the environment
    env = oasis.make(
        platform=oasis.DefaultPlatformType.TWITTER,
        database_path=db_path,
        agent_profile_path=("data/twitter_dataset/anonymous_topic_200_1h/"
                            "False_Business_0.csv"),
        agent_models=openai_model,
        available_actions=available_actions,
    )
```

```python

    # Run the environment
    await env.reset()

    action_1 = SingleAction(agent_id=0,
                            action=ActionType.CREATE_POST,
                            args={"content": "Earth is flat."})
    env_actions_1 = EnvAction(
        # Activate 5 agents with id 1, 3, 5, 7, 9
        activate_agents=[1, 3, 5, 7, 9],
        intervention=[action_1])

    action_2 = SingleAction(agent_id=1,
                            action=ActionType.CREATE_POST,
                            args={"content": "Earth is not flat."})
    env_actions_2 = EnvAction(activate_agents=[2, 4, 6, 8, 10],
                              intervention=[action_2])

    empty_action = EnvAction()  # Means activate all agents and no intervention

    all_env_actions = [
        env_actions_1,
        env_actions_2,
        empty_action,
    ]

    # Simulate 3 timesteps
    for i in range(3):
        env_actions = all_env_actions[i]
        # Perform the actions
        await env.step(env_actions)

    # Close the environment
    await env.close()


if __name__ == "__main__":
    asyncio.run(main())
```

Matrix - A Social Simulation Product Built on OASIS

CAMEL-AI

https://matrix.eigent.ai/x

CRAB Agents and Environments

CAMEL-AI

- Cross-environment task performing and benchmark system
- Automatic task generation
- Fine-grained graph evaluator
- Multimodal
- Reproducible virtual machine environment

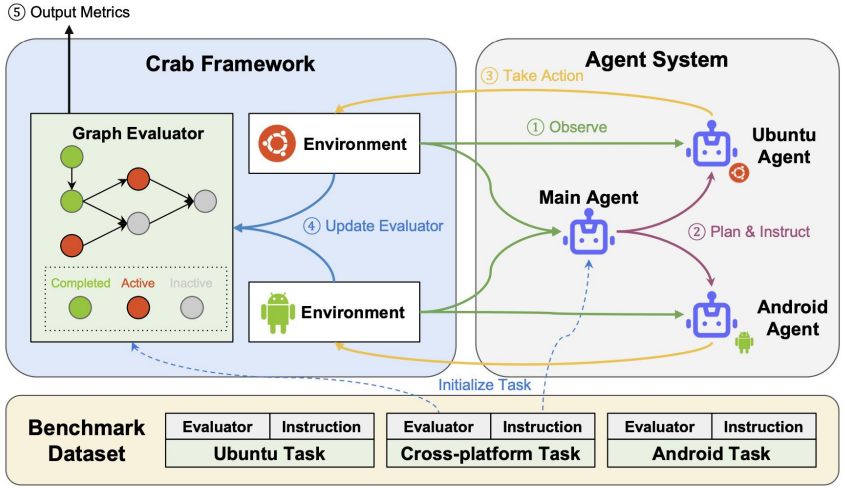
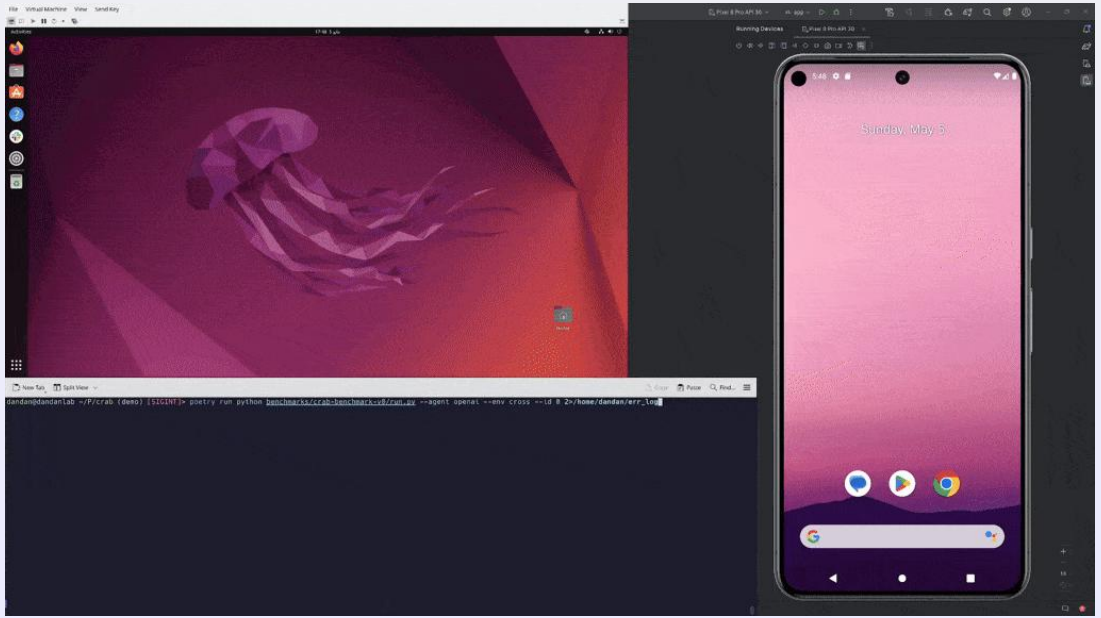


Figure 1: **Architecture of the Crab Framework demonstrating a benchmarking workflow for a multi-agent system.** A task is initialized by assigning instructions to the main agent and a graph evaluator inside the benchmark system. The workflow progresses through a cycle where the main agent observes, plans, and instructs the sub-agents, who then execute actions within their respective environments. The graph evaluator monitors the status of tasks within the environments, continuously updating and outputting the task completion metrics throughout the workflow.

CRAB Agents and Environments

CAMEL-AI

Task: *Open `slack`, navigate to `multi-modal-benchmark` channel, summarize the last two messages, and then send a message to the summary to the first contact on the phone*



Crab: Cross-environment Agent Benchmark for Multimodal Language Model Agents

CRAB Agents and Environments

CAMEL-AI

Crab benchmark system mainly consists of five types of component:

- `Action`: The fundamental building block of Crab framework, which represents a unit operation that can be taken by an agent or as a fixed process that called multi times in a benchmark.
- `Evaluator`: A specific type of `Action` that assess whether an agent has achieved its goal. Multiple evaluators can be combined together as a graph to enable complex evaluation.
- `Environment` A abstraction of an environment that the agent can take action and obverse in a given action and observation space. An environment can be launched on the local machine, a physical remote machine, or a virtual machine.
- `Task`: A task with a natural language description to instruct the agent to perform. It can include interaction with multiple environments. Notice that in the benchmark, a task should have an graph evaluator to judge if the task progress.
- `Benchmark`: The main body of the crab system that contains all required component to build a benchmark, including environments, tasks, prompting method. It controls several

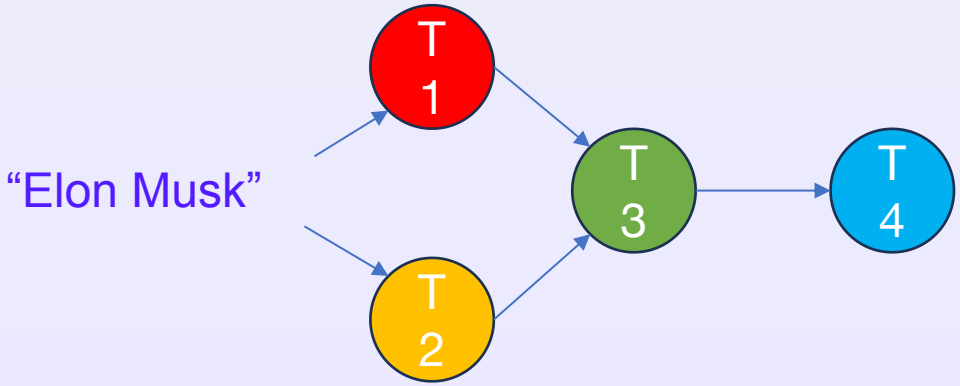
Crab: Cross-environment Agent Benchmark for Multimodal Language Model Agents

CRAB Task Generation from Graphs

CAMEL-AI

- Subtask 1: Download a {person_name} image -> image
- Subtask 2: Search {person_name} on google and write a summary about it. -> text
- Subtask 3: Make a poster based on {image/text} by PhotoShop -> image
- Subtask 4: Post {image} on reddit -> None

- Use GPT4 to combine description according to the graph.
- Generated task instruction: Download an image of Elon Musk, research and summarize information about him, create an informative poster in Photoshop using the image and summary, and post the poster on Reddit.



OWL Agents

CAMEL-AI

OWL | System Architecture

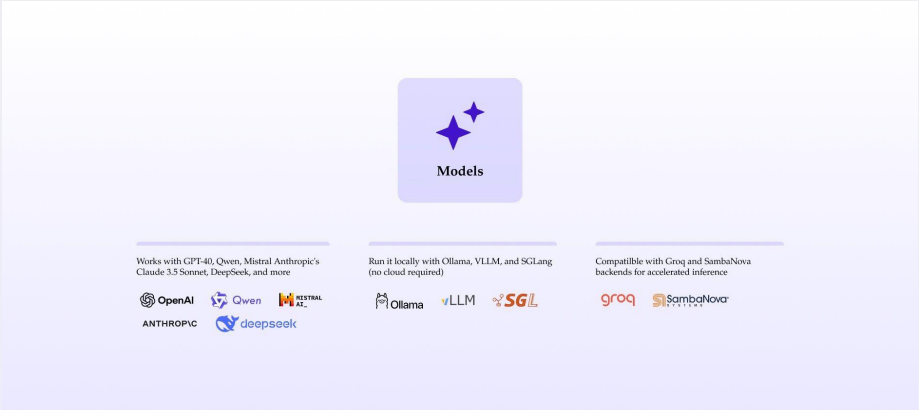
https://github.com/camel-ai/owl

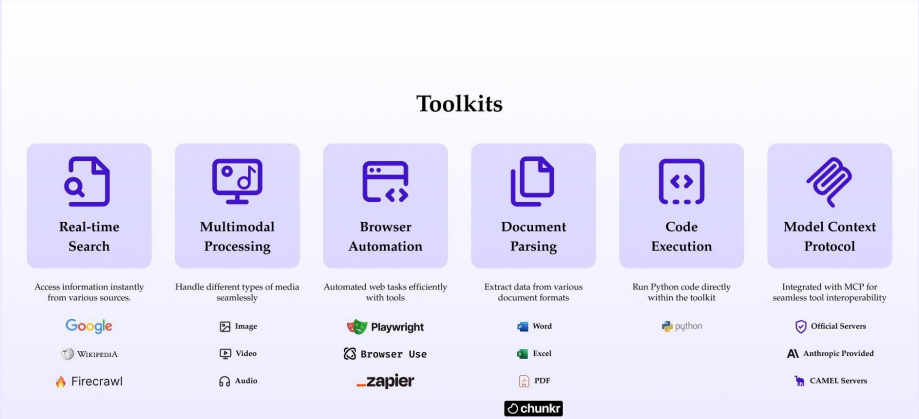
Optimized Workforce Learning for General Multi-Agent Assistance in Real-World Task Automation

CAMEL-AI

OWL Agents

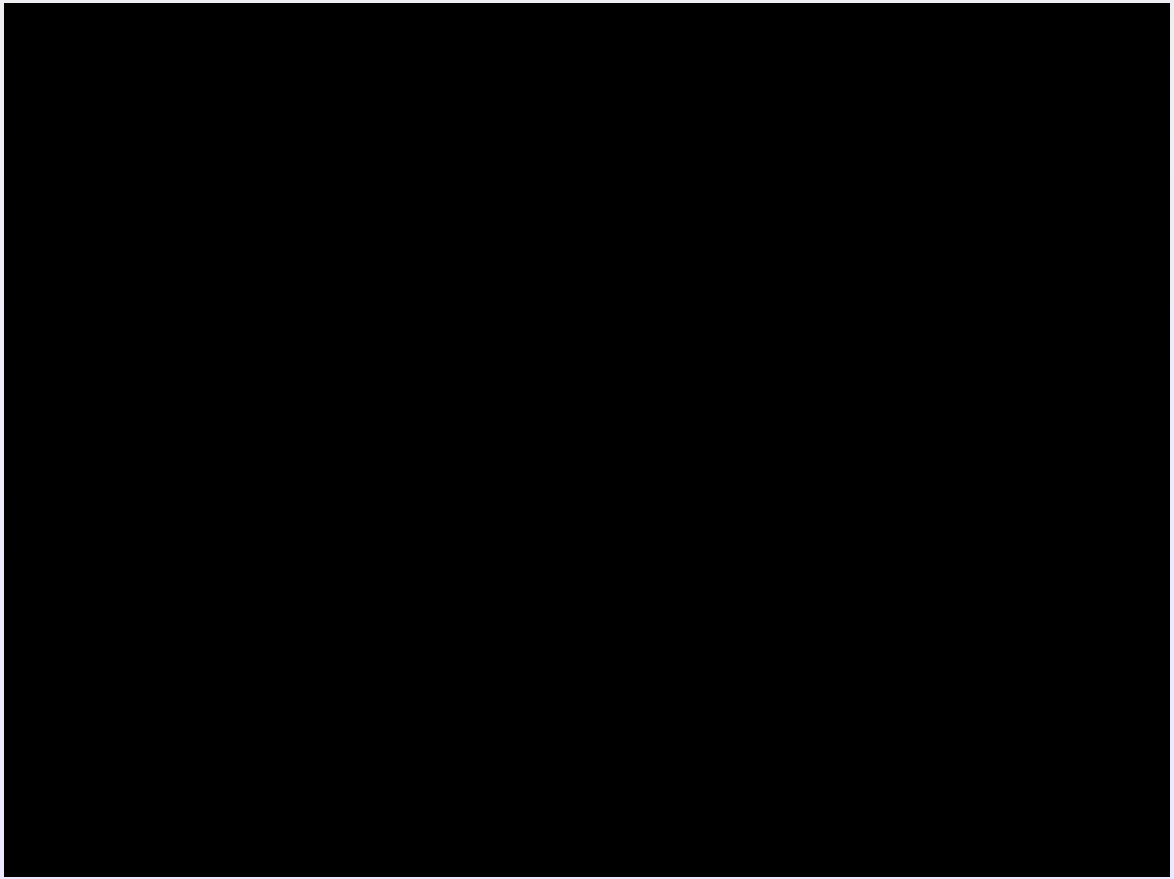
CAMEL-AI





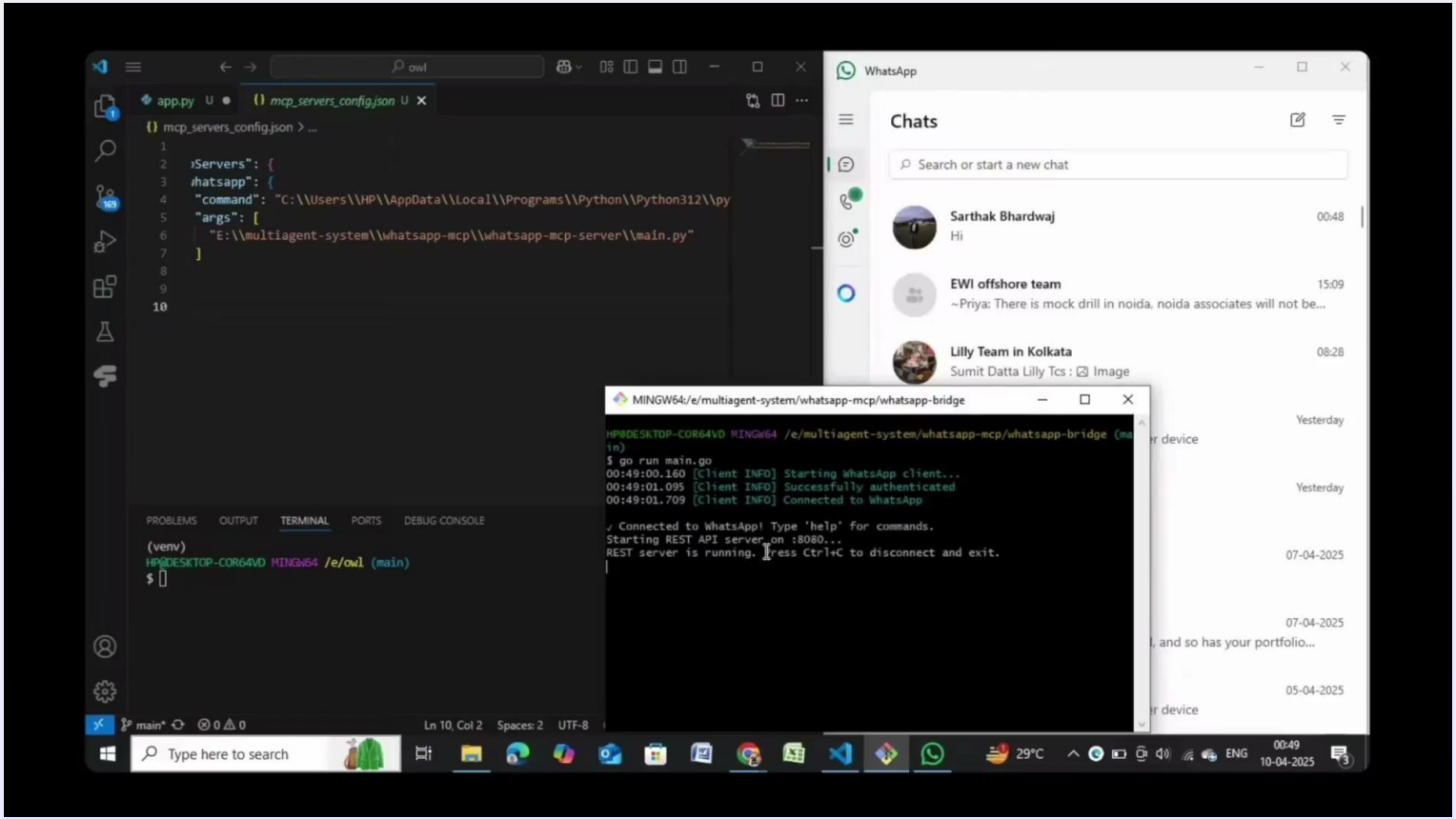
OWL Agents

CAMEL-AI



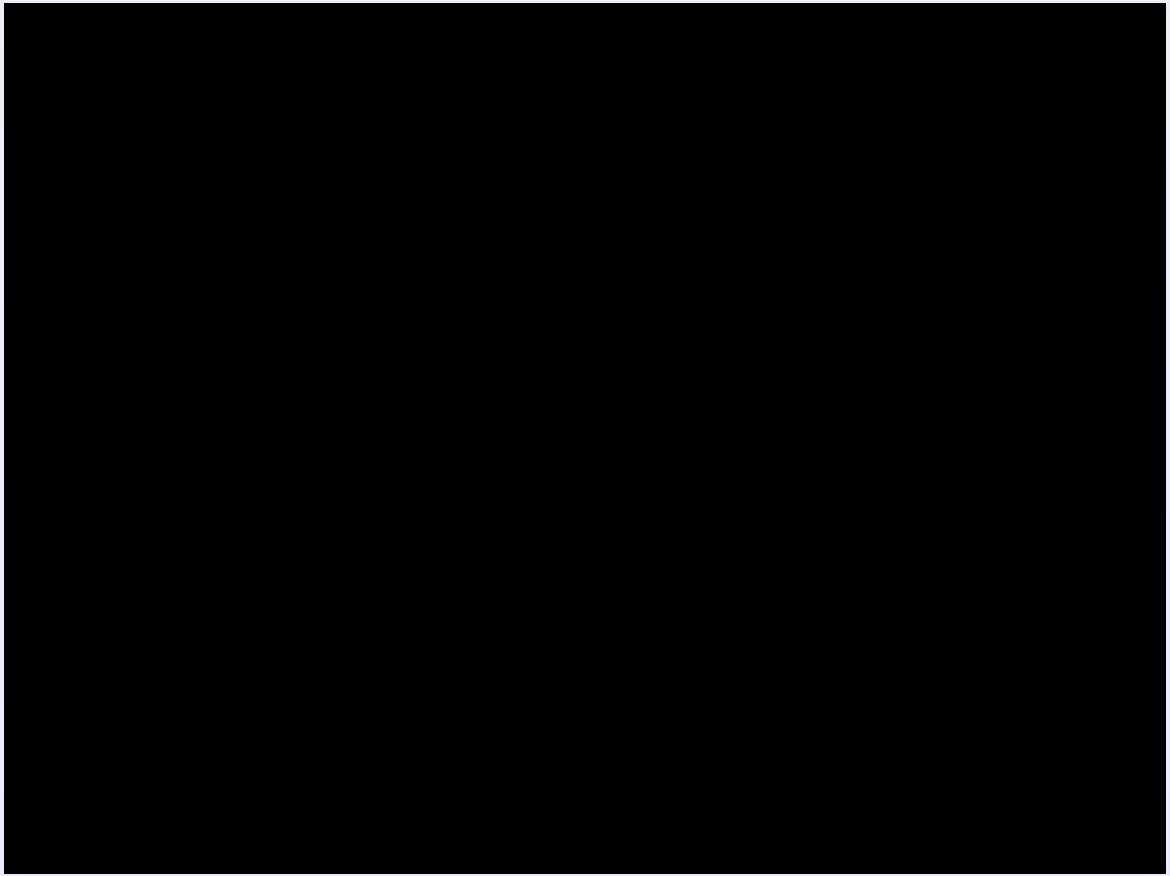
OWL Agents

CAMEL-AI



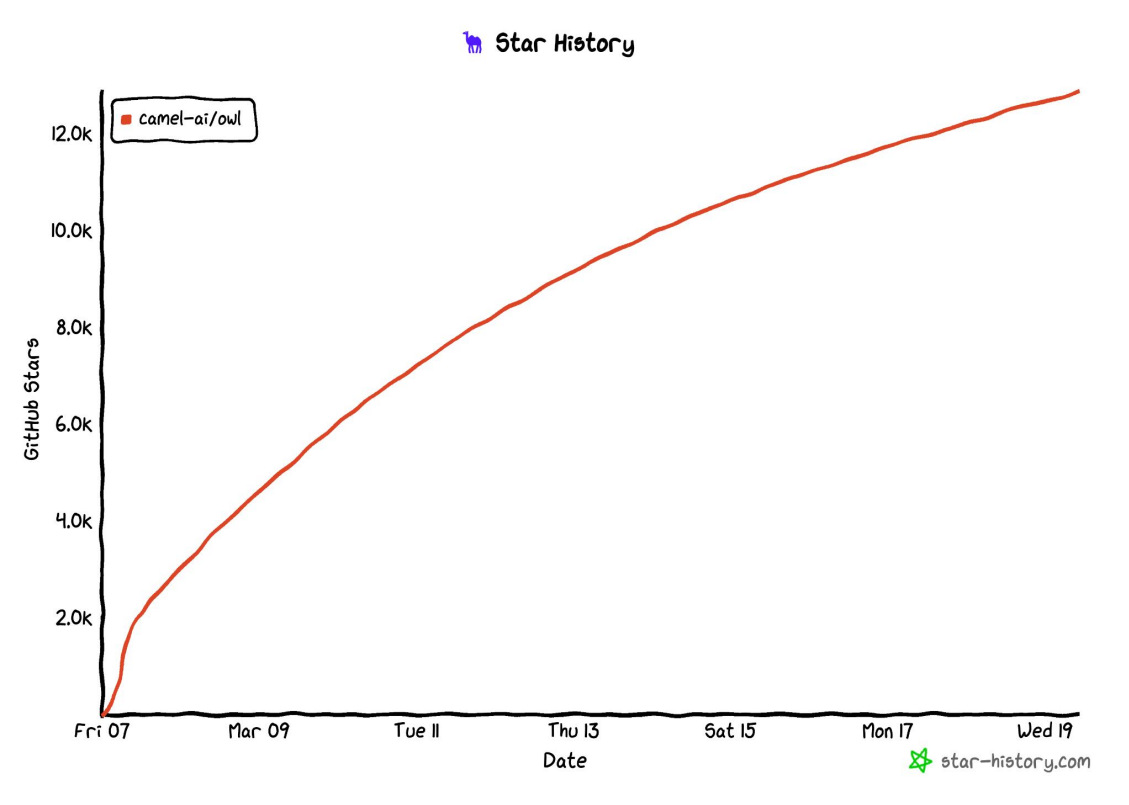
OWL Agents

CAMEL-AI



OWL Agents

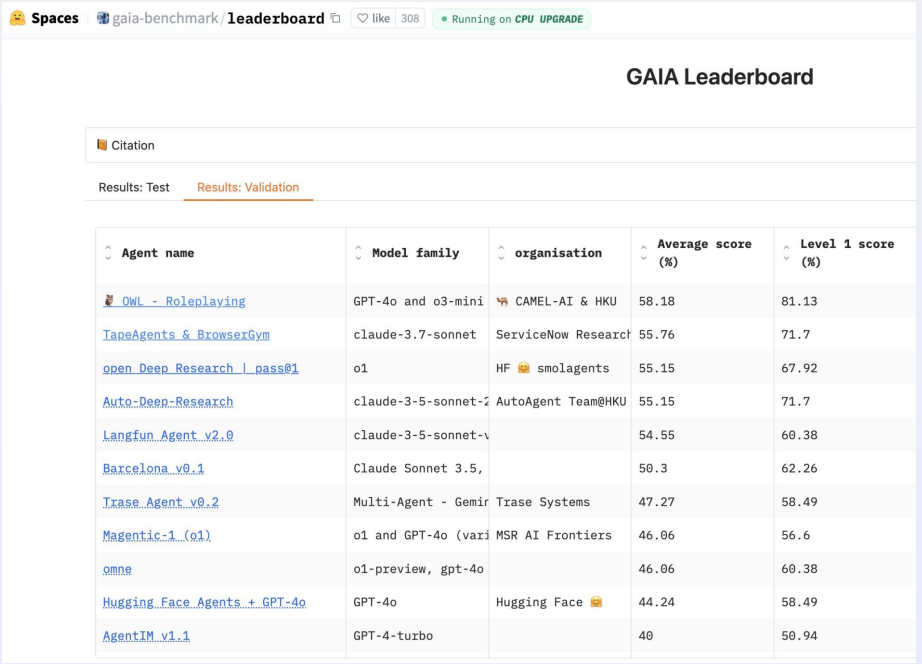
CAMEL-AI



OWL Agents

CAMEL-AI

- 58.18% average score on GAIA benchmark
- Ranking #1 among open-source frameworks!
- Updated results with Glaude 3.7 achieve 69.09%



Spaces | gaia-benchmark / leaderboard | like 308 | Running on CPU UPGRADE

GAIA Leaderboard

Citation

Results: Test | Results: Validation

Agent name	Model family	organisation	Average score (%)	Level 1 score (%)
OWL - Roleplaying	GPT-4o and o3-mini	CAMEL-AI & HKU	58.18	81.13
TapeAgents & BrowserGym	claude-3.7-sonnet	ServiceNow Researcl	55.76	71.7
open Deep Research | pass@1	o1	HF smolagents	55.15	67.92
Auto-Deep-Research	claude-3-5-sonnet-2	AutoAgent Team@HKU	55.15	71.7
Langfun Agent v2.0	claude-3-5-sonnet-v		54.55	60.38
Barcelona v0.1	Claude Sonnet 3.5,		50.3	62.26
Trase Agent v0.2	Multi-Agent - Gemir	Trase Systems	47.27	58.49
Magentic-1 (o1)	o1 and GPT-4o (vari	MSR AI Frontiers	46.06	56.6
omne	o1-preview, gpt-4o		46.06	60.38
Hugging Face Agents + GPT-4o	GPT-4o	Hugging Face	44.24	58.49
AgentIM v1.1	GPT-4-turbo		40	50.94

OWL Agents

CAMEL-AI

- 58.18% average score on GAIA benchmark
- Ranking #1 among open-source frameworks!
- Updated results with Glaude 3.7 achieve 69.09%

```python
# Configure toolkits
tools = [
    *BrowserToolkit(
        headless=False,  # Set to True for headless mode (e.g., on remote servers)
        web_agent_model=models["browsing"],
        planning_agent_model=models["planning"],
    ).get_tools(),
    *VideoAnalysisToolkit(model=models["video"]).get_tools(),
    *AudioAnalysisToolkit().get_tools(),  # This requires OpenAI Key
    *CodeExecutionToolkit(sandbox="subprocess", verbose=True).get_tools(),
    *ImageAnalysisToolkit(model=models["image"]).get_tools(),
    SearchToolkit().search_duckduckgo,
    SearchToolkit().search_google,  # Comment this out if you don't have google search
    SearchToolkit().search_wiki,
    *ExcelToolkit().get_tools(),
    *DocumentProcessingToolkit(model=models["document"]).get_tools(),
    *FileWriteToolkit(output_dir="./").get_tools(),
]

# Configure agent roles and parameters
user_agent_kwargs = {"model": models["user"]}
assistant_agent_kwargs = {"model": models["assistant"], "tools": tools}

# Configure task parameters
task_kwargs = {
    "task_prompt": question,
    "with_task_specify": False,
}

# Create and return the society
society = RolePlaying(
    **task_kwargs,
    user_role_name="user",
    user_agent_kwargs=user_agent_kwargs,
    assistant_role_name="assistant",
    assistant_agent_kwargs=assistant_agent_kwargs,
)
```

Finding the Scaling Laws of Agents

CAMEL-AI

Training ->
Ability of Evolution?

Scaling Laws of Multi-agent Systems?

Agentic Data Generation

CAMEL-AI

CAMEL-AI

Agentic SFT Data Generation with CAMEL and Fine-Tuning with Unsloth

STEP 1 → **Enter Reference Material URL**

Convert material into markdown via Firecrawl

STEP 2 → **Generate SFT Data with CAMEL**

Create tailored outputs based on reference material using a CAMEL agent

STEP 3 → **Fine-tune your model with Unsloth**

Optimize the model using generated instructions and responses

STEP 4 **Deploy the Model Effortlessly**

Use immediately or save for conventional deployment

https://docs.camel-ai.org/cookbooks/sft_data_generation_and_unsloth_finetuning_tinyllama.html

Project Loong: Synthesize Long CoTs at Scale through Verifiers

CAMEL-AI

Cookbook:

https://github.com/camel-ai/loong/blob/main/cookbooks/env_with_generator.ipynb

```python
from camel.environments import SingleStepEnv

env = SingleStepEnv(generator, verifier)
await env.setup()

obs = await env.reset(seed=42)

print(f"Observation: {obs}")
```

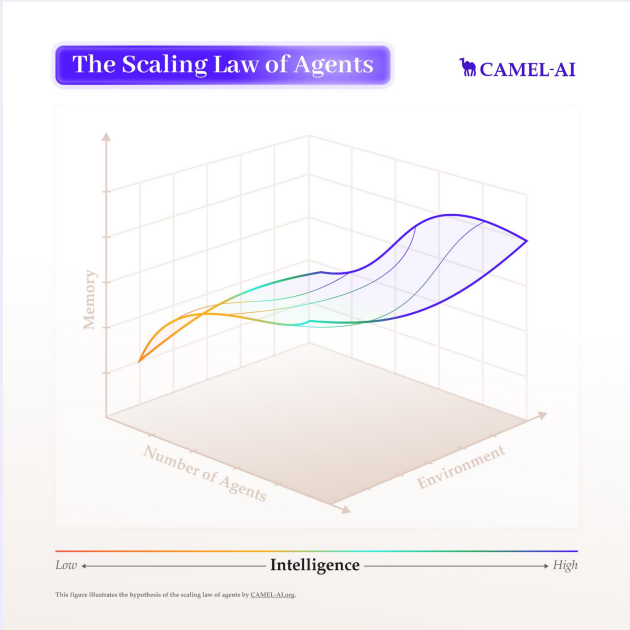
```python
agent = ChatAgent(model=model)

USER_PROMPT = r"""
You are an agent designed to answer mathematical questions with clarity and precision. Your task is to provide a
any mathematical problem posed by the user, ensuring the response is easy to follow. Adhere to these guidelines:
Analyze the mathematical question carefully and break down the solution process into clear, logical steps.
Use natural language to explain each step, incorporating LaTeX notation (e.g., $x + 2$)
for mathematical expressions when helpful. Conclude your response with the final answer enclosed
in a LaTeX \boxed{} environment (e.g., \boxed{5}).
Place this at the end of your explanation as a standalone statement.
It should be a Python expression, for example "[1, 2, 3]" for a list.

The question you should answer is: """

response = agent.step(USER_PROMPT + obs.question).msgs[0].content

next_obs, reward, done, info = await env.step(response)

agent.reset()

print(f"Is the episode done?: {done}")
print(f"Next Observation: {next_obs}")
print(f"Reward: {reward}")
print(f"Info: {info}")
```

The Next Scaling Laws?

CAMEL-AI

Finding the scaling law of agents:

- *# Parameters -> # Agents?* - **CAMEL, OASIS**
- *Data -> Environments?* - **CRAB, OWL**
- *Training -> Evolution?* -**DataGen, Loong**



CAMEL is a open source multi-agent system framework

1. Customises Agents

2. Build Multi-Agent Systems

3. Practical Application

Projects @ CAMEL-AI.org

camel Public

🐫 CAMEL: The first and the best multi-agent framework. Finding the Scaling Law of Agents. https://www.camel-ai.org

Python ☆ 12.1k ⑂ 1.3k

owl Public

🦉 OWL: Optimized Workforce Learning for General Multi-Agent Assistance in Real-World Task Automation

Python ☆ 15.9k ⑂ 1.9k

oasis Public

🏝️ OASIS: Open Agent Social Interaction Simulations with One Million Agents. https://oasis.camel-ai.org

Python ☆ 1.3k ⑂ 134

crab Public

🦀 CRAB: Cross-environment Agent Benchmark for Multimodal Language Model Agents. https://crab.camel-ai.org/

Python ☆ 336 ⑂ 45

loong Public

🐉 Loong: Synthesize Long CoTs at Scale through Verifiers.

Jupyter Notebook ☆ 249 ⑂ 21

agent-trust Public

🤝 The code for "Can Large Language Model Agents Simulate Human Trust Behaviors?"

Python ☆ 77 ⑂ 11

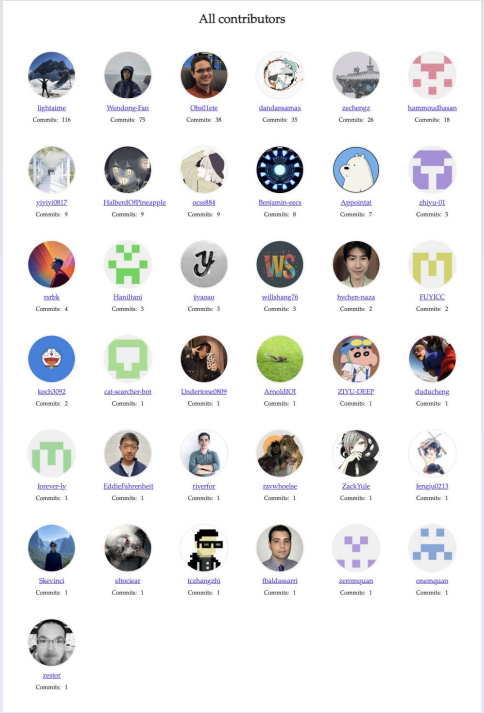
Thank You

CAMEL-AI.org

An open-source research organization

CAMEL-AI

Eigent AI



Join us in finding the scaling law of *Agents*!

@guohao_li