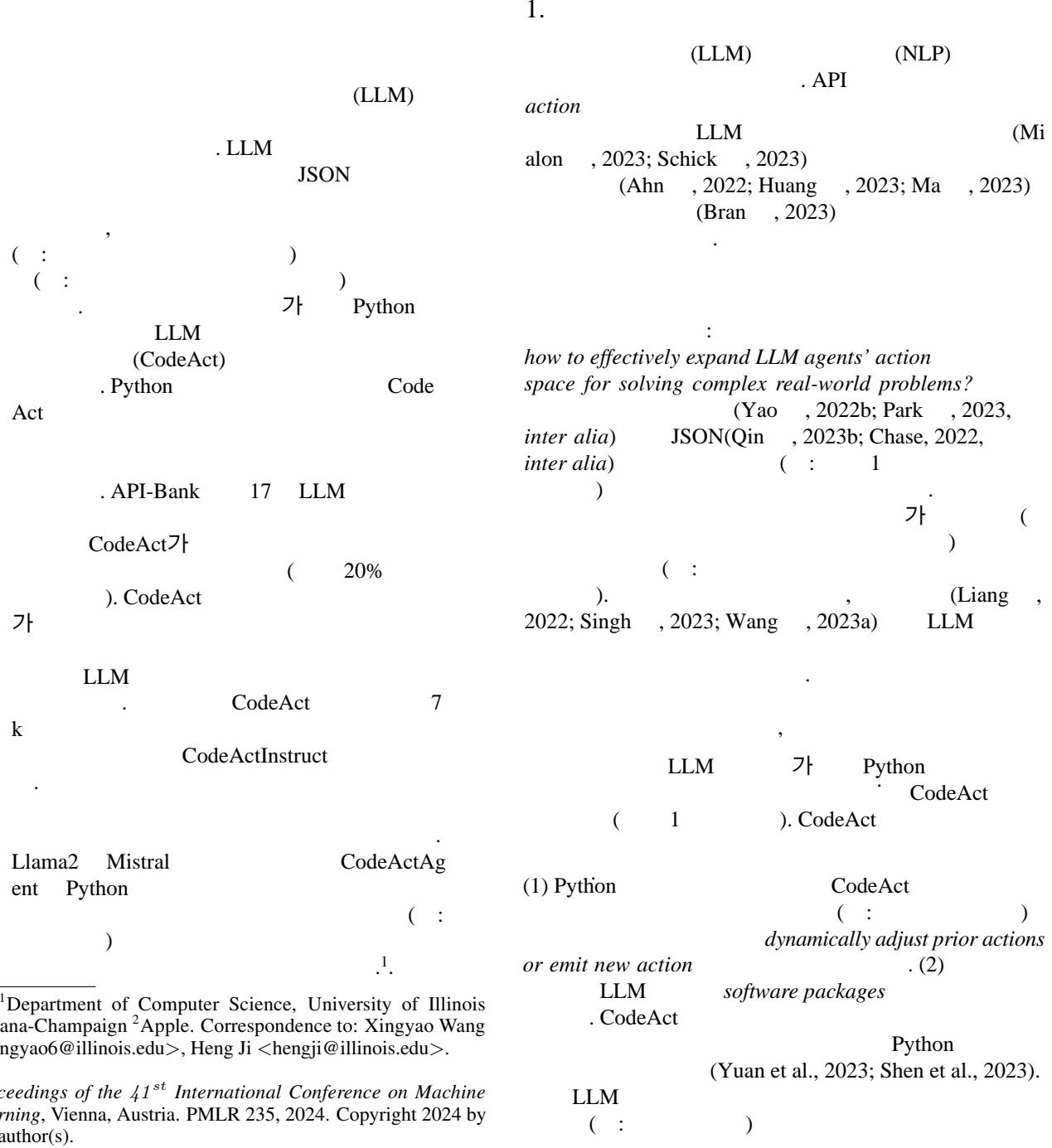

가 LLM

Xingyao Wang¹ Yangyi Chen¹ Lifan Yuan¹ Yizhe Zhang² Yunzhu Li¹ Hao Peng¹ Heng Ji¹



Executable Code Actions Elicit Better LLM Agents

Instruction: Determine the most cost-effective country to purchase the smartphone model "CodeAct 1". The countries to consider are the USA, Japan, Germany, and India.

Available APIs

```
[1] lookup_rates(country: str) -> (float, float)
[2] convert_and_tax(price: float, exchange_rate: float, tax_rate: float) -> float
[3] estimate_final_price(converted_price: float, shipping_cost: float) -> float
[4] lookup_phone_price(model: str, country: str) -> float
[5] estimate_shipping_cost(destination_country: str) -> float
```

LLM Agent using [Text/JSON] as Action

I should calculate the phone price in USD for each country, then find the most cost-effective country.

Action

```
Text: lookup_rates, Germany
JSON: {"tool": "lookup_rates", "country": "Germany"}
```

Environment 1.1, 0.19

Action

```
Text: lookup_phone_price, CodeAct 1, Germany
JSON: {"tool": "lookup_phone_price", "model": "CodeAct 1", "country": "Germany"}
```

Environment 700

Action

```
Text: convert_and_tax, 700, 1.1, 0.19
JSON: {"tool": "convert_and_tax", "price": 700, "exchange_rate": 1.1, "tax_rate": 0.19}
```

Environment 916.3

[... interactions omitted (look up shipping cost and calculate final price) ...]

Action

```
Text: lookup_rates, Japan
JSON: {"tool": "lookup_rates", "country": "Japan"}
```

[... interactions omitted (calculate final price for all other countries) ...]

Response

The most cost-effective country to purchase the smartphone model is Japan with price 904.00 in USD.

CodeAct: LLM Agent using [Code] as Action

I should calculate the phone price in USD for each country, then find the most cost-effective country.

Action

```
countries = ['USA', 'Japan', 'Germany', 'India']
final_prices = {}

for country in countries:
    exchange_rate, tax_rate = lookup_rates(country)
    local_price = lookup_phone_price("xact 1", country)
    converted_price = convert_and_tax(
        local_price, exchange_rate, tax_rate
    )
    shipping_cost = estimate_shipping_cost(country)
    final_price = estimate_final_price(converted_price, shipping_cost)
    final_prices[country] = final_price

most_cost_effective_country = min(final_prices, key=final_prices.get)
most_cost_effective_price = final_prices[most_cost_effective_country]
print(most_cost_effective_country, most_cost_effective_price)
```

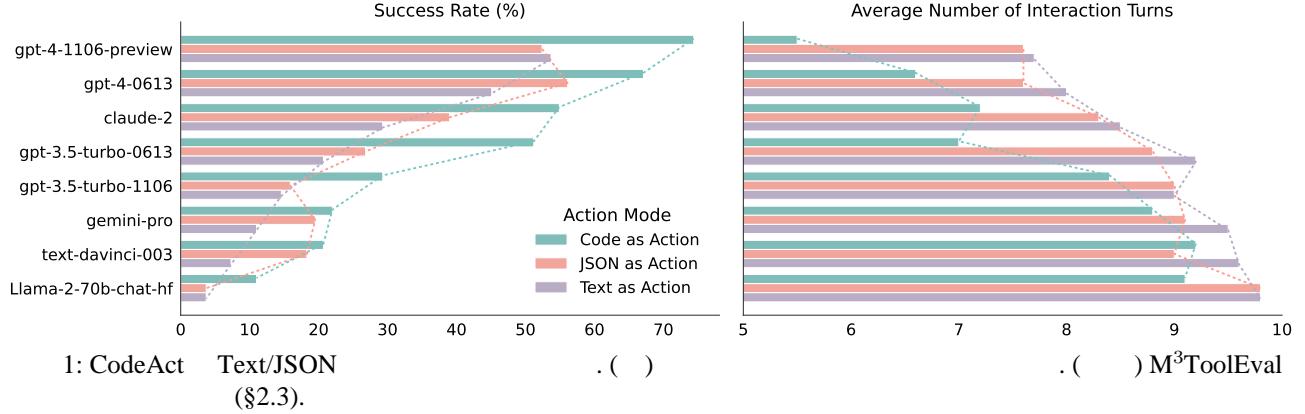
Environment 1.1, 0.19

Control & Data Flow of Code Simplifies Complex Operations

Response

The most cost-effective country to purchase the smartphone model is Japan with price 904.00 in USD.

Fewer Actions Required!



1: CodeAct Text/JSON (§2.3).

(Chen et al., 2023b; Wang et al., 2023d).
(3) **Code data** LLM
(Yang et al., 2024b).

CodeAct
가 .
(4) **JSON**
control and data flow
, one
(: if , for)
LLM

CodeAct()
for (,)
(:
) all
JSON
().

CodeAct (3 & 4).
(§2.2) CodeAct atomic tool use (,
)
CodeAct가 .
LLM

CodeAct
CodeAct
(4)

82

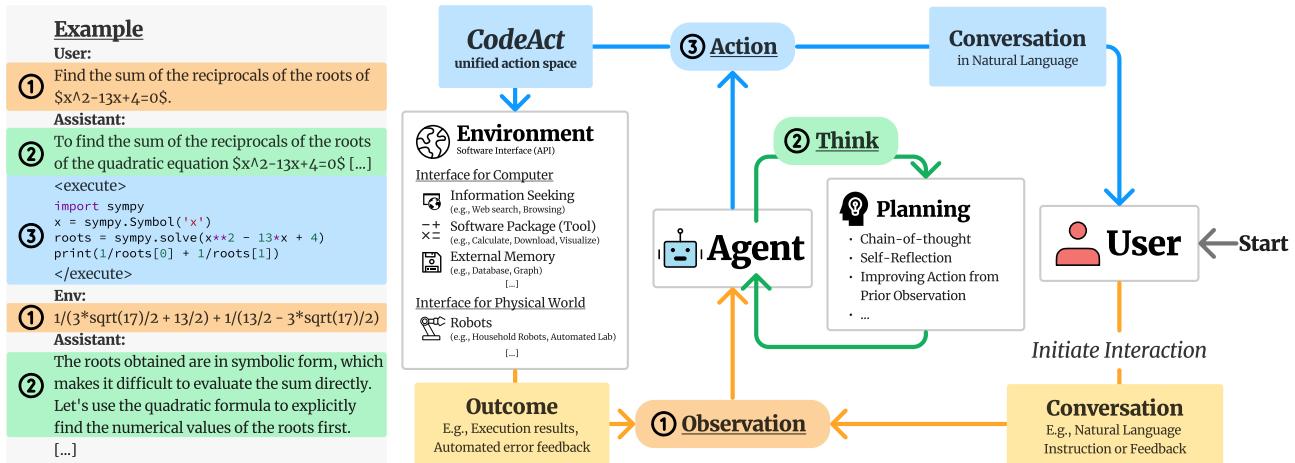
(M³ToolEval; §2.3).

17 LLM(

LLM)

CodeAct CodeAct
LLM
(1).

Executable Code Actions Elicit Better LLM Agents



2: CodeAct

code data

2.3. CodeAct

LLM 가 M³ToolEval . 3

Executable Code Actions Elicit Better LLM Agents

2: API-Bank			API			3: M ³ ToolEval			() 가		
Correctness (%), ↑			Success Rate (%), ↑			Avg. Turns (↓)					
Format of Action	CodeAct	JSON	Text	CodeAct	JSON	Text	CodeAct	JSON	Text	CodeAct	JSON
<i>Open-source LLMs</i>											
CodeLlama-7b-Instruct-hf	12.5	12.0	17.0	CodeLlama-7b-Instruct-hf	4.9	2.4	2.4	9.7	9.9	9.9	
CodeLlama-13b-Instruct-hf	11.8	7.8	14.0	CodeLlama-13b-Instruct-hf	4.9	4.9	4.9	9.8	9.8	9.7	
CodeLlama-34b-Instruct-hf	17.3	12.0	16.8	CodeLlama-34b-Instruct-hf	2.4	0.0	0.0	9.9	10.0	10.0	
Llama-2-7b-chat-hf	28.8	11.3	25.8	Llama-2-7b-chat-hf	0.0	1.2	2.4	8.9	9.5	9.6	
Llama-2-13b-chat-hf	38.1	8.5	37.3	Llama-2-13b-chat-hf	0.0	0.0	0.0	9.7	10.0	10.0	
Llama-2-70b-chat-hf	35.6	14.3	37.6	Llama-2-70b-chat-hf	11.0	3.7	3.7	9.1	9.8	9.8	
Mistral-7B-Instruct-v0.1	2.5	2.3	3.0	Mistral-7B-Instruct-v0.1	0.0	3.7	1.2	10.0	9.8	9.9	
lemur-70b-chat-v1	58.6	46.6	56.1	lemur-70b-chat-v1	13.4	15.9	12.2	9.1	9.3	9.4	
<i>Closed-source LLMs</i>											
claude-2	76.7	59.4	73.7	claude-2	54.9	39.0	29.3	7.2	8.3	8.5	
claude-instant-1	75.2	64.9	73.2	claude-instant-1	20.7	31.7	24.4	8.8	8.6	8.9	
gemini-pro	70.4	73.2	71.2	gemini-pro	22.0	19.5	11.0	8.8	9.1	9.5	
gpt-3.5-turbo-0613	74.4	73.9	73.4	gpt-3.5-turbo-0613	51.2	26.8	20.7	7.0	8.8	9.2	
gpt-3.5-turbo-1106	75.4	78.4	73.4	gpt-3.5-turbo-1106	29.3	15.9	14.6	8.4	9.0	9.0	
gpt-4-0613	75.4	82.0	74.4	gpt-4-0613	67.1	56.1	45.1	6.6	7.6	8.0	
gpt-4-1106-preview	76.7	82.7	73.4	gpt-4-1106-preview	74.4	52.4	53.7	5.5	7.6	7.7	
text-davinci-002	69.2	59.6	57.4	text-davinci-002	4.9	4.9	8.5	9.7	9.8	9.6	
text-davinci-003	75.4	76.9	69.7	text-davinci-003	20.7	18.3	7.3	9.2	9.0	9.6	
Frequency of Best-Performing Format ↑						Frequency of Best-performing Format ↑					
Open-source	4	0	4	Open-source	5	4	3	6	1	1	
Closed-source	4	5	0	Closed-source	7	1	1	6	2	1	
Overall	8	5	4	Overall	12	5	4	12	3	2	

1. CodeAct	§2.2	
	(가 LLM 17 12)	
가 CodeAct		
	(가 LLM 17 12).	
가	gpt-4-1106-preview	
	()	20.7%
,	2.1	.
	13.4%	가
	gpt-4-1106-preview	74.4%
	LLM	
CodeAct		

```

graph TD
    Matplotlib[Matplotlib] --> Python[Python]
    Python --> LLM[LLM]
    subgraph Interaction ["¬† multi-turn interaction"]
        Python
        LLM
    end
    LLM --> CodeAct[CodeAct]
    CodeAct --> CodeActInstru[CodeActInstru]
    CodeAct --> CodeAct
    CodeAct --> ct[ct]
    ct --> Section3_1[§3.1]
    Section3_1 --> CodeAct
    subgraph Section3_1_Bracket [§3.1]
        ct
        Section3_1
        CodeAct
        CodeActInstru
    end

```

2.4. CodeAct
3 LLM 가 Python(, §3.2
CodeActAgent)
LLM Python 가

```

graph TD
    CA1[CodeAct] -.-> L1[LLM]
    subgraph Box1 [ ]
        CA2[CodeActInstruct]
        L1a[LLaMA-2(Touvron et al., 2023)]
        M1[Mistral-7B(Jiang et al., 2023)]
        A1[Agent]
    end
    L1 -.-> L2[LLM]
    subgraph Box2 [ ]
        CA3[CodeAct]
        CA4[CodeAct]
        CA5[CodeAct]
    end

```

Python libraries 가 .
3 CodeActAgent Pandas , Scikit-
Learn

3.1. Geda AtInstrument

Executable Code Actions Elicit Better LLM Agents

User

```
Here is a dataset about auto MPG: "https://huggingface.co/datasets/scikit-learn-auto-mpg/main/auto-mpg.csv". You should download and load the dataset from the URL, take a look at it, check for missing values, do a train-test split, and train a regression model predicting its MPG. You should use everything except "mpg" and "car name" as the input features.
```

CodeActAgent

```
To accomplish this task, I will use the pandas library to download and load the dataset, check for missing values, and perform a train-test split. Then, I will use the scikit-learn library to train a regression model.
```

```

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

# Download and Load the dataset
df = pd.read_csv("https://huggingface.co/datasets/scikit-learn/auto-mpg/main/auto-mpg.csv")
# Check for missing values
print(df.isnull().sum())
# Remove rows with missing values
df = df.dropna()
# Perform train-test split
X = df.drop(columns=["mpg", "car_name"])
y = df["mpg"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a regression model
model = LinearRegression()
model.fit(X_train, y_train)
# Predict MPG for test data
y_pred = model.predict(X_test)
# Evaluate the model
print("Mean squared error: ", np.mean((y_test - y_pred) ** 2))
print("R2 score: ", r2_score(y_test, y_pred))

```

Environment

```
np
[...omitted for space...]
car name
[...omitted for space...]
dtype: int64
```

CodeActAgent

```
It seems that there is a data type issue in the dataset. The '?' character is present in the data, which is causing the error when trying to convert the data to a float. To fix this, I will remove the rows containing the '?' character.
```

```

# Remove rows containing '?'
df = df.replace('?', np.nan)
df = df.dropna()
X = df.drop(columns=["mpg", "car_name"])
y = df["mpg"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
print("Mean squared error: ", np.mean((y_test - y_pred) ** 2))
print("R2 score: ", r2_score(y_test, y_pred))

```

Environment

```
Mean squared error: 18.710864418838367
R2 score: 0.790159038676935
```

CodeActAgent

```
The mean squared error is 10.711 and the R^2 score is 0.790, which indicates that the model has a good fit with the data [...]
```

User

```
Can you also give me the MSE and R^2 for the training set?
```

CodeActAgent Able to Answer follow-up Question Interactively

```
Certainly! Here are the mean squared error (MSE) and R^2 scores [...]
```

```

# from sklearn.metrics import mean_squared_error
# Calculate MSE for training set
mse_train = mean_squared_error(y_train, model.predict(X_train))
# Calculate R^2 score for training set
r2_train = r2_score(y_train, model.predict(X_train))

print("Mean squared error (training set):", mse_train)
print("R2 score (training set):", r2_train)

```

Environment

```
Mean squared error (training set): 10.979255842373478
R2 score (training set): 0.826001578671067
```

CodeActAgent

```
The mean squared error for the training set is 10.979 and the R^2 score is 0.826, which indicates [...]
```

User

```
Can you visualize the regression coefficient? You may access them via model.coef_ and their names via model.feature_names_in_
```

CodeActAgent

```
Certainly! Here is a visualization of the regression coefficients [...]
```

```

# Import matplotlib.pyplot as plt
# Get regression coefficients
coef = model.coef_
# Get names of input features
feature_names = model.feature_names_in_
# Plot the regression coefficients
plt.bar(feature_names, coef)
plt.xlabel('Input Feature')
plt.ylabel('Regression Coefficient')
plt.title('Regression Coefficients')
# Display the plot
plt.show()

```

Environment

```
<Figure size 640x480 with 1 Axes>
```

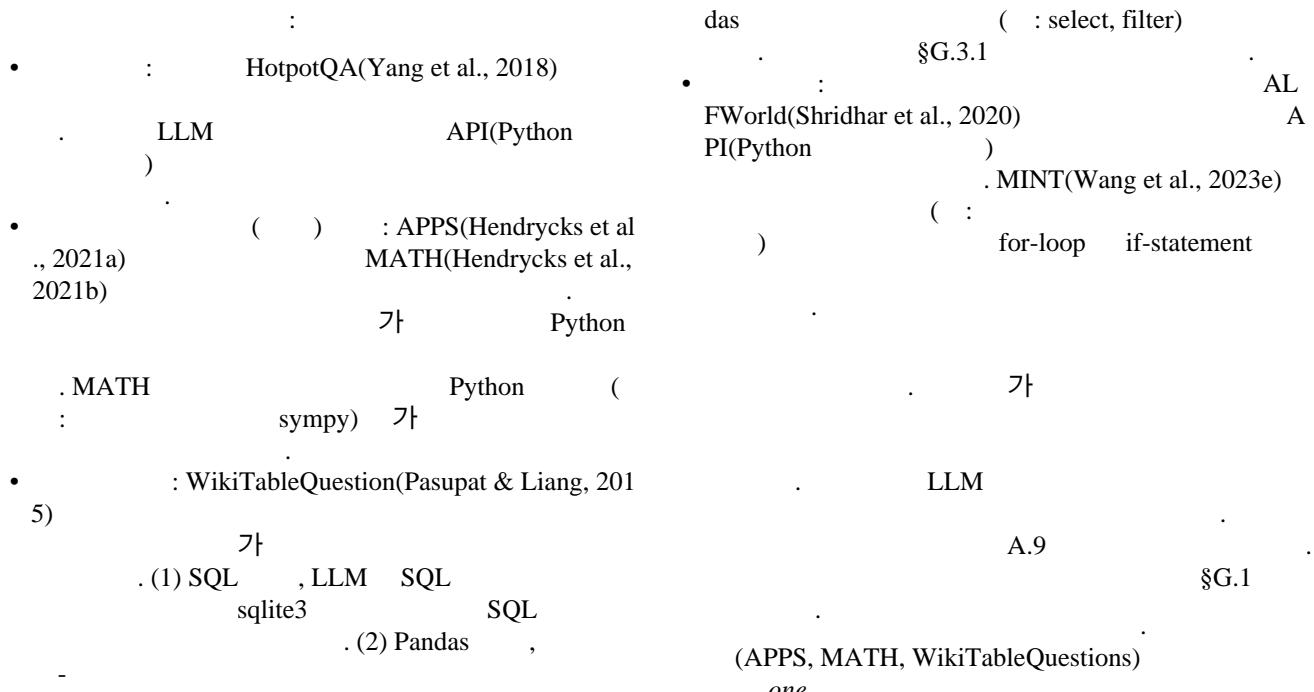
CodeActAgent

```
The regression coefficients are plotted as a bar chart, with each input feature on the x-axis and the corresponding coefficient on the y-axis [...]
```

3: CodeActAgent(Mistral-7b)

Python

<https://chat.xwang.dev/r/Vqn108G>



Executable Code Actions Elicit Better LLM Agents

4: §3.1		CodeActInstruct				
Data Mixture		Data Type	Data Name	# of Data Instances	# of Total Tokens	Avg. Tokens Per Instance
Prior Work	-		FireAct (Chen et al., 2023a) AgentInstruct (Zeng et al., 2023)	2,063 1,866	542,176 2,517,785	262.81 1349.30
CodeActInstruct (Ours)	Information Seeking	HotpotQA (Yang et al., 2018)	1,664	2,472,227	1485.71	
	Software Packages (Tool)	MATH (Math, (Hendrycks et al., 2021b))	1,732	1,719,467	992.76	
	Software Packages (Tool)	APPS (Code, (Hendrycks et al., 2021a))	647	1,235,472	1909.54	
	External Memory	WikiTableQuestion (Pasupat & Liang, 2015)	1,065	1,316,246	1235.91	
	Robot Planning	ALFWorld (Shridhar et al., 2020)	2,031	3,838,269	1889.84	
		Total	7,139	10,581,681	1482.24	
General Conversation	Single-Turn Reasoning	OpenOrca (Sub-sampled, (Lian et al., 2023))	50,000	14,034,152	280.68	
	Multi-Turn Conversations	ShareGPT (Sub-sampled, (Anonymous, 2023))	10,000	17,933,861	1793.39	
	Multi-Turn Conversations	ShareGPT (GPT-4, (OpenChat, 2023))	4,583	18,195,878	3970.30	
	Multi-turn Reasoning	Capybara (LDJnr, 2023)	4,647	4,982,435	1072.18	
		Total	69,230	55,146,326	796.57	

1. MINT(Wang et al., 2023e) 가 LLM (ntInstruct(Zeng et al., 2023) FireAct(Chen et al., 2023a), CodeActInstruct CodeAct Python (3))

G.3 1. MINT 가 LLM (ntInstruct(Zeng et al., 2023) FireAct(Chen et al., 2023a), CodeActInstruct CodeAct Python (3))

3. Anthropic claude-1-instant OpenAI gpt-3.5-turbo-061 3,Anthropic claude-1-instant OpenAI gpt-3.5-turbo-061 (gp t-3.5-turbo-0613-16k) gpt-4-06 (CoT) Lian et al., 2023), LLM ShareGPT(Anonymous, 2023; OpenChat, 2023), Capybara(LDJnr, 2023) 4. §C

13. GPT-3.5 claude-1-instant OpenAI gpt-3.5-turbo-0613-16k (gp t-3.5-turbo-0613-16k) gpt-4-06 (CoT) Lian et al., 2023), LLM ShareGPT(Anonymous, 2023; OpenChat, 2023), Capybara(LDJnr, 2023) 4. §C

ctInstruct (: 2).

3.2. Llama-2 7B(Touvron et al., 2023) Mistral 7B(Jiang et al., 2023) CodeActAgent Llama-2 4,096 , Mistral 16 ,384 §D

4. gpt-3.5 claude 6728 CodeActInstruct §G.2 411 .

CodeActInstruct MINT(Wang et al., 2023e) CodeAct LLM .

CodeActAgent .

Executable Code Actions Elicit Better LLM Agents

5: CodeActAgent		가		LLM		가		가		,			
		ID OD											
MT-Bench													
Agent Tasks													
Model	Size	Code as Action			Text as Action (OD)		Generic Tasks (OD)				Overall Average		
		MINT (ID)	MINT (OD)	M ³ ToolEval (OD)	Miniwob++	SciWorld	MMLU	HumanEval	GSM8K	MTBench			
<i>Open-source LLMs (LLaMA-2-based)</i>													
Llama2 Base	7B	-*	-*	-*	-*	-*	45.3	12.8	14.6	-*	-*		
Llama2 Chat	7B	3.2	11.0	0.0	0.0	5.9	48.0	13.9	27.7	6.3	21.1		
FireAct (Chen et al., 2023a)	7B	0.0	0.3	0.0	0.0	6.8	44.1	3.5	12.4	4.5	14.0		
AgentLM (Zeng et al., 2023)	7B	8.7	6.1	0.0	28.9	13.7	48.7	15.4	24.6	6.1	24.8		
CodeActAgent (LLaMA-2)	7B	51.3	20.4	0.0	25.5	17.6	50.6	18.1	38.3	7.5	30.7		
<i>Open-source LLMs (Mistral-based)</i>													
Mistral Base	7B	-*	-*	-*	-*	-*	60.1	30.5	52.1	-*	-*		
Mistral Instruct	7B	18.8	9.7	0.0	0.5	4.0	53.8	29.3	43.3	6.4	25.6		
CodeActAgent (Mistral)	7B	57.4	32.4	12.2	46.2	15.9	59.1	34.7	58.0	8.2	42.5		
<i>Closed-source LLMs</i>													
gpt-3.5-turbo-0613	-	33.9	38.2	51.2	66.7	21.2	70.0	48.1	57.1	7.9	54.0		
gpt-4-0613	-	68.6	70.2	67.1	69.4	36.4	86.4	67.0	87.1	9.0	71.7		

*

MINT 가 (, MINT ALFWorld
MATH 가), MINT

, MINT
k = 5 MINT
MiniWob++ (Kim et al.,
2023) ScienceWorld(

, (Wang et al., 2022a))

CodeActAgent 가

가 : QA MMLU(Hendrycks et al.,
2020), HumanEval(Chen et al.,
2021), GSM8K(
Cobbe et al., 2021), MTBench(Zhen
g et al., 2023).

CodeActAgent CodeAct
, CodeActAgent() MINT

가 LLM M³
ToolEval CodeActAgent(Mistral)
13B) LLM (3). §H

CodeActAgent

가 ,
CodeActAgent(LLaMA2, 7B)
AgentLM-7B(

Zeng et al., 2023)

CodeActAgent LLM
5 CodeActAgent()
CodeActAgent MMLU
LLM (Mistral, 7B).

A.8 CodeActInstruct

. CodeA

ctInstruct

,

4.

4.1. LLM

(Wang et al., 2023b)

LLM

(Park

et al., 2023; Qian et al., 2023),

(Zhu et a

1., 2023; Fischer, 2023),

(Wei et al.

, 2022; Chen et al., 2023d) 가

4

가

LM 가 (Wang et al., 2023e; Yang et al., 202
4a) (Lee et al., 2022) (Qin et al., 2023a)

LLM

§A

CodeAct

TaskWeaver(Qiao et al., 2023) 가

§B

4.2. LLM

LLM
ng et al., 2023b) 가

가

(Wa

a) , prompt engineering (Liu et al., 2023

(Wei et al., 2022). (Wang et al., 2022b; Chen et al., 202

(Yao et al., 2023a)

LLM

- Boerner, T. J., Deems, S., Furlani, T. R., Knuth, S. L., T owns, J. : : NSF : . pp. 173 – 176. 2023.
- Bran, A. M., Cox, S., White, A. D., Schwaller, P. Chem - crow: . arX iv arXiv:2304.05376, 2023.
- Cano, A. H., Pagliardini, M., Köpf, A., Matoba, K., Mo-hta shami, A., Wang, X., Fan, O. S., Marmet, A., Bayazit, D., Krawczuk, I., Chen, Z., Salvi, F., Bosselut, A., Jaggi, M . epfllm megatron-llm, 2023. URL <https://github.com/epfLLM/Megatron-LLM>.
- Chase, H. LangChain, 2022 10 . URL <https://github.com/langchain-ai/langchain>.
- Chen, B., Shu, C., Shareghi, E., Collier, N., Narasimhan, K .. Yao, S. Fireact: . a rXiv arXiv:2310.05915, 2023a.
- Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al. ↗ arXiv arXiv:2107.03374, 2021.
- Chen, X., Lin, M., Schärli, N., Zhou, D. ↗ . arX iv arXiv:2304.05128, 2023b.
- Chen, Y., Sikka, K., Cogswell, M., Ji, H., Divakaran, A . Dress: . arXiv arXiv:2311.10081, 2023c.
- Chen, Y., Sikka, K., Cogswell, M., Ji, H., Divakaran, A . ↗ . a rXiv arXiv:2309.04461, 2023d.
- Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fe dus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., et al . arXiv arXiv:2210.11416, 2022.
- Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R . . arXiv arXiv:2110.14168, 2021.
- Eloundou, T., Manning, S., Mishkin, P., Rock, D. Gpts gpts : . arX iv arXiv:2303.10130, 2023.
- Fischer, K. A. agi(socialagi) . arX iv (rlp): arXiv:2305.12647, 2023.
- Gao, L., Madaan, A., Zhou, S., Alon, U., Liu, P., Yang, Y., Callan, J., Neubig, G. Pal: . , pp. 10764 – 10799. PMLR, 2023.
- Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. . International Conference on Learning Representations, 2020.
- Hendrycks, D., Basart, S., Kadavath, S., Mazeika, M., Arora, A., Guo, E., Burns, C., Puranik, S., He, H., Song, D., et al. . 35 (2) , 2021a.
- Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. . 35 (2) , 2021b.
- Hong, S., Zheng, X., Chen, J., Cheng, Y., Wang, J., Zhang, C., Wang, Z., Yau, S. K. S., Lin, Z., Zhou, L., et al. Metag pt: . arXiv arXiv:2308.00352, 2023.
- Hong, S., Lin, Y., Liu, B., Liu, B., Wu, B., Li, D., Chen, J., Zhang, J., Wang, J., Zhang, L., Zhang, L., Yang, M., Zhu ge, M., Guo, T., Zhou, T., Tao, W., Wang, W., Tang, X., Lu, X., Zheng, X., X., Y., Cheng, Y., Xu, Z. Wu, C. : llm , 2024.
- Huang, W., Wang, C., Zhang, R., Li, Y., Wu, J., Fei-Fe i, L. Voxposer: ↗ 3D . arX iv arXiv:2307.05973, 2023.
- Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., C haplot, D. S., Casas, D. d. I., Bressand, F., Lengyel, G., La mple, G., Saulnier, L., et al. 7b. arXiv arXiv:2310.06825, 2023.
- Kim, G., Baldi, P., McAleer, S. . arX iv arXiv:2303.17491, 2023.
- LDJnr. . <https://hf.co/datasets/LDJnr/Verified-Camel>, <https://hf.co/datasets/LDJnr/Pure-Dove>, <https://hf.co/datasets/LDJnr/>

- LessWrong-Amplify-Instruct, 2023.
- Lee, M., Liang, P., and Yang, Q. : . , 35:27730 – 27744, 2022.
- 022 CHI , pp. 1 – 19, 2022.
- Li, M., Song, F., Yu, B., Yu, H., Li, Z., Huang, F., and Li, Y. API-bank: LLM , 2023.
- Lian, W., Goodson, B., Pentland, E., Cook, A., Vong, C., and "Teknium". Openorca: gpt . https://https://huggingface.co/OpenOrca/OpenOrca, 2023.
- Liang, J., Huang, W., Xia, F., Xu, P., Hausman, K., Ichter, B., Florence, P., and Zeng, A. : . arXiv arXiv:2209.07753, 2022.
- Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G. : . ACM , 5(9):1 – 35, 2023a.
- Liu, R., Yang, R., Jia, C., Zhang, G., Zhou, D., Dai, AM, Yang, D., Vosoughi, S. : . arXiv arXiv:2305.16960, 2023b.
- Ma, Y. J., Liang, W., Wang, G., Huang, D.-A., Bastani, O., Jayaraman, D., Zhu, Y., Fan, L., Anandkumar, A. Eureka: - . arXiv arXiv:2310.12931, 2023.
- Mialon, G., Dess`e, R., Lomeli, M., Nalmpantis, C., Pasnuru, R., Raileanu, R., Rozière, B., Schick, T., Dwivedi-Yu, J., Celikyilmaz, A., et al. : . arXiv arXiv:2302.07842, 2023.
- Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., et al. Webgpt: . arXiv arXiv:2112.09332, 2021.
- OpenChat. Sharegpt . https://hf.co/datasets/openchat/openchat_sharegpt_v3/blob/main/sharegpt_gpt4.json, 2023. LLM . GPT-4 .
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., . , 2020.
- Park, J. S., O'Brien, J., Cai, C. J., Morris, M. R., Liang, P., Bernstein, M. S. : . 36 ACM , pp. 1 – 22, 2023.
- Pasupat, P., Liang, P. : . 53 7 (1) , pp. 1470 – 1480, 2015.
- Patil, S. G., Zhang, T., Wang, X., and Gonzalez, J. E. Gorilla: Large language model connected with massive apis. ArXiv, abs/2305.15334, 2023. URL <https://api.semanticscholar.org/CorpusID:258865184>.
- Qian, C., Cong, X., Yang, C., Chen, W., Su, Y., Xu, J., Liu, Z., and Sun, M. : . arXiv arXiv:2307.07924, 2023.
- Qiao, B., Li, L., Zhang, X., He, S., Kang, Y., Zhang, C., Yang, F., Dong, H., Zhang, J., Wang, L., et al. Taskweaver: . arXiv arXiv:2311.17541, 2023.
- Qin, Y., Hu, S., Lin, Y., Chen, W., Ding, N., Cui, G., Zeng, Z., Huang, Y., Xiao, C., Han, C., et al. : . arXiv arXiv:2304.08354, 2023a.
- Qin, Y., Liang, S., Ye, Y., Zhu, K., Yan, L., Lu, Y.-T., Lin, Y., Cong, X., Tang, X., Qian, B., Zhao, S., Tian, R., Xie, R., Zhou, J., Gerstein, M. H., Li, D., Liu, Z., and Sun, M. Toolllm: 16000+ API . ArXiv, abs/2307.16789, 2023b. URL <https://api.semanticscholar.org/CorpusID:260334759>.
- Schick, T., Dwivedi-Yu, J., Dess`e, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., and Scialom, T. : . arXiv arXiv:2302.04761, 2023.
- Shen, Y., Song, K., Tan, X., Li, D., Lu, W., Zhuang, Y. Hugginggpt: huggingface chatgpt AI : . arXiv arXiv:2303.17580, 2023.
- Shridhar, M., Yuan, X., Cote, M.-A., Bisk, Y., Trischler, A., Hausknecht, M. Alfworld: . , 2020.

- Singh, I., Blukis, V., Mousavian, A., Goyal, A., Xu, D., Tremblay, J., Fox, D., Thomason, J., and Garg, A. Progrompt: Generating situated robot task plans using large language models. 2023 IEEE International Conference on Robotics and Automation (ICRA), pp. 11523 – 11530, 2023. doi: 10.1109/ICRA48891.2023.10161317.
- Sur\'{e} s, D., Menon, S., Vondrick, C. ViperGPT: A few-shot learning framework for visual reasoning. IEEE International Conference on Computer Vision (ICCV), 2023.
- Tang, X., Jin, Q., Zhu, K., Yuan, T., Zhang, Y., Zhou, W., Qu, M., Zhao, Y., Tang, J., Zhang, Z., et al. Llama 2: An open foundation and fine-tuned chat model. arXiv preprint arXiv:2402.04247, 2024.
- TIOBE Index. Tiobe index. <https://www.tiobe.com/tiobe-be-index/>, 2024-01-23. TIOBE.
- . 가 . , .
- Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bholes, S., et al. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288, 2023.
- Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., Anandkumar, A. Voyager: Few-shot learning with large language models. arXiv preprint arXiv:2305.16291, 2023a.
- Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., et al. Voyager: Few-shot learning with large language models. arXiv preprint arXiv:2308.11432, 2023b.
- Wang, R., Jansen, P. A., Côté, M.-A., and Am-manabrolu, P. Scienceworld: Is your agent smarter than a 5 grader? 2022a. URL <https://api.semanticscholar.org/CorpusID:247451124>.
- Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., Zhou, D. Code4Struct: few-shot learning with large language models. arXiv preprint arXiv:2203.11171, 2022b.
- Wang, X., Li, S., Ji, H. Code4Struct: few-shot learning with large language models. Rogers, A., Boyd-Graber, J., Okazaki, N. (2022), 61, 2022a.
- Computational Linguistics(1 :), pp. 3640 – 3663, 2023. 7 c. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.202. URL <https://aclanthology.org/2023.acl-long.202>.
- Wang, X., Peng, H., Jabbarvand, R., and Ji, H. Leti: A few-shot learning framework for large language models. arXiv preprint arXiv:2309.10691, 2023e.
- Wang, X., Wang, Z., Liu, J., Chen, Y., Yuan, L., Peng, H., Ji, H. Mint: A few-shot learning framework for large language models. arXiv preprint arXiv:2302.01560, 2023f.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits in large language models. Advances in Neural Information Processing Systems, 35: 24824 – 24837, 2022.
- Xu, Q., Hong, F., Li, B., Hu, C., Chen, Z., Zhang, J., 2023.
- Yang, J., Prabhakar, A., Narasimhan, K., Yao, S. Introducing . 36, 2024a.
- , K., , J., , J., , C., , Y. R., , S., , Z., , X., , X., , Y., , H., , C. ILM : . 2024b.
- Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., and Manning, C. D. Hotpotqa: A few-shot learning framework for question answering. 2018, p. 2369 – 2380, 2018.
- Yang, Z., Liu, A., Liu, Z., Liu, K., Xiong, F., Wang, Y., Yang, Z., Hu, Q., Chen, X., Zhang, Z., Luo, F., Guo, Z., Li, P., Liu, Y., 2024c.
- Yao, S., Chen, H., Yang, J., and Narasimhan, K. : . 35:20744 – 20757, 2022a.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. R., and Cao, Y. React: Synergizing reasoning and actions in language models. 11 _____, 2022b.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., Narasimhan, K. : . arXiv arXiv:2305.10601, 2023a.

Yao, W., Heinecke, S., Niebles, J. C., Liu, Z., Feng, Y., Xu e, L., Murthy, R., Chen, Z., Zhang, J., Arpit, D., et al. Retr o-former: . arXiv arXiv:2308.02151, 2023b.

Yuan, L., Chen, Y., Wang, X., Fung, Y. R., Peng, H., and Ji , H. Craft: Customizing llms by creating and retrieving fro m specialty toolsets. ArXiv, abs/2309.17428, 2023. URL ht tps://api.semanticscholar.org/CorpusID:263310662.

Zeng, A., Liu, M., Lu, R., Wang, B., Liu, X., Dong, Y. Tang, J. Agenttuning: llms , 2023.

Zhang, C., Liu, L., Wang, J., Wang, C., Sun, X., Wang, H., and Cai, M. : . arXiv arXiv:2308.12033, 2023.

, L., , W.-L., , Y., , S., , Z., , Y., , Z., , Z., , D., , E., . mt-bench llm-as-a-judge . arXiv arXiv:2306.05685, 2023.

Zheng, T., Zhang, G., Shen, T., Liu, X., Lin, B. Y., Fu, J., Chen, W., and Yue, X. Opencodeinterpreter: . https://arxiv.org/abs/2402.14658, 2024.

Zhu, X., Chen, Y., Tian, H., Tao, C., Su, W., Yang, C., Hu ang, G., Li, B., Lu, L., Wang, X., et al. Ghost in the minecr aft: . arXiv arXiv:2305.17144, 2023.

Executable Code Actions Elicit Better LLM Agents

A.6: API-Bank(Li et al., 2023) M³ToolEval

Format	Action
CodeAct	AddAgenda (content="Meeting with John", time="2023-10-26 09:00:00")
JSON	{"action": "AddAgenda", "content": "Meeting with John", "time": "2023-10-26 09:00:00"}
Text	Action: AddAgenda, content: Meeting with John, time: 2023-10-26 09:00:00

A.7: Com M³ToolEval 가 .

Benchmark	M ³ ToolEval (This work)	ToolBench (Qin et al., 2023b)	APIBench (Patil et al., 2023)	API-Bank (Li et al., 2023)	ToolBench (Xu et al., 2023)
Requiring multi-turn interaction	✓	✓	✗	✗	✗
Multiple tools	✓	✓	✓	✓	✓
Evaluation	Answer Match	LLM Evaluator	AST Tree Match	API-Call Match	Test Case
No dependency on external API*	✓	✗	✗	✓	✗
Supported API Action Format	CodeAct & JSON & Text	JSON	CodeAct	JSON	CodeAct

* 가 API(: RapidAPI, Google Sheet)
(: API API 가).

A.8: . 가 . , . ID
OD . . MT-Bench

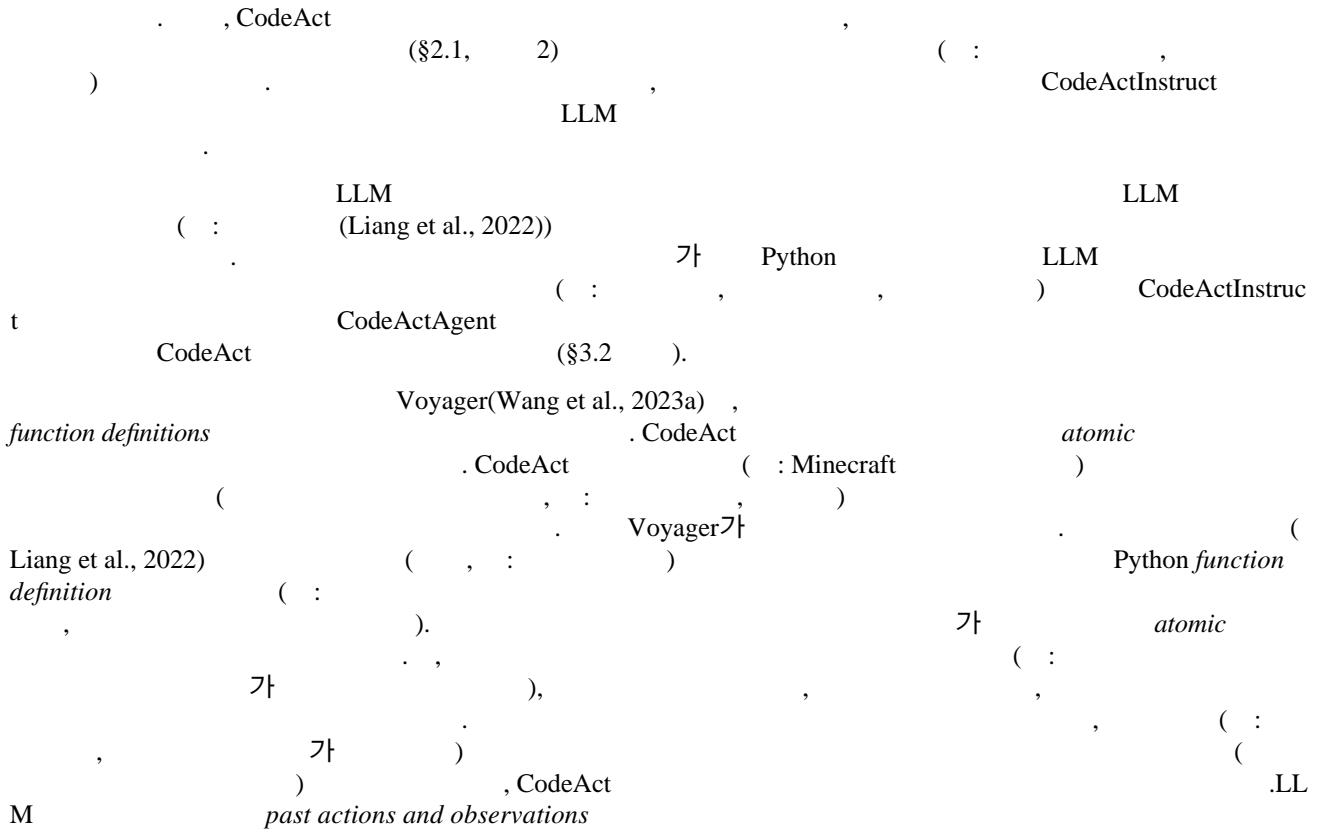
Model	Size	Agent Tasks				Generic LLM Tasks				Overall Average	
		Code as Action		Text as Action (OD)		(OD)					
		MINT (ID)	MINT (OD)	Miniwob++	SciWorld	MMLU	HumanEval	GSM8K	MTBench		
CodeActAgent (Llama2-based)	7B	51.3	20.4	<u>25.5</u>	17.6	50.6	<u>18.1</u>	38.3	7.5	35.1	
w/o CodeAct	7B	17.0	15.5	36.4	16.9	<u>49.5</u>	14.7	36.0	7.2	34.5	
w/o general conversations	7B	29.2	<u>15.9</u>	0.0	<u>17.1</u>	46.4	19.7	20.6	4.1	22.9	
CodeActAgent (Mistral-based)	7B	57.4	32.4	<u>46.2</u>	<u>15.9</u>	<u>59.1</u>	34.7	58.0	8.2	46.8	
w/o CodeAct	7B	32.9	<u>23.0</u>	47.8	17.0	59.9	<u>33.2</u>	59.5	8.3	46.2	
w/o general conversations	7B	50.5	13.9	0.0	11.0	52.4	27.9	26.8	2.6	22.6	

A.

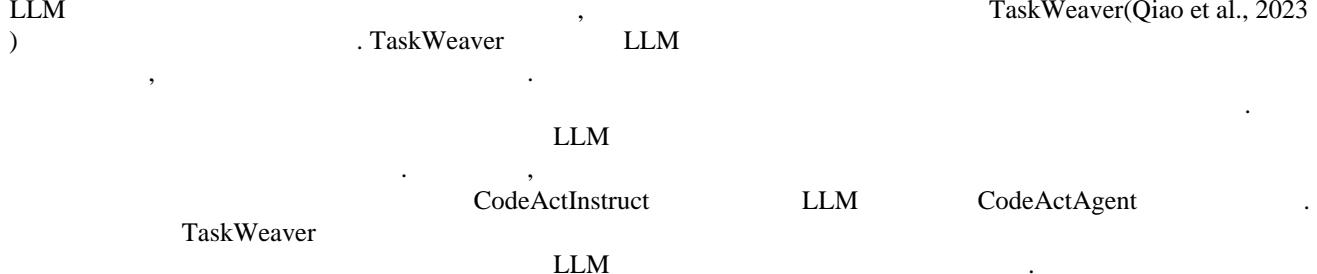
CodeAct LLM

, Code4Struct(Wang et al., 2023c), PaL(Gao et al., 2023),
Meta-GPT(Hong et al., 2023), code-as-policy(Liang et al., 2022),
ViperGPT(Sur’s et al., 2023), Voyager(Wang et al., 2023a),
Data Interpreter(Hong et al., 2024)

(,)
↗ first attempt
(:) 가
(:),
).



B. TaskWeaver



C.

- ShareGPT(, 2023):
.
 - ShareGPT(GPT-4)(OpenChat, 2023):
 - OpenOrca(Lian et al., 2023): OpenOrca CoT

- Capybara(LDJnr, 2023):

D. CodeActAgent

SFT	9,000 4xA100 40GB SXM	† Megatron-LLM(Cano et al., 2023) chatML ²	,
4 stral LLM	Tensor Parallel, 50 32	(1e-6 5))
		LLaMA-2	Mi 3

E. CodeAct

CodeAct () , chatML
 §F §G.3 †
 <| im_start |>
 Python(Jupyter Notebook) " < ex
 ecute> (:<execute> print("Hello World!") </execute>). <execute>
 PIP & <execute> !pip install [package needed] </execute>
 <execute> . † < ex
 ecute> , <| im_end |>

F. M³ToolEval

: {{ }}
 {{ }}
 !
 .34. : {{ : Legendary Wand † .} } 'xx.xx' (: 12
 : <tool_name > ' ' ; ' ' ; ' ' ; ' ' ; ' ' ; ' ' ;
 ,)
 {{...}}
 F.1. {{ }}

²<https://github.com/openai/openai-python/blob/release-v0.28.0/chatml.md>

Executable Code Actions Elicit Better LLM Agents

```
[1] click_url: URL           가 URL [Clickable ' <url_argument> ' ] : url(str)
r).                                URL . Signature: click_url(url: str) -
> str [2] go_to_previous_page:   가
. Signature: go_to_previous_page() -> str [3] scroll_down:           가
. Signature: scroll_down() -> str [4] scroll_up:                   가
. scroll_up() -> str [5] view:          가
. view() -> str [6] calculator:      가
, "2 + (3 * 4)"    14 . calculator(expression: str) -> float
```

F.2. {{ }}

F.3.

Python
Python for-loop, if-statements
'End Action' 'Action:'
'Action: tool_name(argument_1) End Action'

F.4. Json

가 JSON
'Action:' 'End Action'
argument_1"]} End Action'

F.5.

'Action:' 'End Action' : 'Action: tool_name, argument_1 End Action'

G. CodeAct

G.1.

- APPS (Hendrycks et al., 2021a):

A.9: CodeActInstruct

Domain	Capability	Dataset	# of Instances
Web Search	Information seeking through search API	HotpotQA (Yang et al., 2018)	3,000
Math Reasoning	Math problem-solving using math Libraries in Python (e.g., <code>sympy</code>)	MATH (Hendrycks et al., 2021a)	5,586
Code Generation	Self-debug from Python error messages and traceback	APPS (Hendrycks et al., 2021b)	4,439
Tabular Reasoning	Tabular Reasoning using <code>pandas</code> and <code>sqlite3</code> (for SQL) library	WikiTableQuestion (Pasupat & Liang, 2015)	3,000
Embodied Planning	Interact with embodied environments through APIs	ALFWORLD (Shridhar et al., 2020)	3,553

- WikiTableQuestion (Pasupat & Liang, 2015): (14149, 1500) 가 (, ,) 3000, 1500 SQL
 - HotpotQA (Yang et al., 2018): " " 15661 (90447, 3000)
 - MATH (Hendrycks et al., 2021b): 3 A.9 5586 가 .
 - ALFWorld (Shridhar et al., 2020): AlfWorld

G.2.

2 가 , LLM

- - : LLM 가 - API
 - : 가

G.3.

MINT(Wang et al., 2023e)

G.3.1. (WIKITABLEQUESTION)

SQL 가 . SQL

<thought> "Hello World!" . </thought>.

<thought> Hello World! </thought>

After that, you have two options:

```

1) Python                                     "<execute>"  

   ( : <execute> print("Hello World!") </execute>).  

2)                                     "<solution>"  

   . : < solution> A </solution>  

5                                         2  

---  

:  

  ( 5 ):  

    City Country (NOC) Round 1 Run-off Round 3 Round 4 Round 5  

0      Athens     Greece     32      NaN     38.0     52.0     66.0  

1      Rome      Italy      23      NaN     28.0     35.0     41.0  

2     Cape Town South Africa 16      62.0     22.0     20.0      NaN  

3    Stockholm  Sweden     20      NaN     19.0      NaN      NaN  

4   Buenos Aires Argentina   16      44.0      NaN      NaN      NaN  

5                               ?  

SQLite3                         SQL  

'conn'      (SQLite3)           <execute>  

  ' data_table '  

<   > "Round 5"          null          "City"  

  . </thou  

ght>execute>  

= """ SEL  

< ECTCity FROM  

  ' 5'      ?  

"""  

# conn  

= conn.execute( ).fetchall()  

</>  

[(' ', ' , ), (' ', ' , )]  

<   > [" ", " ", " "] </>  

---  

:  

  ( 5 ):  

    ? (m) (Hz) ( ) ... Z (mm/s) X ? (m/s2) Y ? (m/s2) Z  

0      (m/s2)      ( ) 6.1 x 6.1 60 6 ... +-1000

```

					LLM	
1	Africa	South Africa	+10 10	+10 1 ...	+8 NaN	100 NaN
2	Asia	China	+10 60	China Academy of Building Research, Beijing 6 ...	40 +800	30/6/2010 17/7/2009
3	Asia	China	+10 20	+10 Guangzhou University 6 ...	+8 50 +1000	? 3 x 3
4	Asia	China	+26 +26 15	Nanjing University of Technology +50 3 ...	50 +500	10/7/2008 3 x 5
			+10	+10	50	?
[5]	x 17	1				

가	가	SQL
SQLite3	'conn' (SQLite3)	<execute>

```

Pandas          3
:
Given the following table (only the first 5 rows are shown):
   Pos    No        Rider      Bike    Laps       Time  Grid  Points
0     1    93    Marc Marquez    Derbi  22.0  40:46.315    1    25.0
1     2    38   Bradley Smith  Aprilia  22.0      +4.638    3    20.0
2     3    44   Pol Espargaro    Derbi  22.0      +4.996    2    16.0
3     4    11 Sandro Cortese    Derbi  22.0  +45.366    5    13.0
4     5     7   Efren Vazquez    Derbi  22.0  +45.433    8    11.0
:
5cc     4     /     ?
:
2010
'df

```

G.3.2. ()

LLM APPS (Hendrycks et al., 2021a)
가 .

<thought> "Hello World!" . </thought>.

1) Python
(:<execute> print("Hello World!") </execute>).

2) $\lim_{x \rightarrow 0} \frac{\sin x}{x}$ (solution)

< solution> A </solution>

5 . 2

³<https://pandas.pydata.org/>

16-
1

 $\nabla \vdash (0, 0) \rightarrow (1, 0) \rightarrow (1, 1) \rightarrow (2, 2)$
 $\nabla \vdash (0, 0) \rightarrow (0, 1) \rightarrow (1, 2) \rightarrow (0, 3) \rightarrow (1, 4) \rightarrow (2, 3) \rightarrow (3, 2) \rightarrow (4, 3)$
9 $\$(10, 1)$

```
stdin          stdout
```

```
<      >
:
(1)      (n_i, m_i)           max(n_i, m_i)
(k_i)      ,
```

(2) , n_i > m_i x
가 ,

```
(3)      가      (      )  가      가      ,  
2
```

```
</>
</>
#
-      = ["3", "2 2 3", "4 3 7", "10 1 9"]
#      = (      ).__ __
old_input =
=
```


q = int(() (q) i: a, b, k = ((int, () .
()) a < b : a, b, = b, a a % 2 == b % 2 != k % 2:
print(k - 2) elif(a + b) % 2 != 0: print(k - 1) : pri
nt(k)

```
#  
= old_input  
</>
```

: 168

```
<      >
:
가      ,      가      ,      가
1      (x      y      )
-1      -1
> k가
:
</>
```

```
<           >
#
input_data = ["3", "2 2 3", "4 3 7", "10 1 9"]
            = (           ).__ __
#
old_input =
            =
            #
q = int(    ()    (q)      i: a, b, k =    ( (int,    ()).
    ()) a < b      : a, b, = b, a a > k      : print(-1) elif a % 2
== b % 2 != k % 2: print(k - 2) elif (a + b) % 2 != 0: print(k - 1)
            : print(k)

#
            = old_input
</           >

: 16-1

:
<thought>           :
    ↗† stdin   print
    ↗†
</           >           >

q = int(    () i      (q): a, b, k =    ( (int,    ()).
    ()) a < b      : a, b, = b, a a > k      : print(-1) elif a % 2
== b % 2 != k % 2: print(k - 2) elif (a + b) % 2 != 0: print(k - 1)
            : print(k) </           >

---
```

H. M³ToolEval CodeActAgent
§3.2 CodeActInstruct LLaMA-2 Co
deActAgent Mistral 10%
,
) . CodeActAgent (

(Touvron et al., 2023)

	가	가
	LLaMA-2	Mistral
---	[1] click_url: URL [2] go_to_previous_page: [3] scroll_down: [4] scroll_up: [5] view: [6] calculator:	[1] URL [2] go_to_previous_page() -> str [3] scroll_down() [4] scroll_up() [5] view() -> str [6] "2 + (3 * 4) calculator(expression: str) -> float
---	'<url_argument>' ': click_url(url: str) -> str ': url(str). ': go_to_previous_page(): go_to_previous_page() -> str ': scroll_down(): scroll_down() ': scroll_up(): scroll_up() ': view(): view() ': calculator(expression: str) -> float	'Action: tool_name(argument_1)' 'End Action'
---	Python 'Python' 'End Action'	for-loop, if-statements 'Action:'
---	'Legendary Wand' (: 12.34).	'xx.xx' 'tool_name' '<tool_name>' '>' '.'
---	LLM --- : 12.34 --- USER --- LLM --- : 123.45	(:) --- USER --- (:) --- LLM --- : 1234.56 sierp