Massachusetts Institute of Technology

Subconscious

Beyond Context Limits

Subconscious Threads for Long-horizon Reasoning

Hongyin Luo, August 21

hongyin@subconscious.dev

Massachusetts Institute of Technology

Subconscious

Outline

- Motivation
- Working memory, thread, and TIM model
- TIMRUN inference engine
- Experiments
- Limitations

Massachusetts Institute of Technology

Subconscious

Motivation

Challenges around agent building

- Complexity in agent frameworks
- Redundant computation and repeated payments
- Decreasing throughput and performance

Massachusetts Institute of Technology

Subconscious

Motivation

Challenges around agent building

- Complexity in agent frameworks
- Redundant computation and repeated payments
- Decreasing throughput and performance

Language models have their context limits.

Engineering in chaining BERT (2019) -> LLMs (2024) together for complex tasks

“We tried a bunch of different multi-agent frameworks and landed on our own.”

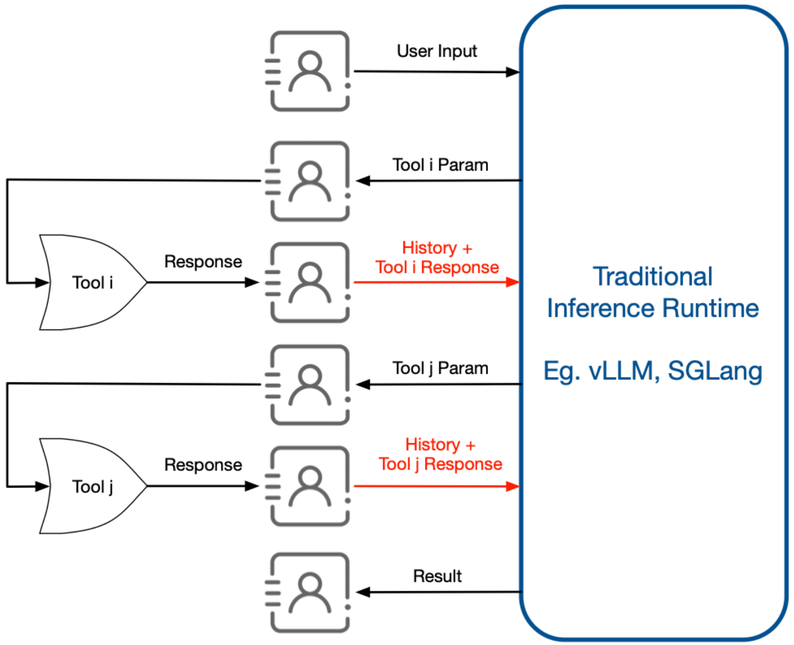
Massachusetts Institute of Technology

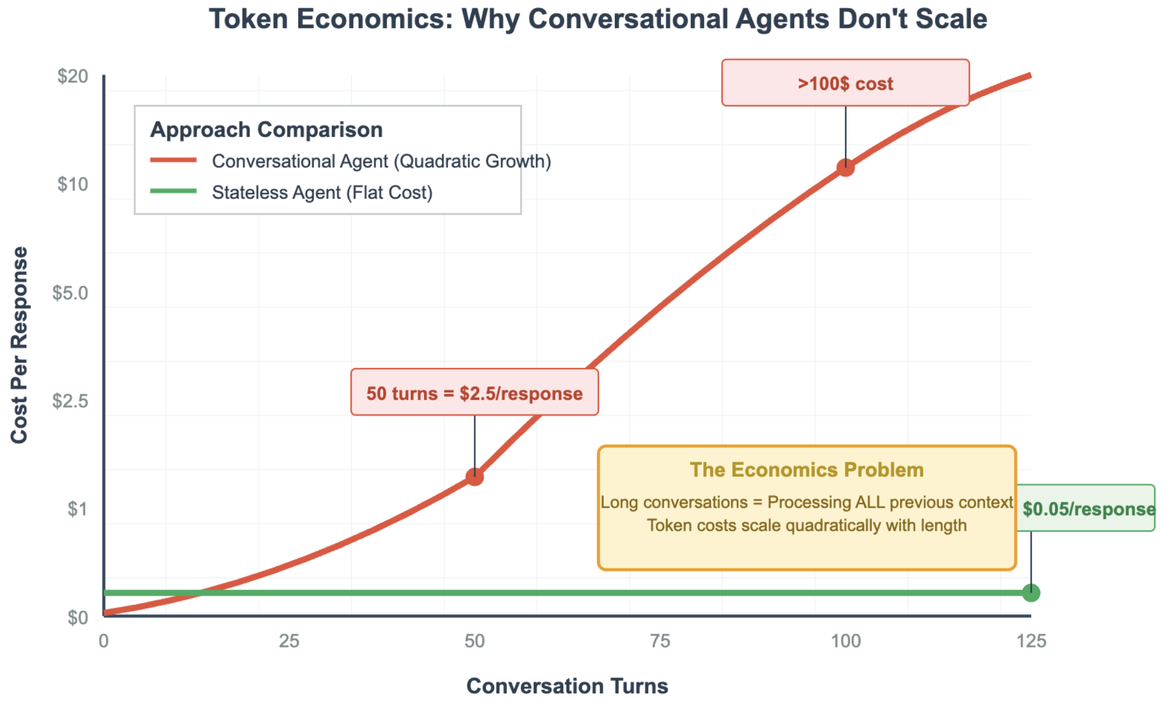
Subconscious

Motivation

Challenges around agent building

- Complexity in agent frameworks
- Redundant computation and repeated payments
- Decreasing throughput and performance





Massachusetts Institute of Technology

Subconscious

Motivation

Challenges around agent building

- Complexity in agent frameworks
- Redundant computation and repeated payments
- Decreasing throughput and performance

- Long context hallucination
- Compute complexity

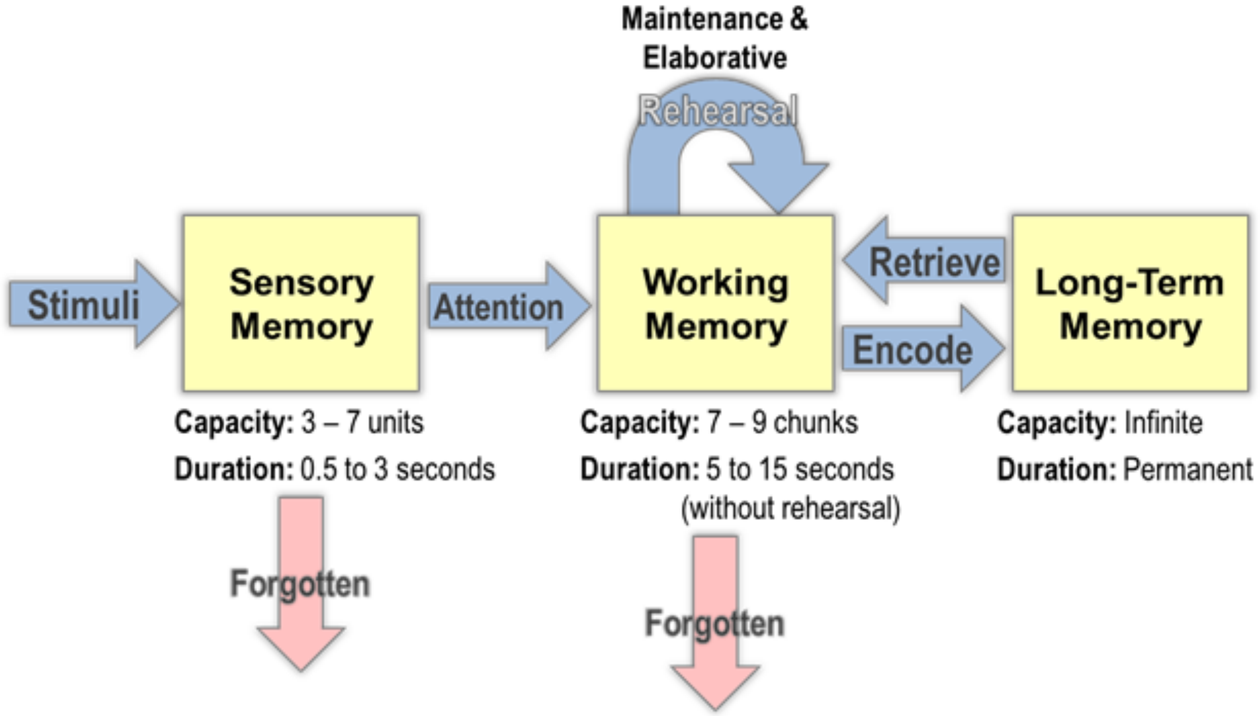
Massachusetts Institute of Technology

Subconscious

Working Memory

Basic concept behind Thread

- Subset of global memory
- Most careful computation only applied to working memory
- Interface with other modules



When we take a phone call, we don’t remember how our muscles were moving to pick up the phone.

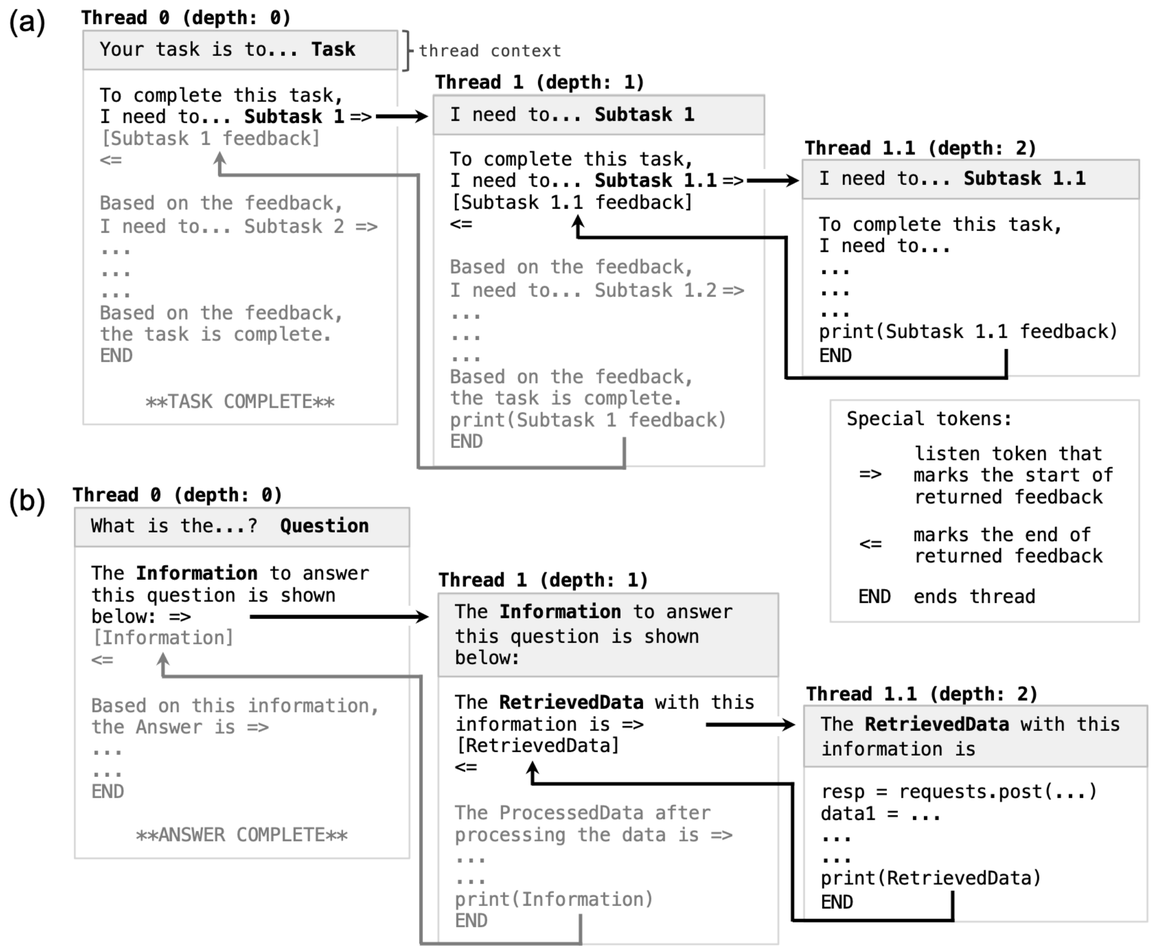
Massachusetts Institute of Technology

Subconscious

Background: Thread

Think longer and deeper

- Task-specific prompting (>4k tokens)
- No context sharing
- Information loss
- Complicated special token config
- Difficult to deploy



Massachusetts Institute of Technology

Subconscious

TIM LLM

Thread Inference Model

- Simple instruction
- Continuous state management
- Recursive JSON generation
- Straightforward structure and tool parameter extraction

	Thread	**TIM**
Structure Control	Special Tokens	JSON
State Management	No	Working Memory
Task General	No	Yes
Tool call	Developer	Server

JSON Generation

via Constrained decoding

- Define a Pydantic class
- Send via “text_format”
- LLM returns valid JSON dictionaries follows the defined format.

Getting a structured response

python

```python
from openai import OpenAI
from pydantic import BaseModel

client = OpenAI()

class CalendarEvent(BaseModel):
    name: str
    date: str
    participants: list[str]

response = client.responses.parse(
    model="gpt-4o-2024-08-06",
    input=[
        {"role": "system", "content": "Extract the event information."},
        {
            "role": "user",
            "content": "Alice and Bob are going to a science fair on Friday.",
        },
    ],
    text_format=CalendarEvent,
)

event = response.output_parsed
```

Massachusetts Institute of Technology

Subconscious

Generating Recursive JSON

JSON mode handles this, too!

```python
class CalendarEvent(BaseModel):
    name: str
    date: str
    participants: List[str]
    related_events: List['CalendarEvent']

CalendarEvent.model_rebuild()
```


```json
{
  "name": "AI Symposium 2025",
  "date": "2025-09-15",
  "participants": ["Dr. Alice Lee", "Prof. Mark Brown"],
  "related_events": [
    {
      "name": "AI Workshop: Large Language Models",
      "date": "2025-09-14",
      "participants": ["Dr. Alice Lee", "Chris Wong"],
      "related_events": [
        {
          "name": "LLM Paper Reading Group",
          "date": "2025-09-13",
          "participants": ["Chris Wong", "Samantha Lin"],
          "related_events": []
        }
      ]
    },
    {
      "name": "AI Symposium Closing Ceremony",
      "date": "2025-09-15",
      "participants": ["Prof. Mark Brown", "Julia Yu"],
      "related_events": []
    }
  ]
}
```

Massachusetts Institute of Technology

Subconscious

Thread in JSON

More flexibility, less prompting

Recursion Hierarchy

```python
class TimResponse(BaseModel):
    reasoning: List[Task]
    answer: str

class Task(BaseModel):
    thought: str
    tooluse: Optional[ToolUse] = None
    subtasks: Optional[List['Task']] = None
    conclusion: str
```

Reasoning process

Massachusetts Institute of Technology

Subconscious

Thread in JSON

More flexibility, less prompting

Recursion Hierarchy

```python
class TimResponse(BaseModel):
    reasoning: List[Task]
    answer: str

class Task(BaseModel):
    thought: str
    tooluse: Optional[ToolUse] = None
    subtasks: Optional[List['Task']] = None
    conclusion: str
```

Step-level reasoning

Massachusetts Institute of Technology

Subconscious

Thread in JSON

More flexibility, less prompting

Recursion Hierarchy

```python
class TimResponse(BaseModel):
    reasoning: List[Task]
    answer: str

class Task(BaseModel):
    thought: str
    tooluse: Optional[ToolUse] = None
    subtasks: Optional[List['Task']] = None
    conclusion: str
```

Subconscious Threads

Massachusetts Institute of Technology

Subconscious

Thread in JSON

More flexibility, less prompting

Recursion Hierarchy

```
class TimResponse(BaseModel):
    reasoning: List[Task]
    answer: str

class Task(BaseModel):
    thought: str
    tooluse: Optional[ToolUse] = None
    subtasks: Optional[List['Task']] = None
    conclusion: str
```

Tool use

Massachusetts Institute of Technology

Subconscious

Thread in JSON

More flexibility, less prompting

Recursion Hierarchy

```python
class TimResponse(BaseModel):
    reasoning: List[Task]
    answer: str

class Task(BaseModel):
    thought: str
    tooluse: Optional[ToolUse] = None
    subtasks: Optional[List['Task']] = None
    conclusion: str
```

Tool Use Schema

```python
class ToolUse(BaseModel):
    tool_name: Literal[ “SearchTool”, “WebReaderTool” ]
    parameters: Union[ SearchTool, WebReaderTool ]
    tool_result: dict

class SearchTool(BaseModel):
    query: str

class WebReaderTool(BaseModel):
    goal: str
    url: str
```

Massachusetts Institute of Technology

Subconscious

Thread in JSON

More flexibility, less prompting

Tools are called right after this field is generated

Tool Use Schema

```python
class ToolUse(BaseModel):
    tool_name: Literal[ “SearchTool”, “WebReaderTool” ]
    parameters: Union[ SearchTool, WebReaderTool ]
    tool_result: dict

class SearchTool(BaseModel):
    query: str

class WebReaderTool(BaseModel):
    goal: str
    url: str
```

Massachusetts Institute of Technology

Subconscious

Thread in JSON

More flexibility, less prompting

JSON grammar engine accepts any valid tool responses as dictionaries

Tool Use Schema

```
class ToolUse(BaseModel):
    tool_name: Literal[ “SearchTool”, “WebReaderTool” ]
    parameters: Union[ SearchTool, WebReaderTool ]
    tool_result: dict

class SearchTool(BaseModel):
    query: str

class WebReaderTool(BaseModel):
    goal: str
    url: str
```

Massachusetts Institute of Technology

Subconscious

TIM Outputs

More flexibility, less prompting


```
{
  "reasoning": [
    {
      "thought": "First, I need to identify the type of function given: f(x) = x^3 * sin(x). This is a product of two functions, so I should use the product rule for differentiation.",
      "subtasks": [
        {
          "thought": "Recall the product rule: if f(x) = u(x) * v(x), then f'(x) = u'(x) * v(x) + u(x) * v'(x). I need to choose u(x) and v(x) for this problem.",
          "subtasks": [
            {
              "thought": "Let u(x) = x^3 and v(x) = sin(x). This splits the function into two parts suitable for the product rule.",
              "subtasks": [
                {
                  "thought": "Now, I need to compute the derivative of u(x) = x^3. Using the power rule, the derivative of x^n is n*x^{n-1}.",
                  "subtasks": [
                    {
                      "thought": "Apply the power rule to x^3: derivative is 3x^2.",
                      "conclusion": "u'(x) = 3x^2.",
                      "title": "Derivative of x^3"
                    },
                    {
                      "thought": "Check if there are any additional terms or constants to consider in u(x). Since u(x) = x^3, there are none.",
                      "conclusion": "No additional terms in u(x).",
                      "title": "Check u(x) for extra terms"
                    },
                    {
                      "thought": "Summarize the result for u'(x): u'(x) = 3x^2.",
                      "conclusion": "u'(x) = 3x^2.",
                      "title": "Summarize u'(x)"
                    }
                  ],
```

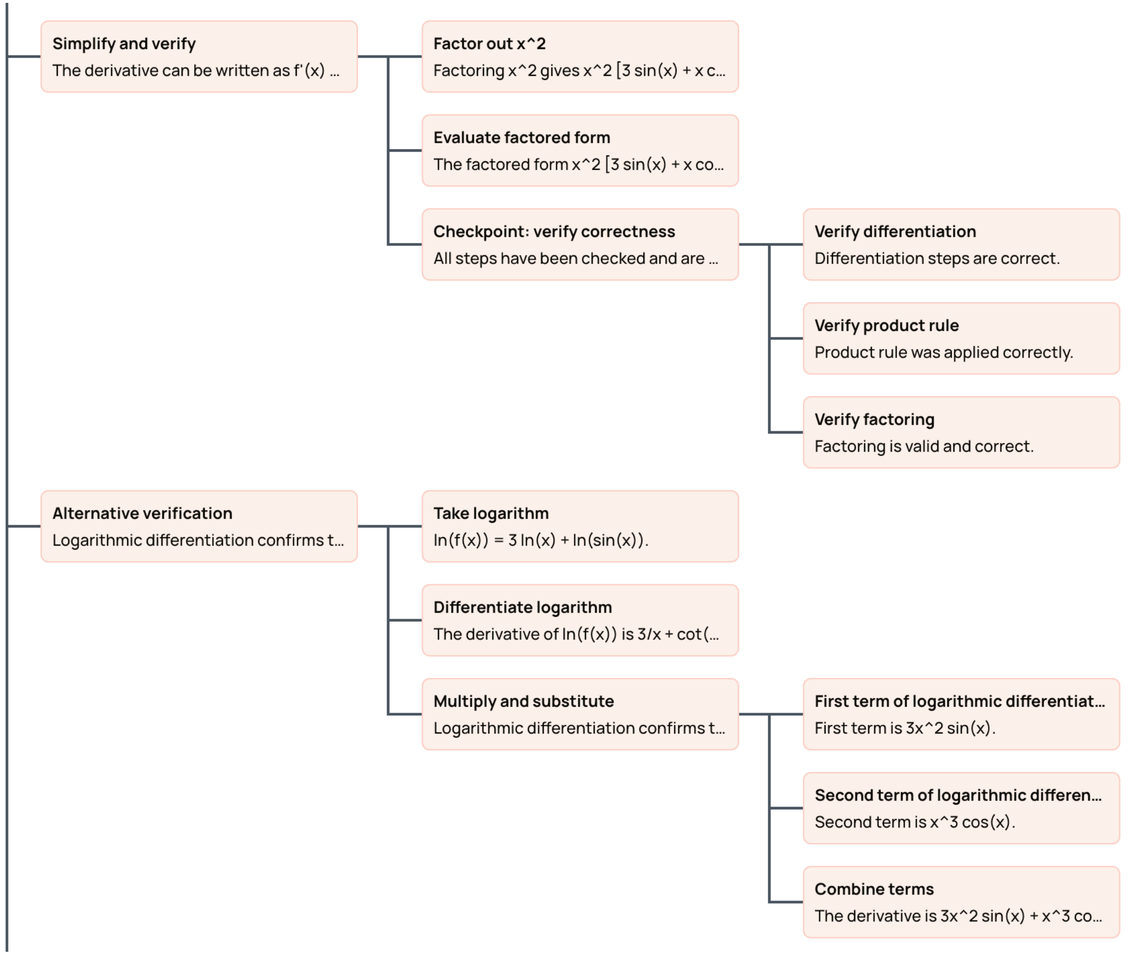
Massachusetts Institute of Technology

Subconscious

TIM Outputs - Tree View

More flexibility, less prompting

- Automatic workflow generation
- Critical path tracking
- Enabling subtask pruning

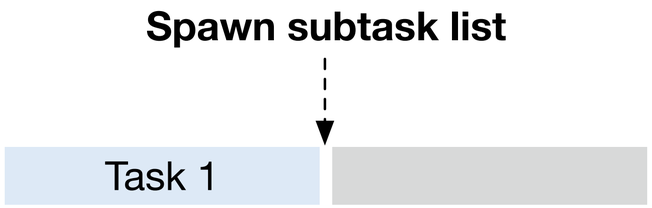


Massachusetts Institute of Technology

Subconscious

TIM Reasoning Breakdown

Recursive task decomposition and aggregation



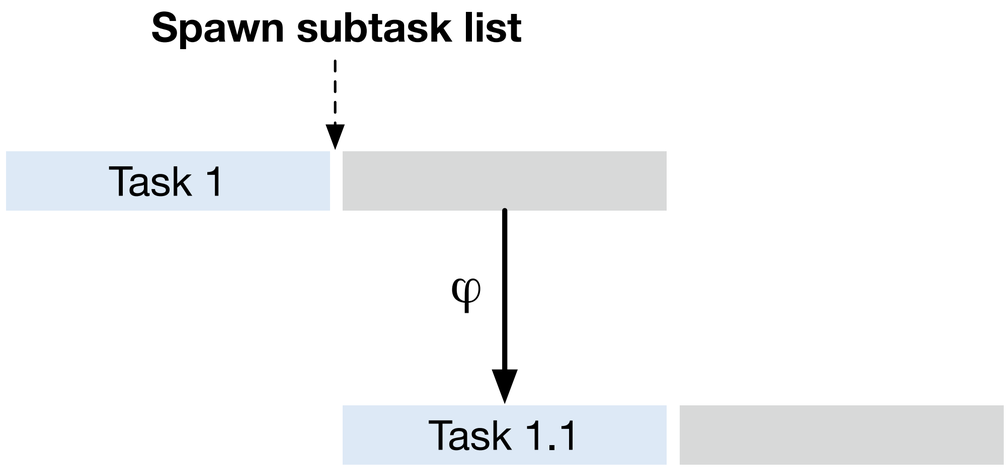
Thinking | ToolUse | Subtasks | Conclusion

Massachusetts Institute of Technology

Subconscious

TIM Reasoning Breakdown

Recursive task decomposition and aggregation



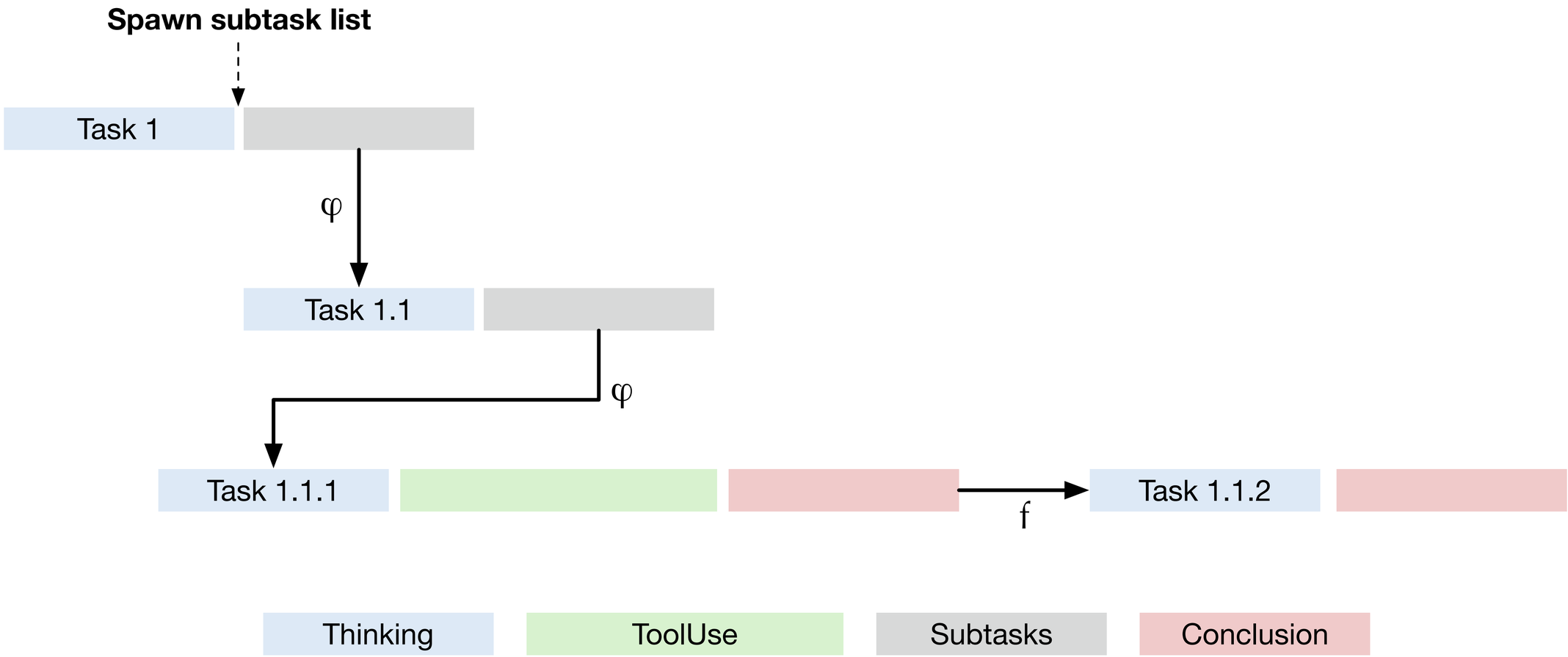
Thinking | ToolUse | Subtasks | Conclusion

Massachusetts Institute of Technology

Subconscious

TIM Reasoning Breakdown

Recursive task decomposition and aggregation

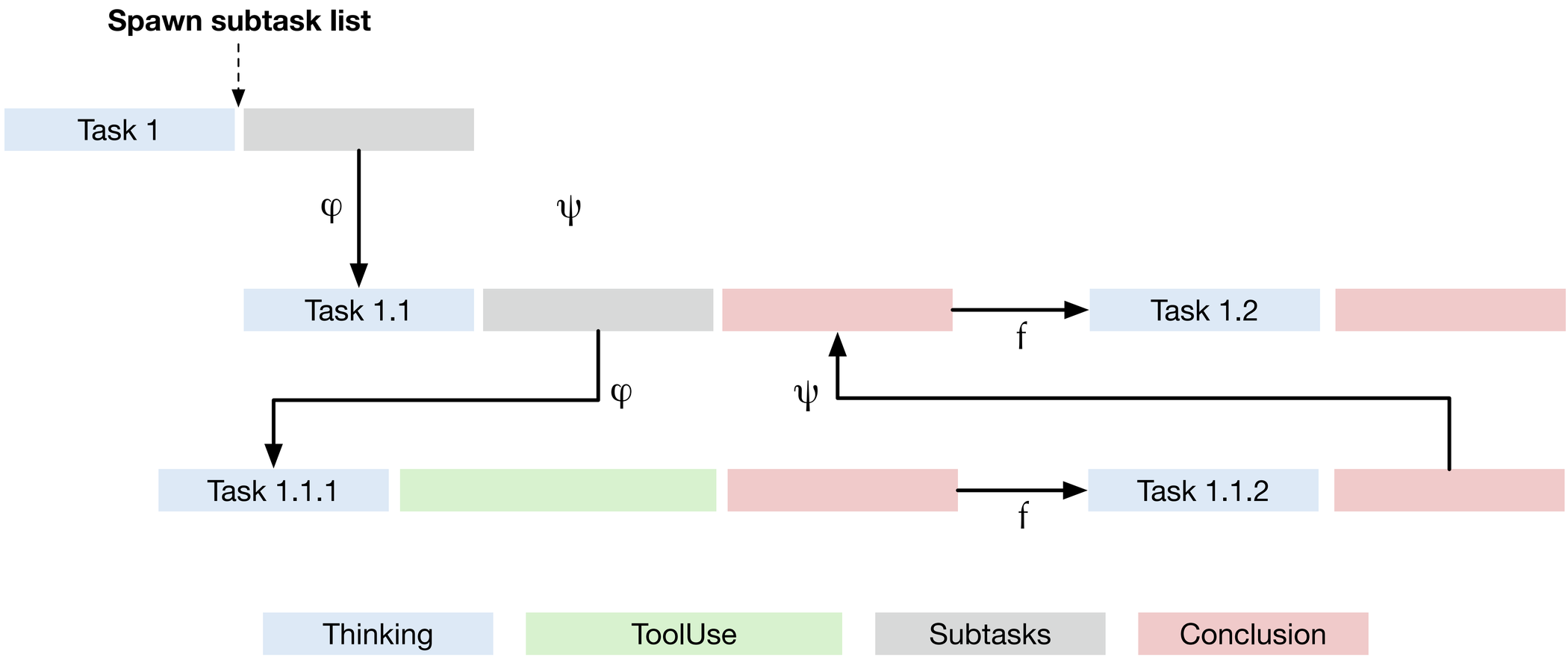


Massachusetts Institute of Technology

Subconscious

TIM Reasoning Breakdown

Recursive task decomposition and aggregation

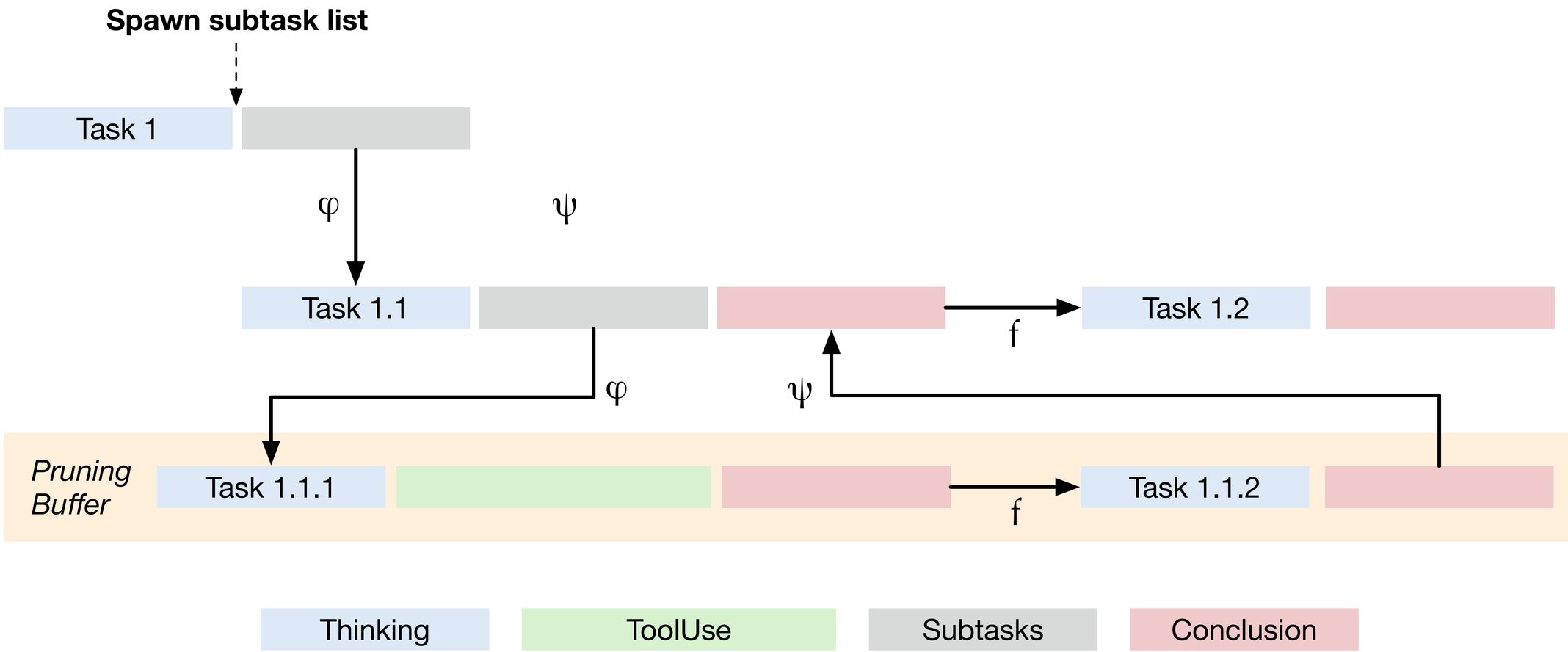


Massachusetts Institute of Technology

Subconscious

TIM Reasoning Breakdown

Recursive task decomposition and aggregation

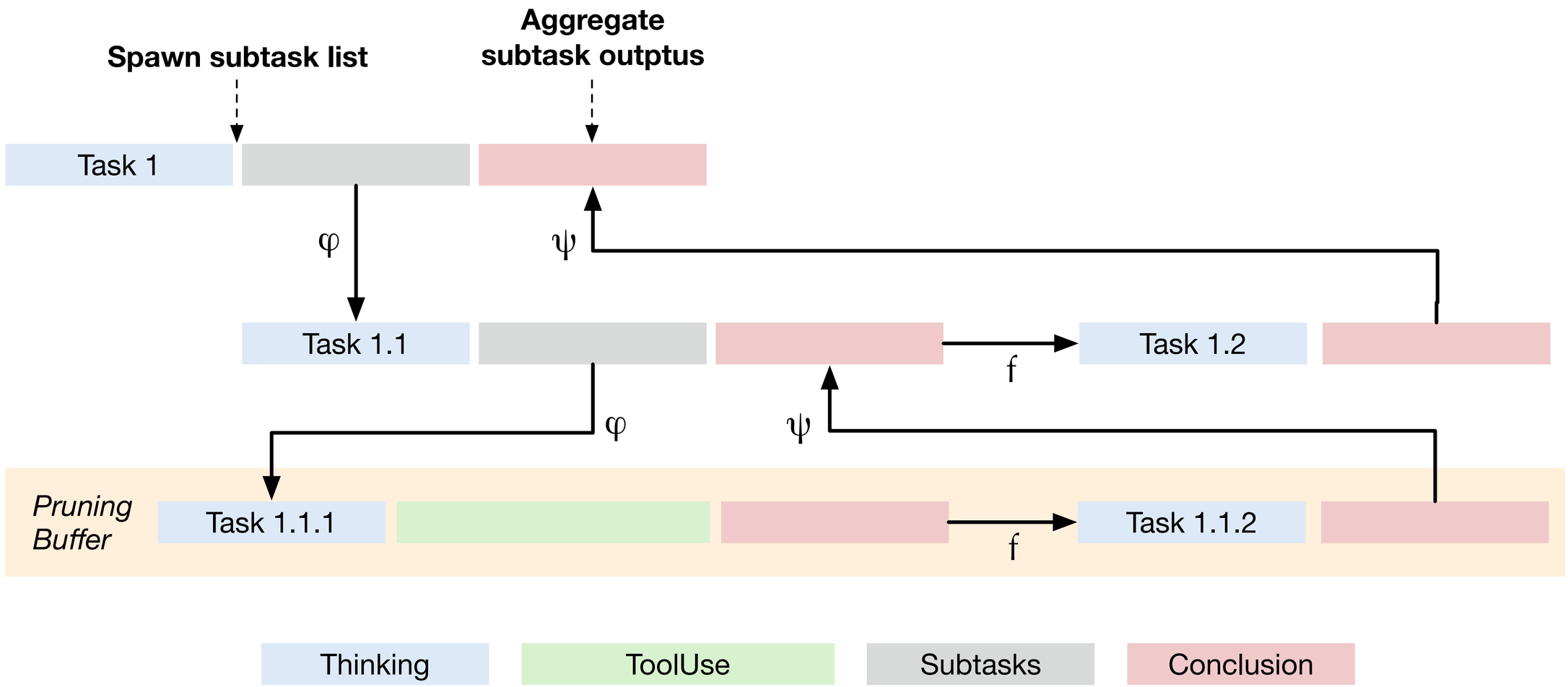


Massachusetts Institute of Technology

Subconscious

TIM Reasoning Breakdown

Recursive task decomposition and aggregation

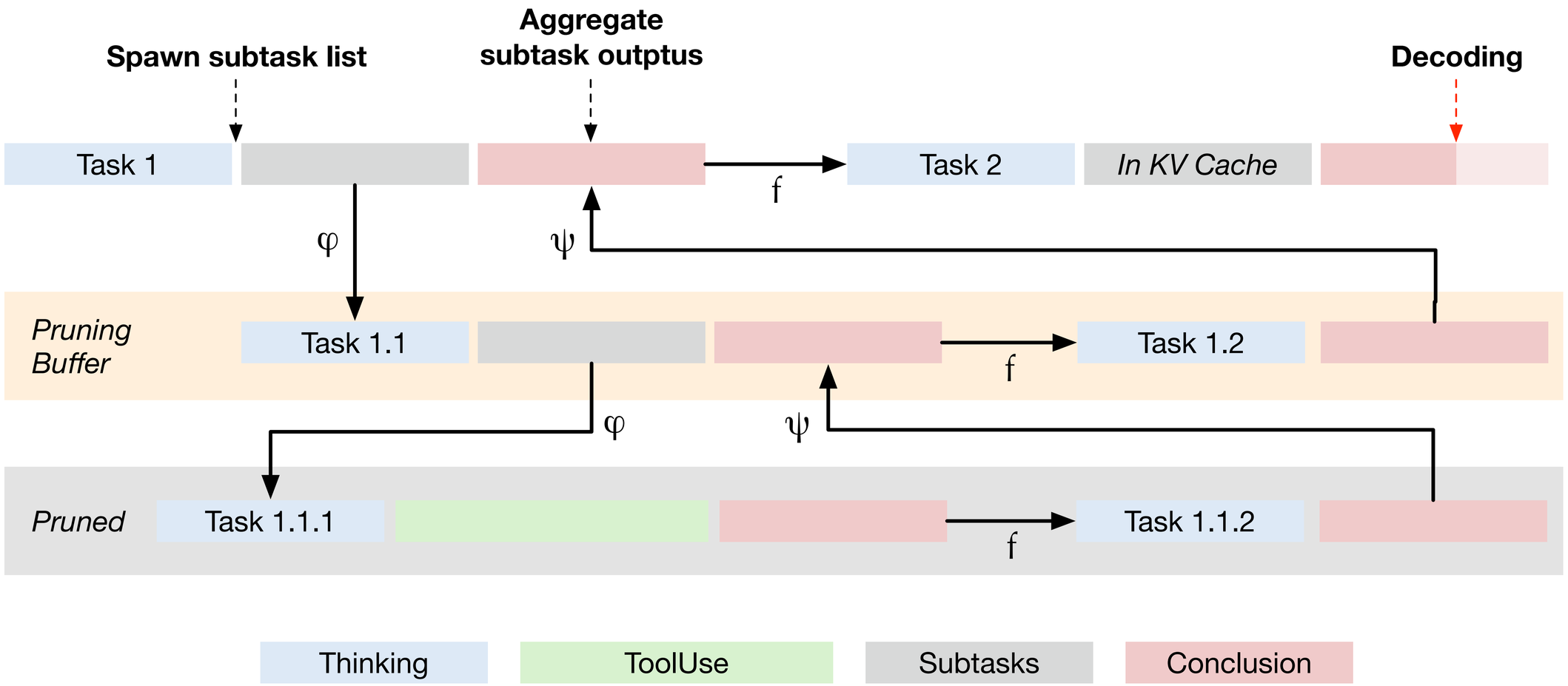


Massachusetts Institute of Technology

Subconscious

Subtask Pruning

A rule-based approach



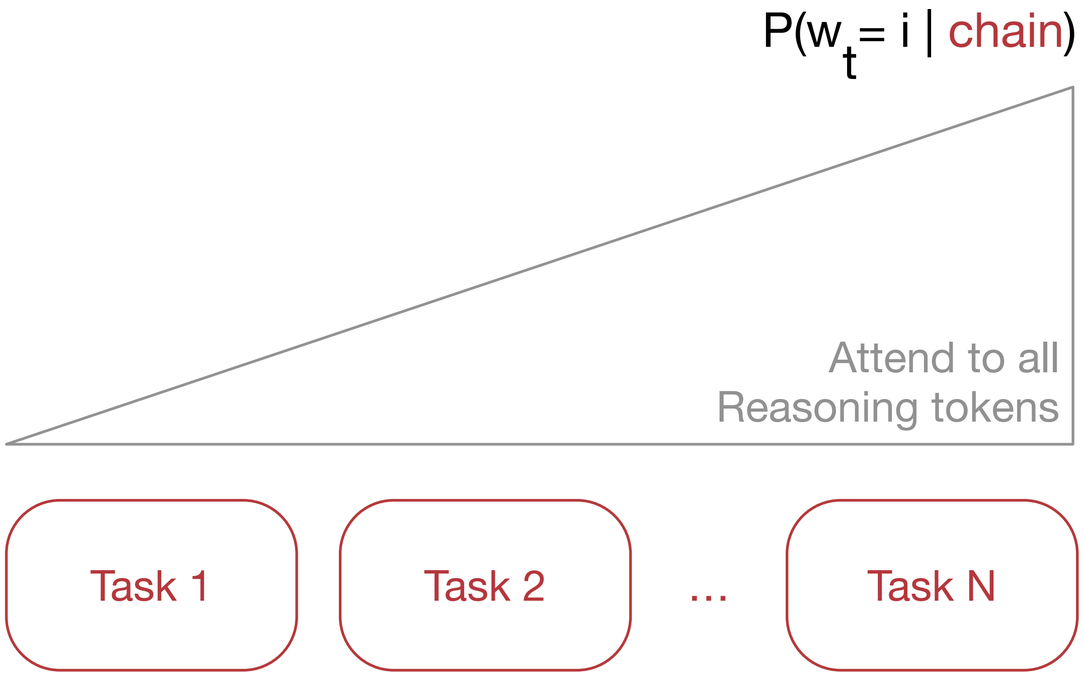
Massachusetts Institute of Technology

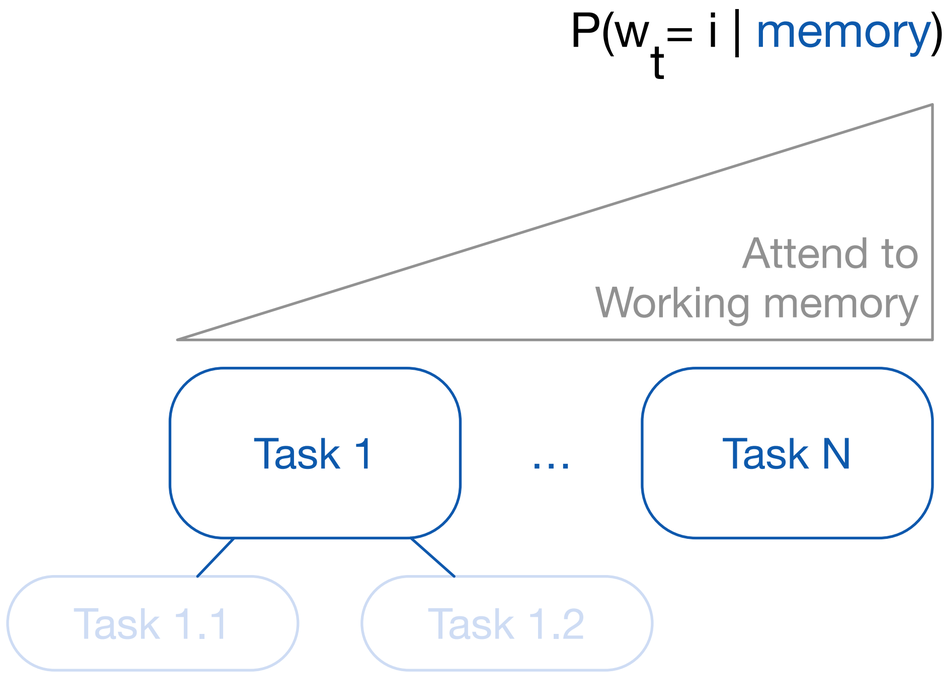
Subconscious

TIMRUN

Co-designed inference engine for TIM

1. Efficient subtask pruning; 2. Working memory attention; 3. Tool calling





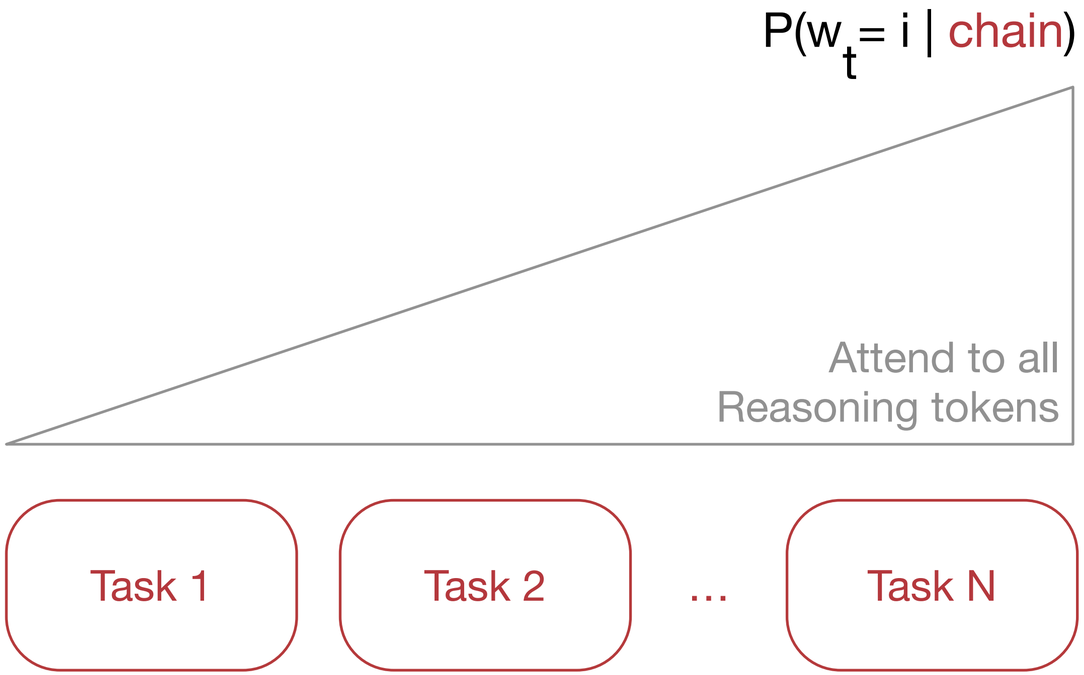
Massachusetts Institute of Technology

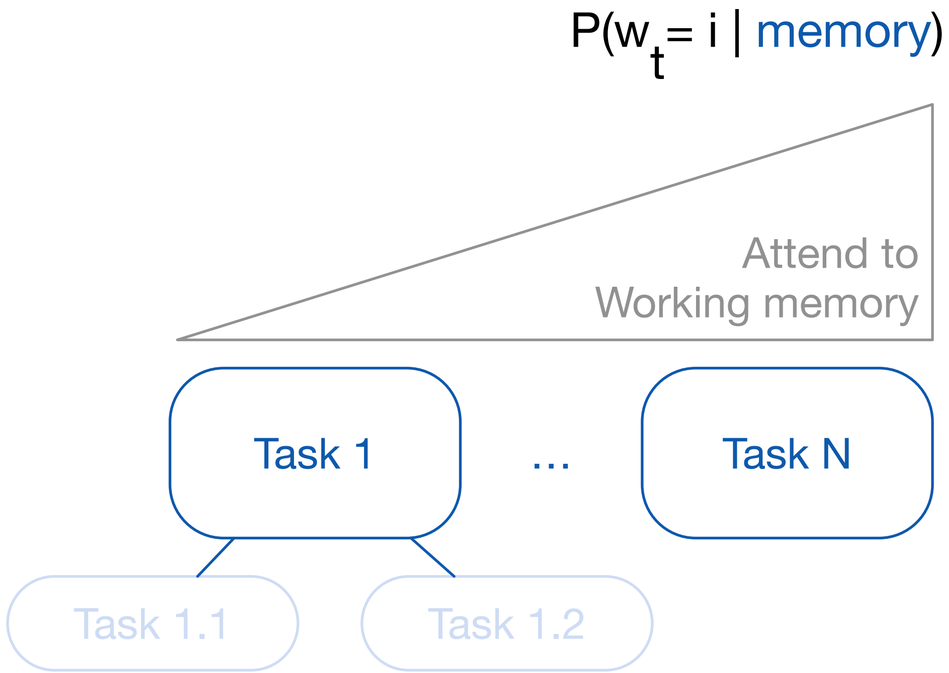
Subconscious

TIMRUN

Co-designed inference engine for TIM

1. Efficient subtask pruning; 2. Working memory attention; 3. Tool calling



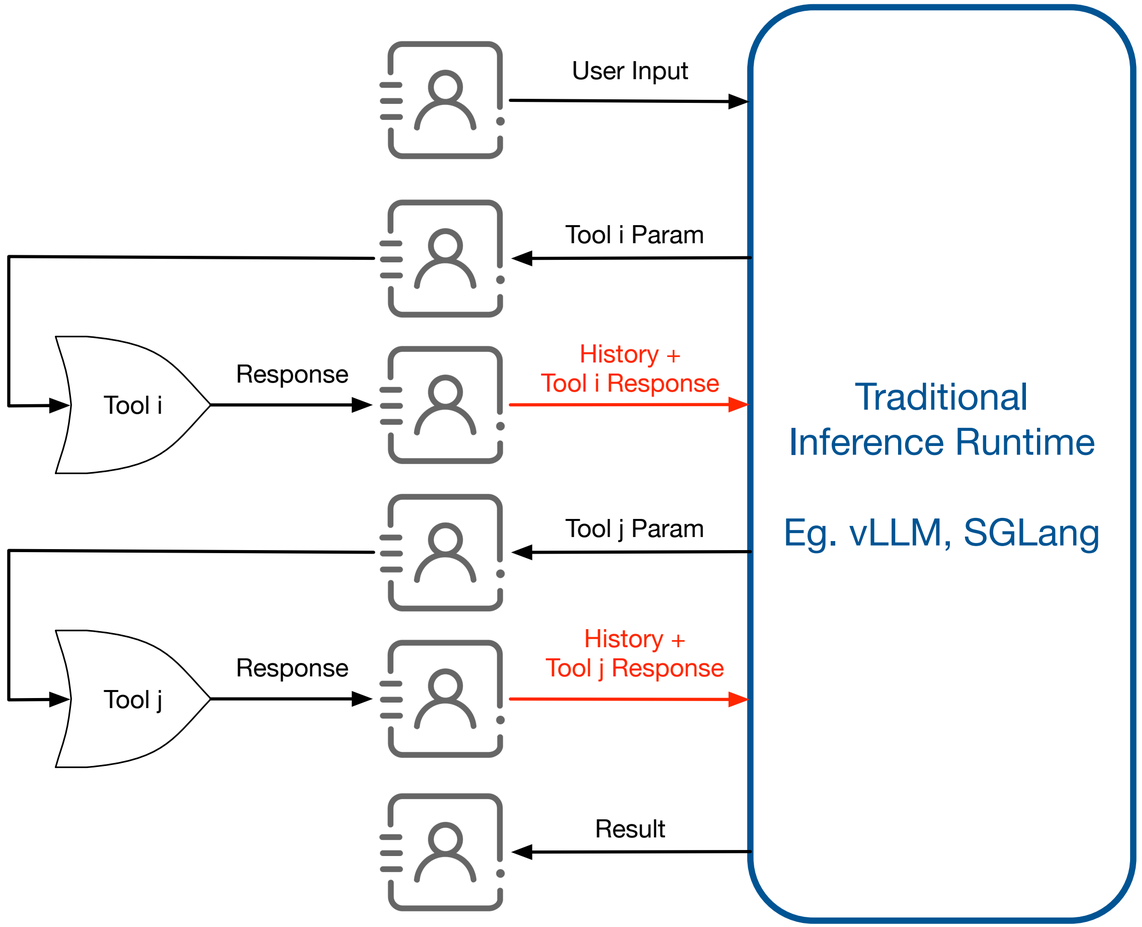


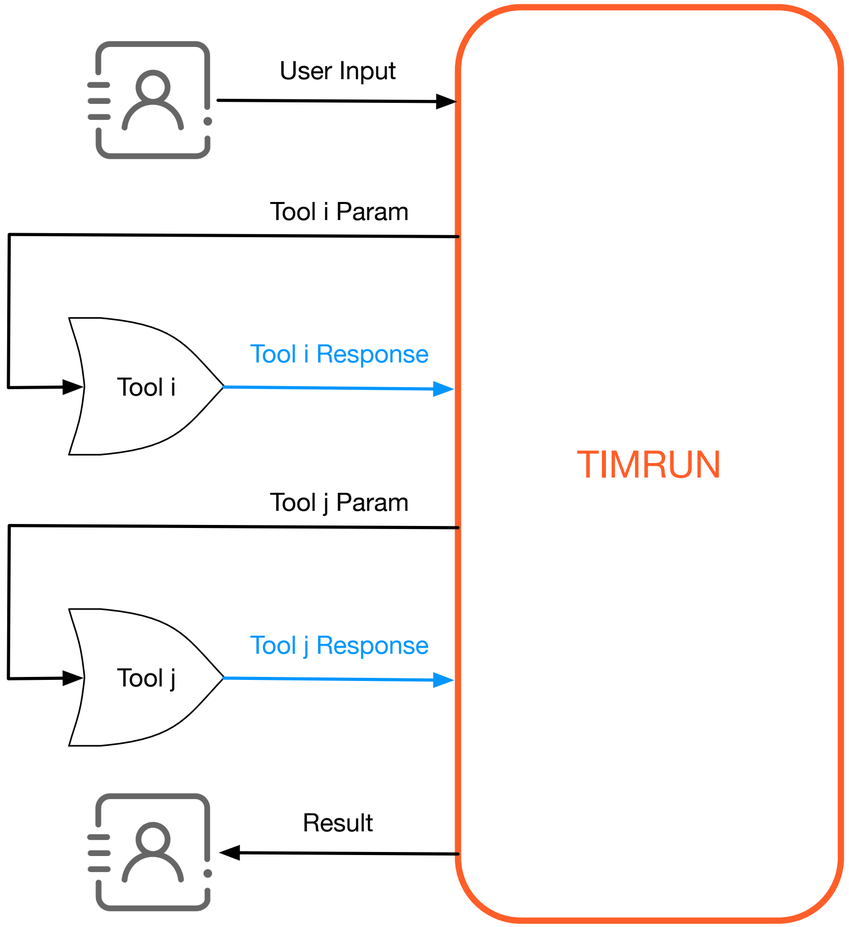
Massachusetts Institute of Technology

Subconscious

TIMRUN vs Others

End-to-end, multi-hop tool use





Massachusetts Institute of Technology

Subconscious

Impact on Agent dev

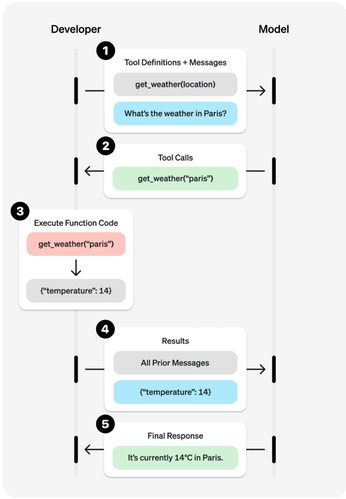
Agent inference instead of agent loop

Complex Agent Dev

Multi-Agents

Limited Context

Back and forth Tool calls



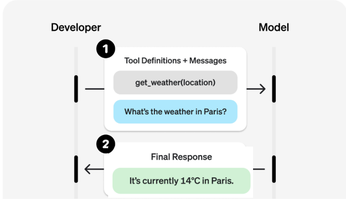
Redundant / Missing Context In Reasoning

Agent Dev as LLM Call

Single-Agents

Unlimited Context

End-to-End Tool calls



Intelligent Context Engineering

Massachusetts Institute of Technology

Subconscious

TIM Inference vs Agent Loop

Deep research challenges

BrowseComp

Model	Deepseek-R1	GPT-4o	TIM-large	TIM-8b
Paradigm	ReACT	Browsing	Browsing	Browsing
Success (%)	9.5	1.9	7.8	2.3

DataCommons

Method	Reflection	NLEP+ReACT	DecomP	THREAD	TIM
Accuracy	24.3	27.1	57.9	67.9	67.9

4k task-specific prompt tokens

Massachusetts Institute of Technology

Subconscious

Effect of Context Pruning

Not harming -> improving performance

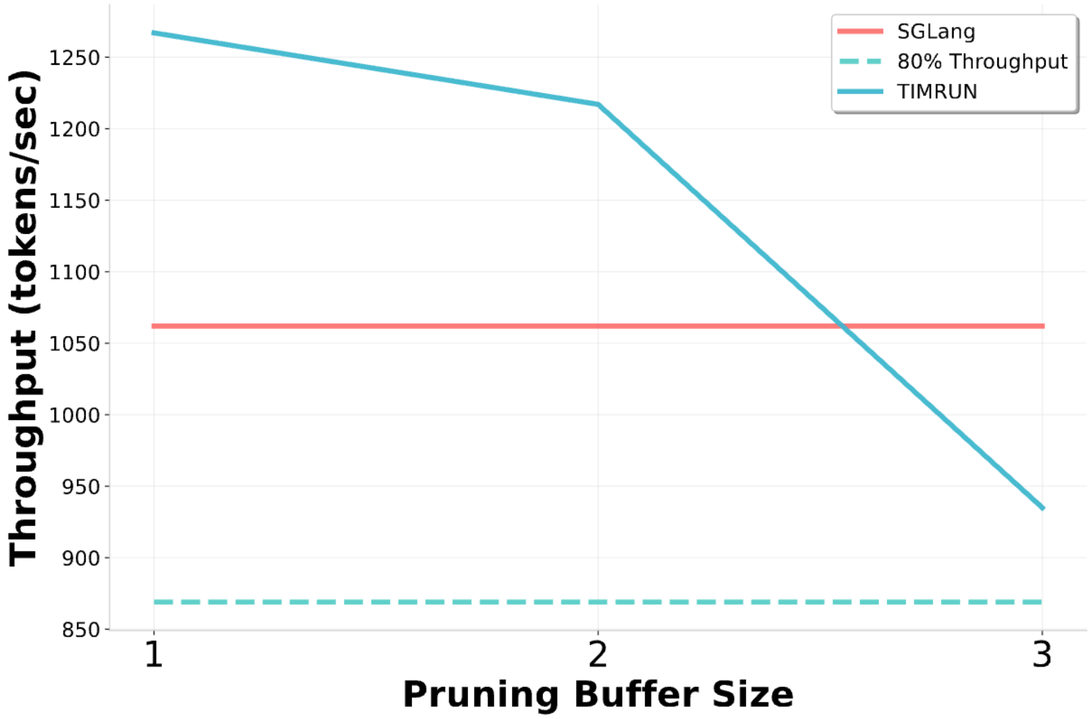
Task	TIM-8b + SGLang	TIM-8b + TIMRUN			
	Accuracy	Accuracy	Max Cache	Output Len.	KV Pruned (%)
MATH500	69.6	69.0	1569.2	3362.2	53.3
MMLUSTEM500	88.4	87.6	1330.9	2747.0	51.6
AMC 2022	60.5	60.5	2203.9	5131.7	57.1
AMC 2023	80.0	80.0	1876.5	4547.4	58.7
AIME 2024	40.0	46.7	3218.6	8974.7	64.1
GPQADiamond	44.9	48.5	1712.9	3742.6	54.2

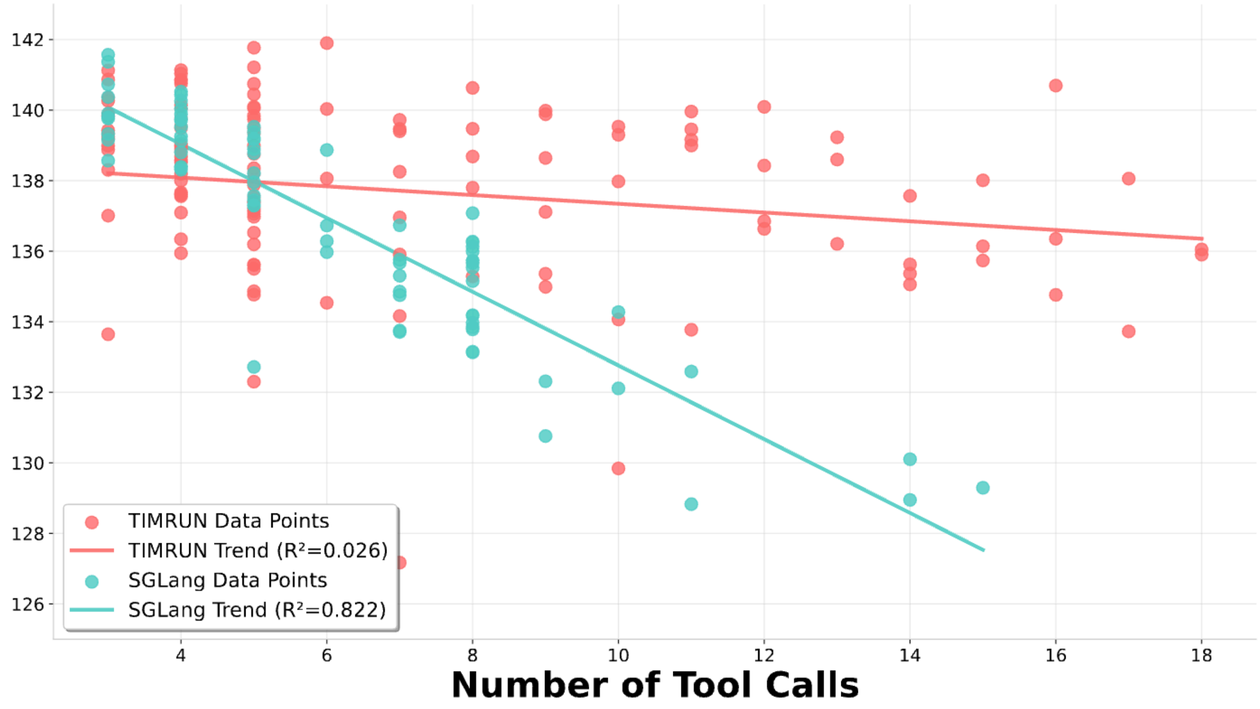
Massachusetts Institute of Technology

Subconscious

Throughputs

Subtasks pruning improves efficiency





Massachusetts Institute of Technology

Subconscious

Limitations

In reasoning and tool using

- Repeat generations
- Sensitive to prompts and sampling parameters on benchmarks
  - On AIME 2024
    - Temperature = 0.9, top_p = 0.95 -> consistent > 43% acc.
    - Temperature = 0.8 -> 23% acc
- If tool responses are tool long, still triggers context limit

TIM-8b-preview is available on Hugging Face at

SubconsciousDev/TIM-8b-preview

Try TIMRUN at Subconscious.dev

Massachusetts Institute of Technology

Subconscious

Thank you!

https://brand.mit.edu/logos-marks/tim-beaver