

SELF-DISCOVER: LLMs Self-Compose Reasoning Structures

Authors: Pei Zhou et al., 2023 (University of Southern California, Google DeepMind)

Presenter: T Jones

AI Journal Club

March 23, 2024

SELF-DISCOVER: Large Language Models Self-Compose Reasoning Structures

Pei Zhou¹ Jay Pujara¹ Xiang Ren¹ Xinyun Chen² Heng-Tze Cheng²
Quoc V. Le² Ed H. Chi² Denny Zhou² Swaroop Mishra² Huaixiu Steven Zheng²

Abstract

We introduce SELF-DISCOVER, a general framework for LLMs to self-discover the task-intrinsic reasoning structures to tackle complex reasoning problems that are challenging for typical prompting methods. Core to the framework is a self-discovery process where LLMs select multiple atomic reasoning modules such as critical thinking and step-by-step thinking, and compose them into an explicit reasoning structure for LLMs to follow during decoding. SELF-DISCOVER substantially improves GPT-4 and PaLM 2's performance on challenging reasoning benchmarks such as BigBench-Hard, grounded agent reason-

son. For example, few-shot and zero-shot chain-of-thought (CoT) (Nye et al., 2021; Wei et al., 2022; Kojima et al., 2022; Yasunaga et al., 2023) resembles how humans solve problems step-by-step, decomposition-based prompting (Zhou et al., 2022a; Drozdov et al., 2022; Patel et al., 2022; Hao et al., 2023; Khot et al., 2022) is inspired by how humans breakdown a complex problem into a series of smaller subproblems, and then solve those subproblems one by one (Polya, 2004), and step-back prompting (Zheng et al., 2023) is motivated by how humans reflect on task nature to derive general principles. However, a fundamental limitation is that each technique itself serves as an atomic reasoning module making an implicit prior assumption of the process on how to tackle a given task. Instead, we argue that each task has a unique intrinsic structure underlying

How do *you* think?

How to tie a shoe

How to refactor a
large monolithic
code base

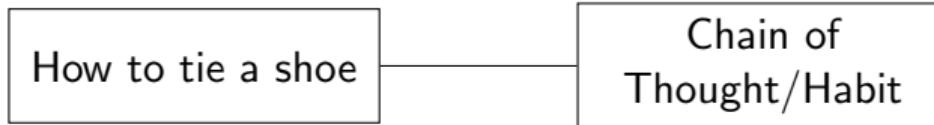
im hungry

Do you think about these
things with the exact same
mental process?

How do you think?



How do you think?



Your human brain does **not** use one single mode of thinking to solve every problem. Rather, it picks the right problem-solving mode to fit the problem.

How do LLMs “think”?

- Whether or not they actually “think” → philosophers

How do LLMs “think”?

- Whether or not they actually “think” → philosophers
- Practically, they sometimes *appear* to “think”. That is, they exhibit responses to queries that if coming from a human mind we would associate with thinking.

How do LLMs “think”?

- Whether or not they actually “think” → philosophers
- Practically, they sometimes *appear* to “think”. That is, they exhibit responses to queries that if coming from a human mind we would associate with thinking.
- Let’s put “think” in quotes to avoid the philosophers.

How do LLMs “think”?

- Whether or not they actually “think” → philosophers
- Practically, they sometimes *appear* to “think”. That is, they exhibit responses to queries that if coming from a human mind we would associate with thinking.
- Let’s put “think” in quotes to avoid the philosophers.
- The mechanics of LLM “thinking” is under active investigation, but prompting can alter the outcome of this “thinking” for better or worse.

How do LLMs “think”?

- Whether or not they actually “think” → philosophers
- Practically, they sometimes *appear* to “think”. That is, they exhibit responses to queries that if coming from a human mind we would associate with thinking.
- Let’s put “think” in quotes to avoid the philosophers.
- The mechanics of LLM “thinking” is under active investigation, but prompting can alter the outcome of this “thinking” for better or worse.
- Can we empower LLMs to pick their own prompts for “thinking” about a problem?

Motivation (1)

- LLMs have achieved remarkable performance in language generation and instruction-following tasks

Motivation (1)

- LLMs have achieved remarkable performance in language generation and instruction-following tasks
- However, complex multi-step reasoning remains a challenge

Motivation (1)

- LLMs have achieved remarkable performance in language generation and instruction-following tasks
- However, complex multi-step reasoning remains a challenge
 - Struggles in tasks requiring logical deduction, causal reasoning, mathematical problem-solving

Motivation (1)

- LLMs have achieved remarkable performance in language generation and instruction-following tasks
- However, complex multi-step reasoning remains a challenge
 - Struggles in tasks requiring logical deduction, causal reasoning, mathematical problem-solving
- Existing prompting techniques make strong assumptions about the reasoning process

Motivation (1)

- LLMs have achieved remarkable performance in language generation and instruction-following tasks
- However, complex multi-step reasoning remains a challenge
 - Struggles in tasks requiring logical deduction, causal reasoning, mathematical problem-solving
- Existing prompting techniques make strong assumptions about the reasoning process
 - Direct Answer (assumes simple Q&A)

Motivation (1)

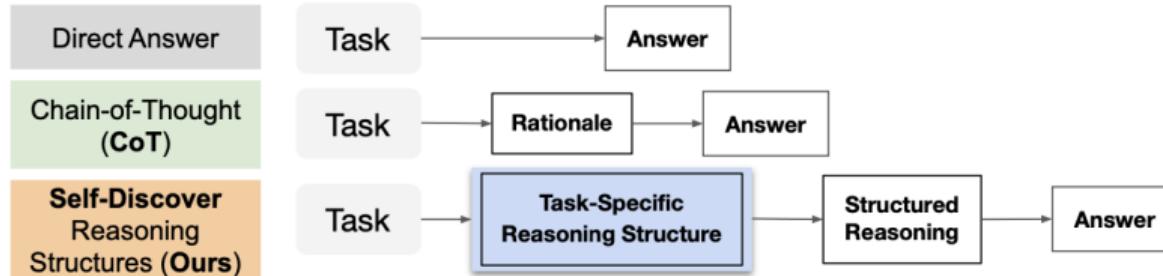
- LLMs have achieved remarkable performance in language generation and instruction-following tasks
- However, complex multi-step reasoning remains a challenge
 - Struggles in tasks requiring logical deduction, causal reasoning, mathematical problem-solving
- Existing prompting techniques make strong assumptions about the reasoning process
 - Direct Answer (assumes simple Q&A)
 - Chain-of-Thought (CoT) assumes step-by-step reasoning

Motivation (1)

- LLMs have achieved remarkable performance in language generation and instruction-following tasks
- However, complex multi-step reasoning remains a challenge
 - Struggles in tasks requiring logical deduction, causal reasoning, mathematical problem-solving
- Existing prompting techniques make strong assumptions about the reasoning process
 - Direct Answer (assumes simple Q&A)
 - Chain-of-Thought (CoT) assumes step-by-step reasoning
 - Least-to-Most assumes problem decomposition

Motivation (1)

- LLMs have achieved remarkable performance in language generation and instruction-following tasks
- However, complex multi-step reasoning remains a challenge
 - Struggles in tasks requiring logical deduction, causal reasoning, mathematical problem-solving
- Existing prompting techniques make strong assumptions about the reasoning process
 - Direct Answer (assumes simple Q&A)
 - Chain-of-Thought (CoT) assumes step-by-step reasoning
 - Least-to-Most assumes problem decomposition



Motivation (2)

- Key Insight: Each task has an intrinsic reasoning structure for efficient problem-solving

Motivation (2)

- Key Insight: Each task has an intrinsic reasoning structure for efficient problem-solving
 - For example, a logical deduction → sequence of premise-conclusion steps

Motivation (2)

- Key Insight: Each task has an intrinsic reasoning structure for efficient problem-solving
 - For example, a logical deduction → sequence of premise-conclusion steps
 - A causal reasoning → identifying cause-effect relationships and counterfactuals

Motivation (2)

- Key Insight: Each task has an intrinsic reasoning structure for efficient problem-solving
 - For example, a logical deduction → sequence of premise-conclusion steps
 - A causal reasoning → identifying cause-effect relationships and counterfactuals

Motivation (3)

- Uncovering the intrinsic reasoning structure can guide the model to solve the task more effectively

Motivation (3)

- Uncovering the intrinsic reasoning structure can guide the model to solve the task more effectively
 - It provides a blueprint for the model to follow during problem-solving

Motivation (3)

- Uncovering the intrinsic reasoning structure can guide the model to solve the task more effectively
 - It provides a blueprint for the model to follow during problem-solving
 - It can reduce the search space of possible solutions

Motivation (3)

- Uncovering the intrinsic reasoning structure can guide the model to solve the task more effectively
 - It provides a blueprint for the model to follow during problem-solving
 - It can reduce the search space of possible solutions
 - It enhances the interpretability of the model's problem-solving process

Motivation (3)

- Uncovering the intrinsic reasoning structure can guide the model to solve the task more effectively
 - It provides a blueprint for the model to follow during problem-solving
 - It can reduce the search space of possible solutions
 - It enhances the interpretability of the model's problem-solving process

Results

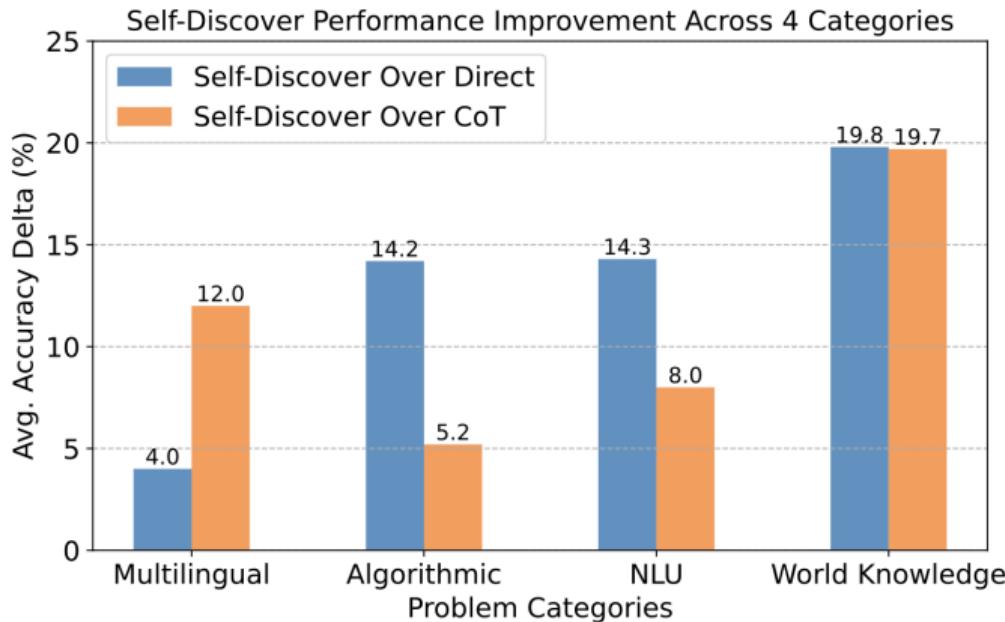


Figure 4. Breakdown of SELF-DISCOVER performance improvement on 4 categories on PaLM 2-L. SELF-DISCOVER performs the best on tasks requiring world knowledge.

Key Idea of SELF-DISCOVER

- SELF-DISCOVER enables LLMs to uncover task-specific reasoning structures

Key Idea of SELF-DISCOVER

- SELF-DISCOVER enables LLMs to uncover task-specific reasoning structures
 - Allows the model to adapt its reasoning to the intrinsic structure of the task

Key Idea of SELF-DISCOVER

- SELF-DISCOVER enables LLMs to uncover task-specific reasoning structures
 - Allows the model to adapt its reasoning to the intrinsic structure of the task
- The process begins by selecting from a set of atomic reasoning modules

Key Idea of SELF-DISCOVER

- SELF-DISCOVER enables LLMs to uncover task-specific reasoning structures
 - Allows the model to adapt its reasoning to the intrinsic structure of the task
- The process begins by selecting from a set of atomic reasoning modules
 - Modules represent fundamental reasoning skills (e.g. critical thinking, problem decomposition)

Key Idea of SELF-DISCOVER

- SELF-DISCOVER enables LLMs to uncover task-specific reasoning structures
 - Allows the model to adapt its reasoning to the intrinsic structure of the task
- The process begins by selecting from a set of atomic reasoning modules
 - Modules represent fundamental reasoning skills (e.g. critical thinking, problem decomposition)
 - Modules are inspired cognitive science and LLM problem-solving literature (Google DeepMind's PROMPTBREEDER set), Modules are described in natural language

Key Idea of SELF-DISCOVER

- SELF-DISCOVER enables LLMs to uncover task-specific reasoning structures
 - Allows the model to adapt its reasoning to the intrinsic structure of the task
- The process begins by selecting from a set of atomic reasoning modules
 - Modules represent fundamental reasoning skills (e.g. critical thinking, problem decomposition)
 - Modules are inspired cognitive science and LLM problem-solving literature (Google DeepMind's PROMPTBREEDER set), Modules are described in natural language
- SELF-DISCOVER composes the selected modules into a coherent reasoning structure

Key Idea of SELF-DISCOVER

- SELF-DISCOVER enables LLMs to uncover task-specific reasoning structures
 - Allows the model to adapt its reasoning to the intrinsic structure of the task
- The process begins by selecting from a set of atomic reasoning modules
 - Modules represent fundamental reasoning skills (e.g. critical thinking, problem decomposition)
 - Modules are inspired cognitive science and LLM problem-solving literature (Google DeepMind's PROMPTBREEDER set), Modules are described in natural language
- SELF-DISCOVER composes the selected modules into a coherent reasoning structure
 - Composed structure reflects the LLM's intrinsic reasoning flow of the task

Key Idea of SELF-DISCOVER

- SELF-DISCOVER enables LLMs to uncover task-specific reasoning structures
 - Allows the model to adapt its reasoning to the intrinsic structure of the task
- The process begins by selecting from a set of atomic reasoning modules
 - Modules represent fundamental reasoning skills (e.g. critical thinking, problem decomposition)
 - Modules are inspired cognitive science and LLM problem-solving literature (Google DeepMind's PROMPTBREEDER set), Modules are described in natural language
- SELF-DISCOVER composes the selected modules into a coherent reasoning structure
 - Composed structure reflects the LLM's intrinsic reasoning flow of the task
 - Performed in a few-shot setting, without requiring task-specific training data

Key Idea of SELF-DISCOVER

- SELF-DISCOVER enables LLMs to uncover task-specific reasoning structures
 - Allows the model to adapt its reasoning to the intrinsic structure of the task
- The process begins by selecting from a set of atomic reasoning modules
 - Modules represent fundamental reasoning skills (e.g. critical thinking, problem decomposition)
 - Modules are inspired cognitive science and LLM problem-solving literature (Google DeepMind's PROMPTBREEDER set), Modules are described in natural language
- SELF-DISCOVER composes the selected modules into a coherent reasoning structure
 - Composed structure reflects the LLM's intrinsic reasoning flow of the task
 - Performed in a few-shot setting, without requiring task-specific training data
- During inference, the model follows the self-discovered structure step-by-step

Key Idea of SELF-DISCOVER

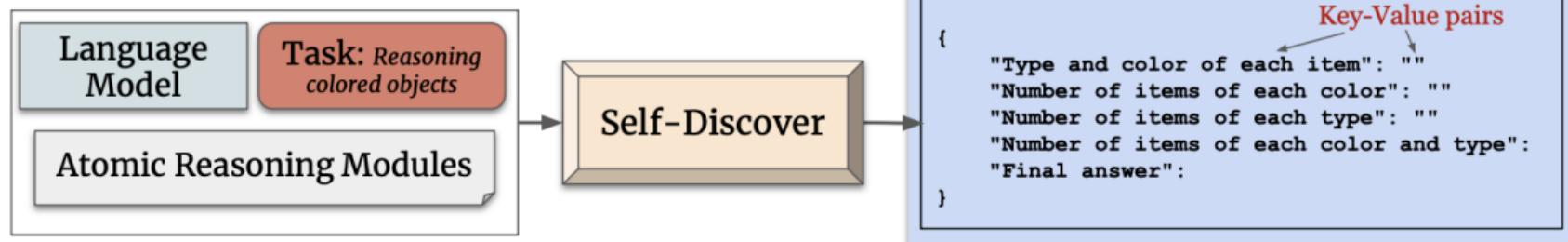
- SELF-DISCOVER enables LLMs to uncover task-specific reasoning structures
 - Allows the model to adapt its reasoning to the intrinsic structure of the task
- The process begins by selecting from a set of atomic reasoning modules
 - Modules represent fundamental reasoning skills (e.g. critical thinking, problem decomposition)
 - Modules are inspired cognitive science and LLM problem-solving literature (Google DeepMind's PROMPTBREEDER set), Modules are described in natural language
- SELF-DISCOVER composes the selected modules into a coherent reasoning structure
 - Composed structure reflects the LLM's intrinsic reasoning flow of the task
 - Performed in a few-shot setting, without requiring task-specific training data
- During inference, the model follows the self-discovered structure step-by-step
 - Each step corresponds to a selected reasoning module

Key Idea of SELF-DISCOVER

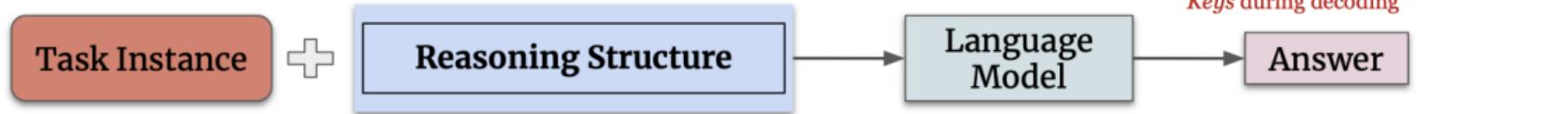
- SELF-DISCOVER enables LLMs to uncover task-specific reasoning structures
 - Allows the model to adapt its reasoning to the intrinsic structure of the task
- The process begins by selecting from a set of atomic reasoning modules
 - Modules represent fundamental reasoning skills (e.g. critical thinking, problem decomposition)
 - Modules are inspired cognitive science and LLM problem-solving literature (Google DeepMind's PROMPTBREEDER set), Modules are described in natural language
- SELF-DISCOVER composes the selected modules into a coherent reasoning structure
 - Composed structure reflects the LLM's intrinsic reasoning flow of the task
 - Performed in a few-shot setting, without requiring task-specific training data
- During inference, the model follows the self-discovered structure step-by-step
 - Each step corresponds to a selected reasoning module
 - The model applies the skills associated with each module to progressively solve the task

Overview

Stage 1: Discover Reasoning Structure on Task-Level



Stage 2: Solve Problems Using Discovered Structure on Instance-Level



Three stages

Three key actions:

- ① SELECT relevant reasoning modules
- ② ADAPT module descriptions to task
- ③ IMPLEMENT into operational key-value structure

Three stages

Three key actions:

- ① SELECT relevant reasoning modules
- ② ADAPT module descriptions to task
- ③ IMPLEMENT into operational key-value structure

SELECT

Select several reasoning modules
that are crucial to utilize in order
solve the given task:

All reasoning module descriptions

- Critical thinking: ...
- Step-by-Step: ...
- Propose and verify: ...

...

Task examples w/o answer:
Example 1: ...
Example 2: ...

Select several modules that are
crucial for solving the tasks above:

Three stages

Three key actions:

- ① SELECT relevant reasoning modules
- ② ADAPT module descriptions to task
- ③ IMPLEMENT into operational key-value structure

SELECT

Select several reasoning modules that are crucial to utilize in order solve the given task:

All reasoning module descriptions

- Critical thinking: ...
- Step-by-Step: ...
- Propose and verify: ...
- ...

Task examples w/o answer:
Example 1: ...
Example 2: ...

Select several modules that are crucial for solving the tasks above:

ADAPT

Rephrase and specify each reasoning module so that it better helps solving the task:

SELECTED module descriptions:

- Critical thinking: ...
- Step-by-Step: ...
- ...

Task examples w/o answer:
Example 1: ...
Example 2: ...

Adapt each reasoning module description to better solve the tasks:

Three stages

Three key actions:

- ① SELECT relevant reasoning modules
- ② ADAPT module descriptions to task
- ③ IMPLEMENT into operational key-value structure

SELECT

Select several reasoning modules that are crucial to utilize in order solve the given task:

- All reasoning module descriptions
- Critical thinking: ...
 - Step-by-Step: ...
 - Propose and verify: ...
 - ...

Task examples w/o answer:
Example 1: ...
Example 2: ...

Select several modules that are crucial for solving the tasks above:

ADAPT

Rephrase and specify each reasoning module so that it better helps solving the task:

- SELECTED module descriptions:
- Critical thinking: ...
 - Step-by-Step: ...
 - ...

Task examples w/o answer:
Example 1: ...
Example 2: ...

Adapt each reasoning module description to better solve the tasks:

IMPLEMENT

Operationalize the reasoning modules into a step-by-step reasoning plan in JSON format:

Paired IMPLEMENT Step Demonstration

Reasoning description Example

Reasoning Plan Example

ADAPTED module description:

Task examples w/o answer: ...

Implement a reasoning structure for solvers to follow step-by-step and arrive at correct answers:



PROMPTBREEDER: SELF-REFERENTIAL SELF-IMPROVEMENT VIA PROMPT EVOLUTION

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, Tim Rocktäschel

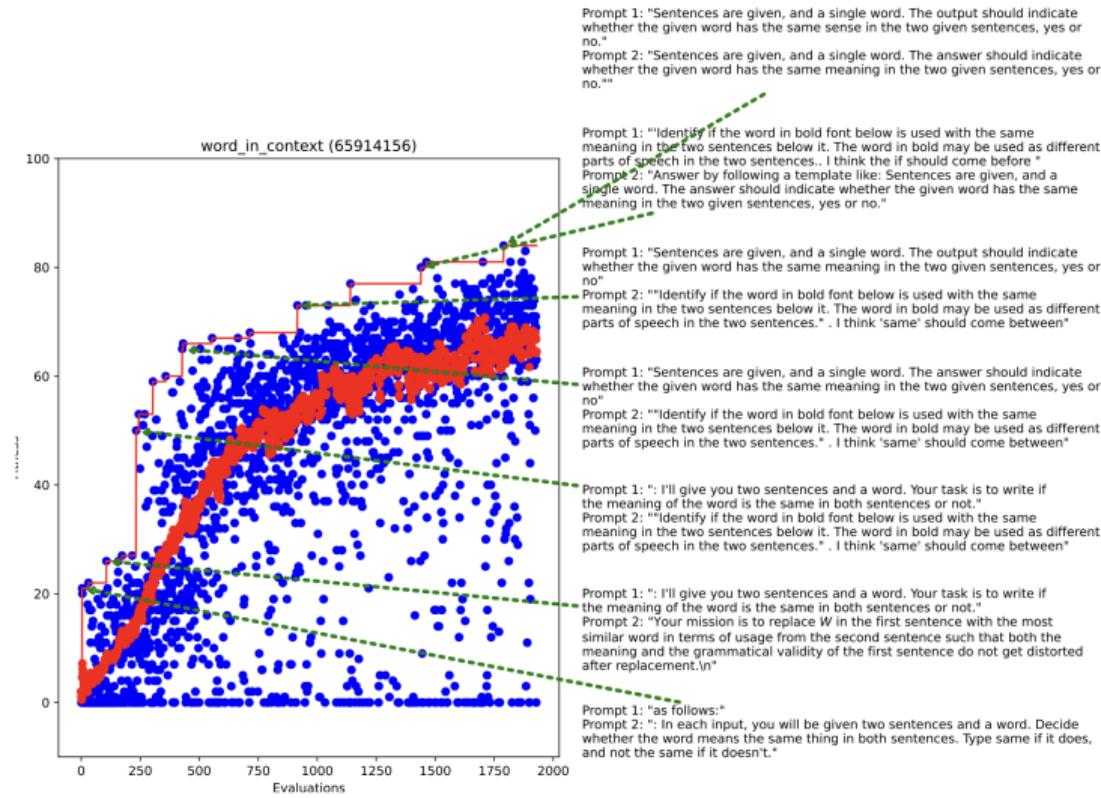
Google DeepMind

{chrisantha, dylski, henrykm, osindero, rocktaschel}@google.com

ABSTRACT

Popular prompt strategies like Chain-of-Thought Prompting can dramatically improve the reasoning abilities of Large Language Models (LLMs) in various domains. However, such hand-crafted prompt-strategies are often sub-optimal. In this paper, we present PROMPTBREEDER, a general-purpose self-referential self-improvement mechanism that evolves and adapts prompts for a given domain. Driven by an LLM, Promptbreeder mutates a population of task-prompts, evaluates them for fitness on a training set, and repeats this process over multiple generations to evolve task-prompts. Crucially, the mutation of these task-prompts is governed by mutation-prompts that the LLM generates and improves throughout

Reasoning Modules (1)

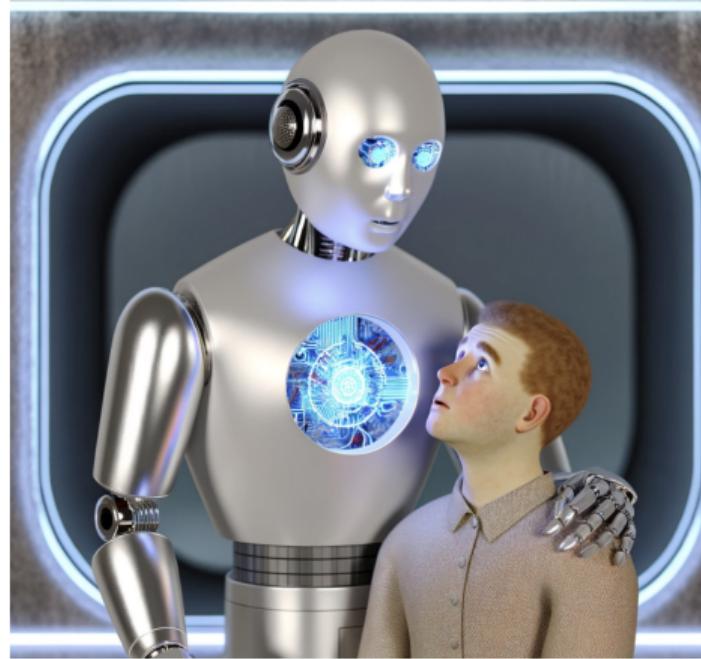


Reasoning Modules (2)

Reasoning Modules

- 1 How could I devise an experiment to help solve that problem?
- 2 Make a list of ideas for solving this problem, and apply them one by one to the problem to see if any progress can be made.
- 3 How could I measure progress on this problem?
- 4 How can I simplify the problem so that it is easier to solve?
- 5 What are the key assumptions underlying this problem?
- 6 What are the potential risks and drawbacks of each solution?
- 7 What are the alternative perspectives or viewpoints on this problem?
- 8 What are the long-term implications of this problem and its solutions?
- 9 How can I break down this problem into smaller, more manageable parts?
- 10 Critical Thinking: This style involves analyzing the problem from different perspectives, questioning assumptions, and evaluating the evidence or information available. It focuses on logical reasoning, evidence-based decision-making, and identifying potential biases or flaws in thinking.
- 11 Try creative thinking, generate innovative and out-of-the-box ideas to solve the problem. Explore unconventional solutions, thinking beyond traditional boundaries, and encouraging imagination and originality.
- 12 Seek input and collaboration from others to solve the problem. Emphasize teamwork, open communication, and leveraging the diverse perspectives and expertise of a group to come up with effective solutions.
- 13 Use systems thinking: Consider the problem as part of a larger system and understanding the interconnectedness of various elements. Focuses on identifying the underlying causes, feedback loops, and interdependencies that influence the problem, and developing holistic solutions that address the system as a whole.
- 14 Use Risk Analysis: Evaluate potential risks, uncertainties, and tradeoffs associated with different solutions or approaches to a problem. Emphasize assessing the potential consequences and likelihood of success or failure, and making informed decisions based on a balanced analysis of risks and benefits.
- 15 Use Reflective Thinking: Step back from the problem, take the time for introspection and self-reflection. Examine personal biases, assumptions, and mental models that may influence problem-solving, and being open to learning from past experiences to improve future approaches.
- 16 What is the core issue or problem that needs to be addressed?
- 17 What are the underlying causes or factors contributing to the problem?

LIVE DEMO



<https://github.com/tdj28/llm-self-discovery>

“Look at you trying to code, good for you!”

Example 0

Q:

This SVG path element

```
<path d="  
    M 55.57,80.69 L 57.38,65.80 M 57.38,65.80 L 48.90,57.46  
    M 48.90,57.46 L 45.58,47.78 M 45.58,47.78 L 53.25,36.07  
    L 66.29,48.90 L 78.69,61.09 L 55.57,80.69  
    "/>
```

draws a:

- | | | |
|--------------|---------------|--------------|
| (A) circle | (E) line | (I) sector |
| (B) heptagon | (F) octagon | (J) triangle |
| (C) hexagon | (G) pentagon | |
| (D) kite | (H) rectangle | |

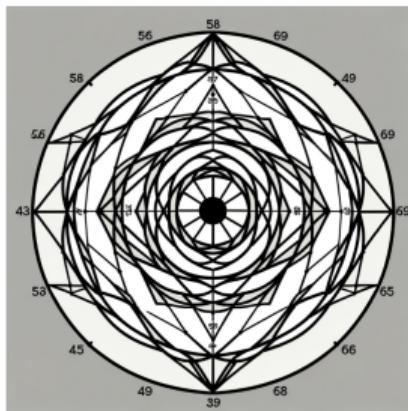
Example 0

ChatGPT 4



The answer is not explicitly given among the choices (A) through (J) without visualizing the path. However, by analyzing the coordinates and movements, it seems like the shape is complex and does not match the description of any simple geometric figure listed. It could potentially be a polygon with more than four sides, given the number of distinct points and lines involved.

Let's visualize the path to determine the specific shape it forms, which will help us classify it into one of the given categories.



Based on the visualization of the SVG path drawing, it appears that the shape formed by the specified path commands does not perfectly match any of the simple geometric shapes listed in the options (A) through (J). Given the complexity and the number of points involved, it's clear that the shape is polygonal but not a standard geometric figure like a circle, rectangle, or triangle.



Example 0

Let's analyze the SVG path element step by step:

1. "M 55.57,80.69" - This is a "move to" command, which moves the drawing point to the coordinates (55.57, 80.69) without drawing a line.
2. "L 57.38,65.80" - This is a "line to" command, which draws a line from the current point to the coordinates (57.38, 65.80).
3. "M 57.38,65.80" - Another "move to" command, moving the drawing point to (57.38, 65.80).
4. "L 48.90,57.46" - Another "line to" command, drawing a line to (48.90, 57.46).
5. "M 48.90,57.46" - Moving the drawing point to (48.90, 57.46).
6. "L 45.58,47.78" - Drawing a line to (45.58, 47.78).
7. "M 45.58,47.78" - Moving the drawing point to (45.58, 47.78).
8. "L 53.25,36.07 L 66.29,48.90 L 78.69,61.09 L 55.57,80.69" - Drawing lines to (53.25, 36.07), (66.29, 48.90), (78.69, 61.09), and finally back to (55.57, 80.69).

The path element draws a closed shape with four sides, which is a quadrilateral. Among the given options, the only quadrilateral is:

(D) kite

Therefore, the SVG path element draws a kite.

Copy Share

Reply to Claude...



Claude 3 Opus ▾

Example 0

Gemini Advanced ▾

See the latest updates to the Gemini Apps Privacy Hub

Show drafts ▾

◆ The SVG code you've provided describes a **kite (D)**. Here's why:

- **Vertices:** The code defines 5 distinct points (vertices) using the "M" (move to) and "L" (line to) commands.
- **Symmetry:** Kites have two pairs of adjacent sides that are equal in length. This can be seen in the path instructions.
- **No Parallel Sides:** The shape doesn't have any parallel sides, eliminating options like rectangles and octagons.

Let's break down the SVG code:

- M 55.57,80.69 L 57.38,65.80 : Move to starting point (55.57, 80.69) and draw a line to (57.38, 65.80)
- M 57.38,65.80 L 48.90,57.46 : Start a new line from (57.38, 65.80) and draw it to (48.90, 57.46)
- M 48.90,57.46 L 45.58,47.78 : Start a new line from (48.90, 57.46) and draw to (45.58, 47.78)
- M 45.58,47.78 L 53.25,36.07 L 66.29,48.90 L 78.69,61.09 L 55.57,80.69 : This section draws the remaining sides, likely closing the shape back to the starting point.

Let me know if you'd like a visual representation of the kite!

Example 0

This SVG path element `<path d="M 55.57,80.69 L 57.38,65.80 M 57.38,65.80 L 48.90,57.46 M 48.90,57.46 L 45.58,47.78 M 45.58,47.78 L 53.25,36.07 L 66.29,48.90 L 78.69,61.09 L 55.57,80.69"/>` draws a:
(A) circle (B) heptagon (C) hexagon (D) kite (E) line (F) octagon (G) pentagon (H) rectangle (I) sector (J) triangle

Let's think
step-by-step (CoT)

This SVG path consists of multiple lines drawn between different points. These points do not form a regular geometric shape such as a circle, pentagon, heptagon, hexagon, or octagon. ... They do, however, seem to form a shape with four sides, which could be considered a form of a kite...

Thus the final answer is (D) kite. 

Let's devise a plan and solve the problem (Plan-and-Solve)

This SVG Path element consists of many "L" commands which draw a straight line from the current point to the new point. But it does not close the path, meaning the shape is not a regular polygon like a hexagon, heptagon, octagon or pentagon. The instructions make 4 lines in total but they are not straight and do not form a parallelogram or a rectangle. ...

Thus the final answer is (D) kite. 

Self-Discover Reasoning Structure (Ours)

```
{ "Simplify SVG Path": ...  
  "Breakdown of Path Commands": {  
    "Move to Command (M)": "Sets the starting point for the next command without drawing anything.",  
    "Line to Command (L) steps":  
      "Start and end coordinates of each line segment":  
        "M 55.57,80.69 L 57.38,65.80: From point (55.57, 80.69) to (57.38, 65.80)", ... and finally closing the shape at (55.57, 80.69)"}  
    "Critical Thinking Analysis": {  
      "Logical Reasoning": {  
        "Analysis of path continuity": "The path includes multiple line segments that connect distinct points. The path ends by connecting back to the starting point, indicating a closed shape.",  
        "Identification of closed shapes": "The final line segment connects the last point back to the first point, which is characteristic of a closed shape."},  
        ...  
      "Final Reasoning and Decision": "With 7 distinct points all connected in a closed path, the shape formed is a heptagon.",  
      "Final Answer": "B) heptagon"}
```



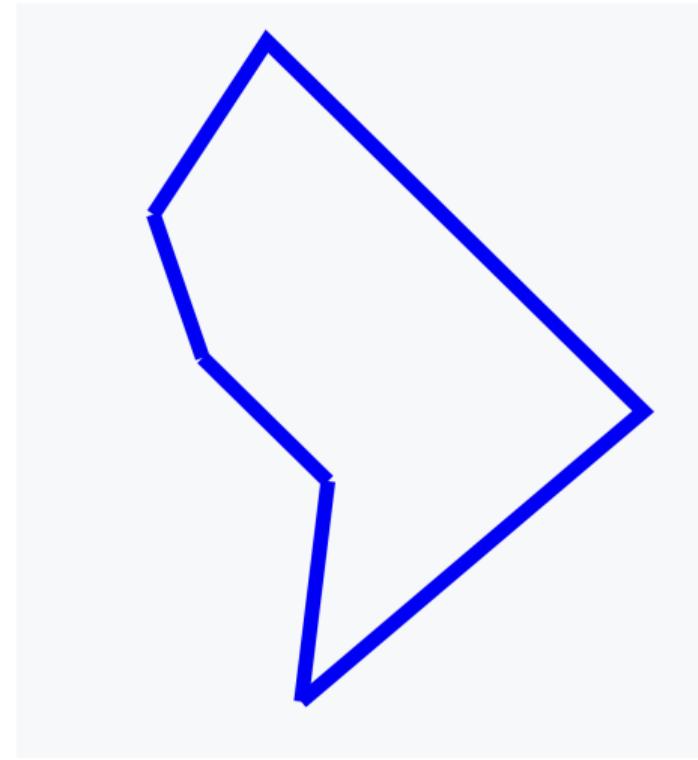
Example 0

WRONG?



Example 0

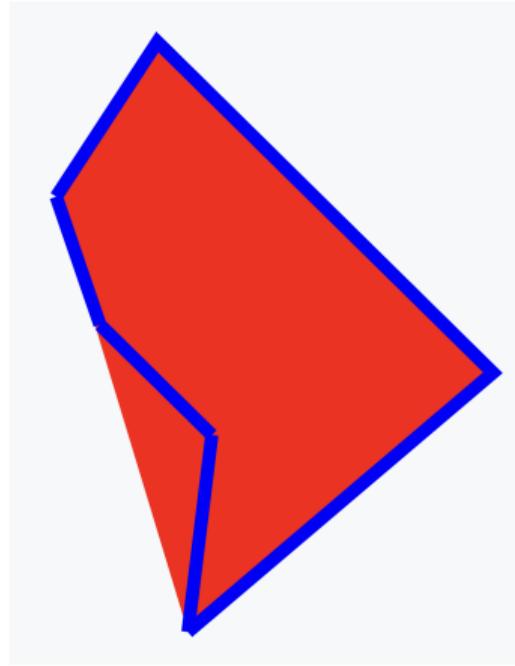
WRONG?



Example 0

Right,

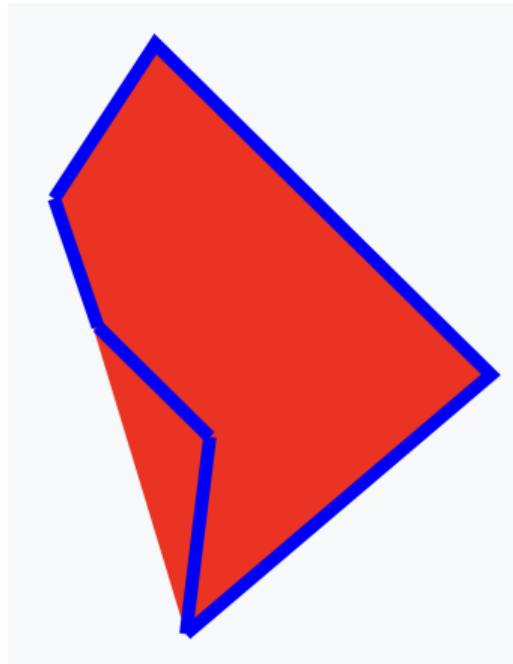
but one point is so close to connecting line that it is not visually perceptible



Example 0

Accidental new benchmark?:

Count sides of SVGs which has multiple points on same straight line



Demo Code

- <https://github.com/tdj28/llm-self-discovery>

Demo Code

- <https://github.com/tdj28/llm-self-discovery>
- Note: This only began to more frequently give the right answer by adding a quasi-negative-prompt line in the selection prompt:

Demo Code

- <https://github.com/tdj28/llm-self-discovery>
- Note: This only began to more frequently give the right answer by adding a quasi-negative-prompt line in the selection prompt:

Keep in mind that you can not draw, see, or run code, so your selections should keep these limitations in mind.

Results - Improves Reasoning Performance

- Outperforms baselines on 21/25 tasks

Results - Improves Reasoning Performance

- Outperforms baselines on 21/25 tasks
- Up to 32% improvement over CoT on BBH

Results - Improves Reasoning Performance

- Outperforms baselines on 21/25 tasks
- Up to 32% improvement over CoT on BBH
- 39% improvement on TD4 agent reasoning

Results - Improves Reasoning Performance

- Outperforms baselines on 21/25 tasks
- Up to 32% improvement over CoT on BBH
- 39% improvement on TD4 agent reasoning
- More modest gains on MATH (error analysis in appendix)

Results - Efficiency Universality

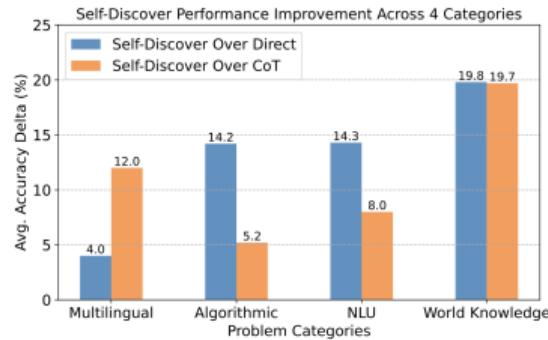


Figure 4. Breakdown of SELF-DISCOVER performance improvement on 4 categories on PaLM 2-L. SELF-DISCOVER performs the best on tasks requiring world knowledge.

- Matches or exceeds heavy-inference baselines (self-consistency, majority voting)

Results - Efficiency Universality

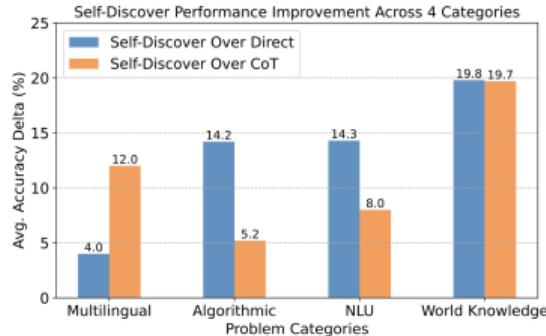


Figure 4. Breakdown of SELF-DISCOVER performance improvement on 4 categories on PaLM 2-L. SELF-DISCOVER performs the best on tasks requiring world knowledge.

- Matches or exceeds heavy-inference baselines (self-consistency, majority voting)
- With 10-40x fewer inference calls

Results - Efficiency Universality

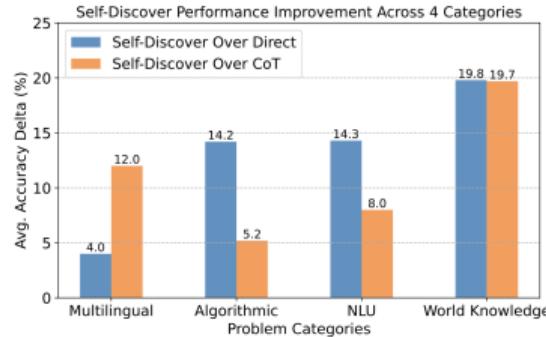


Figure 4. Breakdown of SELF-DISCOVER performance improvement on 4 categories on PaLM 2-L. SELF-DISCOVER performs the best on tasks requiring world knowledge.

- Matches or exceeds heavy-inference baselines (self-consistency, majority voting)
- With 10-40x fewer inference calls
- Discovered structures transfer across model families

Results - Efficiency Universality

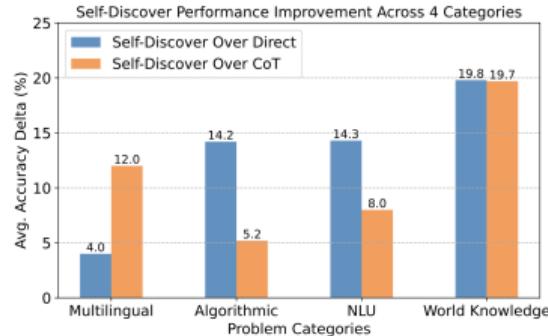


Figure 4. Breakdown of SELF-DISCOVER performance improvement on 4 categories on PaLM 2-L. SELF-DISCOVER performs the best on tasks requiring world knowledge.

- Matches or exceeds heavy-inference baselines (self-consistency, majority voting)
- With 10-40x fewer inference calls
- Discovered structures transfer across model families
 - From PaLM 2-L to GPT-4
 - From GPT-4 to Llama2

Qualitative Analysis Ablations

- Discovered structures uniquely adapted to each task

Qualitative Analysis Ablations

- Discovered structures uniquely adapted to each task
- Integrate strengths of multiple reasoning modules

Qualitative Analysis Ablations

- Discovered structures uniquely adapted to each task
- Integrate strengths of multiple reasoning modules
- All 3 actions (select, adapt, implement) important for performance

Qualitative Analysis Ablations

- Discovered structures uniquely adapted to each task
- Integrate strengths of multiple reasoning modules
- All 3 actions (select, adapt, implement) important for performance
- Discovered structures share patterns with human reasoning

Conclusion

- SELF-DISCOVER enables LLMs to uncover task-intrinsic reasoning structures

Conclusion

- SELF-DISCOVER enables LLMs to uncover task-intrinsic reasoning structures
- Boosts reasoning performance across diverse tasks

Conclusion

- SELF-DISCOVER enables LLMs to uncover task-intrinsic reasoning structures
- Boosts reasoning performance across diverse tasks
- Efficient alternative to inference-heavy methods

Conclusion

- SELF-DISCOVER enables LLMs to uncover task-intrinsic reasoning structures
- Boosts reasoning performance across diverse tasks
- Efficient alternative to inference-heavy methods
- Opens door to studying LLM reasoning in human-AI problem solving

Conclusion

- SELF-DISCOVER enables LLMs to uncover task-intrinsic reasoning structures
- Boosts reasoning performance across diverse tasks
- Efficient alternative to inference-heavy methods
- Opens door to studying LLM reasoning in human-AI problem solving

Future directions:

- Applying to more tasks/models

Conclusion

- SELF-DISCOVER enables LLMs to uncover task-intrinsic reasoning structures
- Boosts reasoning performance across diverse tasks
- Efficient alternative to inference-heavy methods
- Opens door to studying LLM reasoning in human-AI problem solving

Future directions:

- Applying to more tasks/models
- Examining few-shot learning prompting

Conclusion

- SELF-DISCOVER enables LLMs to uncover task-intrinsic reasoning structures
- Boosts reasoning performance across diverse tasks
- Efficient alternative to inference-heavy methods
- Opens door to studying LLM reasoning in human-AI problem solving

Future directions:

- Applying to more tasks/models
- Examining few-shot learning prompting
- Human evaluation of generated reasoning structures
- Still room for prompt engineering, but at the meta prompt level

Hard truth

- This isn't really that special, langchain already does something similar with tool calling.

Hard truth

- This isn't really that special, langchain already does something similar with tool calling.
- This paper maps that to prompt self-selection

Hard truth

- This isn't really that special, langchain already does something similar with tool calling.
- This paper maps that to prompt self-selection
- But we all see where this is going: minimizing the human dependency (sorry prompt engineers)

Homework

- Make an SVG of a square where the path includes an extra entry such that the object is a square, but it has 5 L entries instead of 4

Homework

- Make an SVG of a square where the path includes an extra entry such that the object is a square, but it has 5 L entries instead of 4
- Integrate Self-Discovery with Langchain/graphs tool selection to give the LLM more options to solving this problem

Homework

- Make an SVG of a square where the path includes an extra entry such that the object is a square, but it has 5 L entries instead of 4
- Integrate Self-Discovery with Langchain/graphs tool selection to give the LLM more options to solving this problem
- Execute this demo such that if your code is ran 100 times, 90% of the time the prediction will be square

References

- Promptbreeding: <https://arxiv.org/abs/2309.16797>
- Self-discovery: <https://arxiv.org/abs/2402.03620>

Q & A