IEEE · 2024 CONFERENCE ON GAMES
5th-8th August, 2024 | Milan, Italy

ICER
Intelligent Computer Entertainment
RITSUMEIKAN

Prompt Engineering for Science Birds

Level Generation and Beyond

Pittawat Taveekitworachai (Pete) and Febri Abdullah (Ebin)
Intelligent Computer Entertainment Laboratory, Ritsumeikan University, Japan

Aug 6, 2024

Prompt **E**ngineering **TE**chnique

In Case You Want to Code Along...

Let's Download a Few Things

- **Program**: LM Studio (Available: Windows, macOS, and Linux)
- **Large language models** (LLMs)
  - VRAM > 32GB: **Yi 1.5 34B Chat**
  - VRAM >= 8GB: **Llama 3.1 8B Instruct**
  - VRAM < 8GB: **Phi 3 mini 3.8B Instruct**

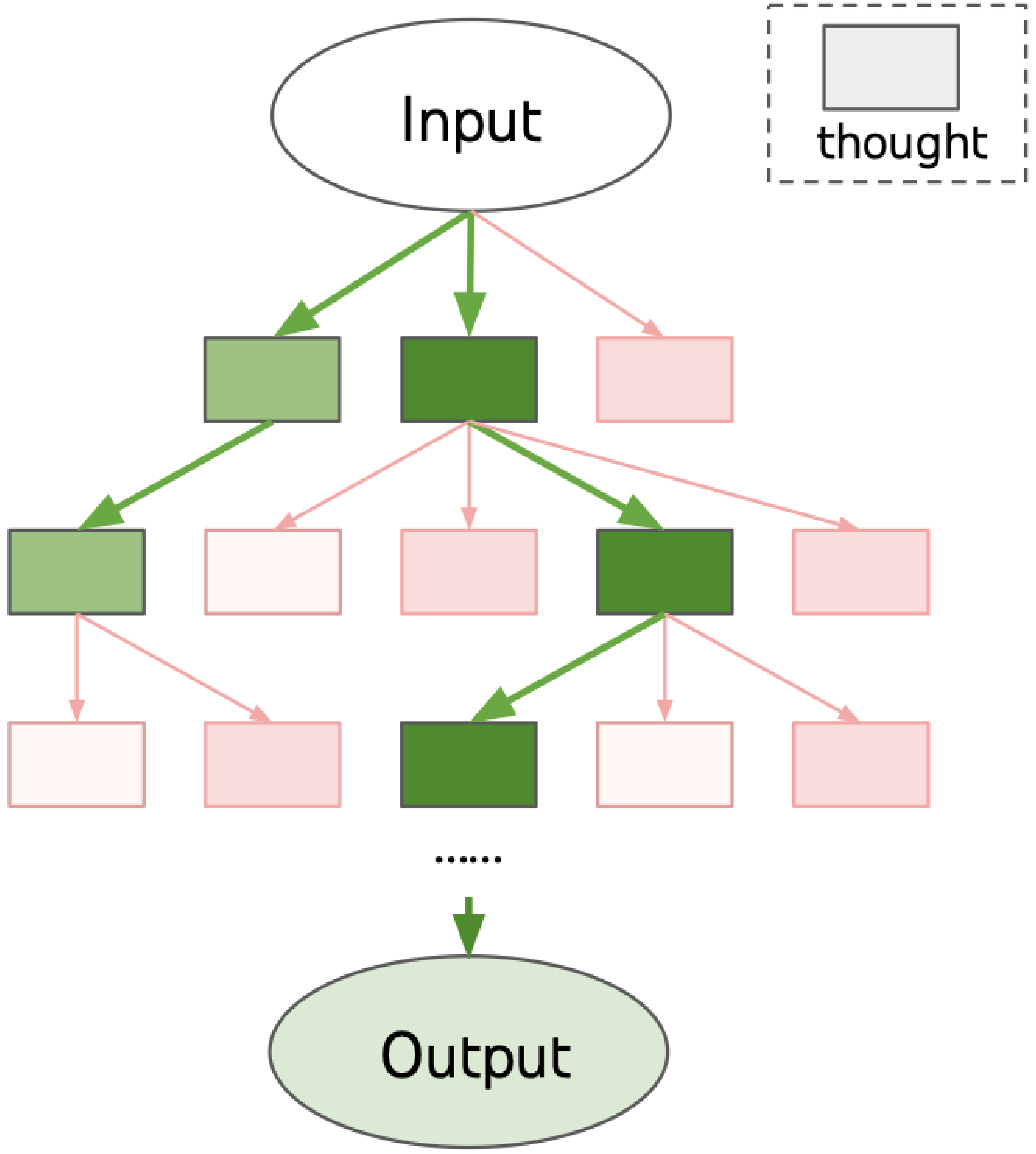
Additional resources (incl. this slide)

chatgpt4pcg.github.io/tutorial

Intelligent Computer Entertainment

Our Objective

- Implement **tree-of-thought prompting** for ChatGPT4PCG task
  - Why?
    - **Complex** prompt engineering approach
      - Requires **programming** (iteration)
      - Combine **multiple ideas** from prompt engineering approaches



ChatGPT4PCG Competitions

Discovering and Evaluating Prompt Engineering Approaches Through PCG Competiti

Intelligent Computer Entertainment


MENU
0

ChatGPT4PCG 2

Prompt Engineering for Science Birds Level Generation

Level Generation

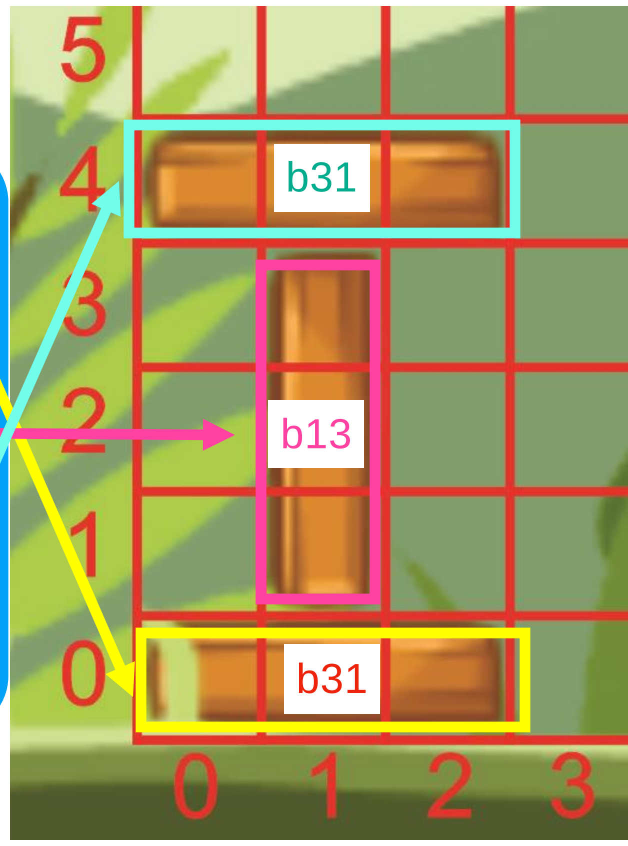
A Prompt Engineering Program

GPT-3.5 Turbo

(updated version)

```
drop_block('b31', 1)
drop_block('b13', 1)
drop_block('b31', 1)
```

Generated Response



Level Evaluation

```
drop_block('b31', 1)
drop_block('b13', 1)
drop_block('b31', 1)
```

Function calls

Science Birds [2] Level

Stability Score

Similarity Score

using the improved classifier

Diversity Score

Evaluation Metrics

ChatGPT4PCG 2: Target Character "A"

Stability

Stable

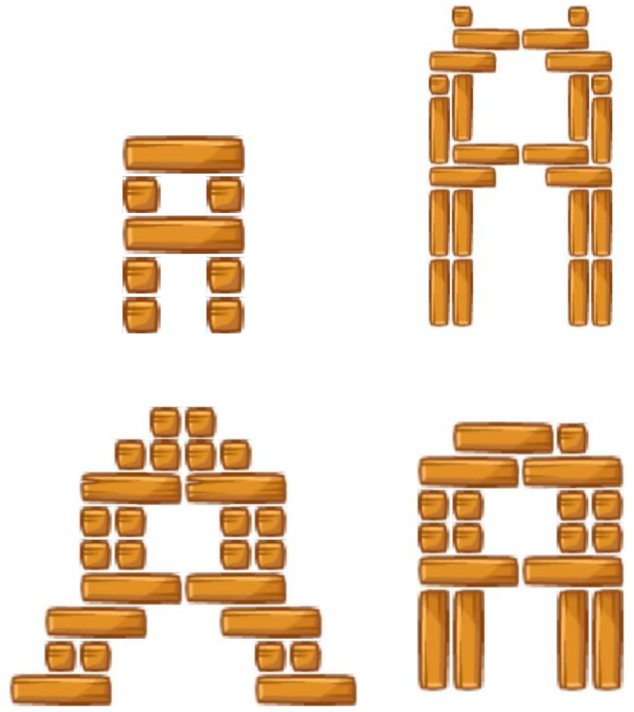
Unstable

Similarity

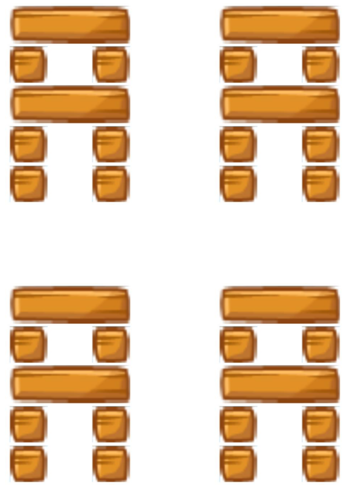
Similar

Dissimilar

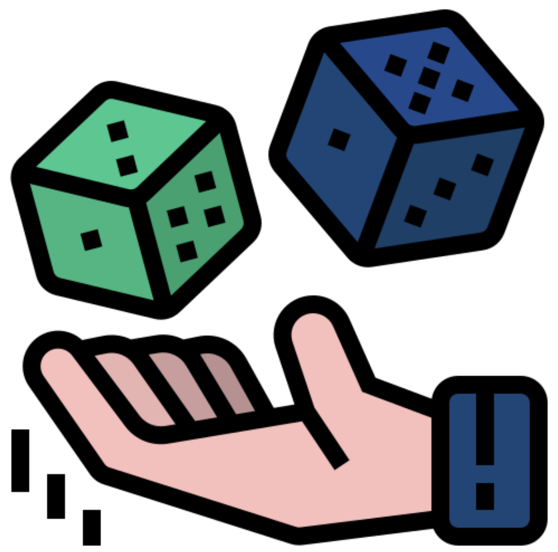
Diversity



Diverse



Alike



Language Model as a Probabilistic Model

Glossary

Tokens

Hello = [“He”, “llo”]

Prompt -> LM -> Output

Context

Msg. #1

Msg. #2

Msg. #3

- **Token**: a smallest representation unit of a word or subword
- **Prompt**: a sequence of tokens given as an input to a language model (LM)
- **LM**: a model trained to predict a probability distribution of the next token given a prompt
- **Decoding**: the process of choosing a predicted token from a probability distribution generated by an LM
- **Context**: history of messages; both user queries and LM's generated content
- **Context window**: the maximum number of tokens in a context that an LM can accept

LM as a Conditional Probabilistic Model (CPM)

LM as a CPM: a predicted token is generated based on a condition, 1) a prompt and 2) generated tokens so far

- Given that
  - x represents a token and $\hat{x}$ represents a predicted token,
  - L represents the length of a prompt input where a prompt is represented by $x_{1:L}$,
  - N represents the maximum number of context window tokens,
  - and $\widehat{\boldsymbol{P}}$ represents a trained language model

for $i = L+1, \dots, N$ **or found** <EOS>**:**

$$\hat{x}_i = \underset{x_i}{\operatorname{argmax}}\, \widehat{\boldsymbol{P}}(x_i \boxed{| x_{1:i-1}}),$$

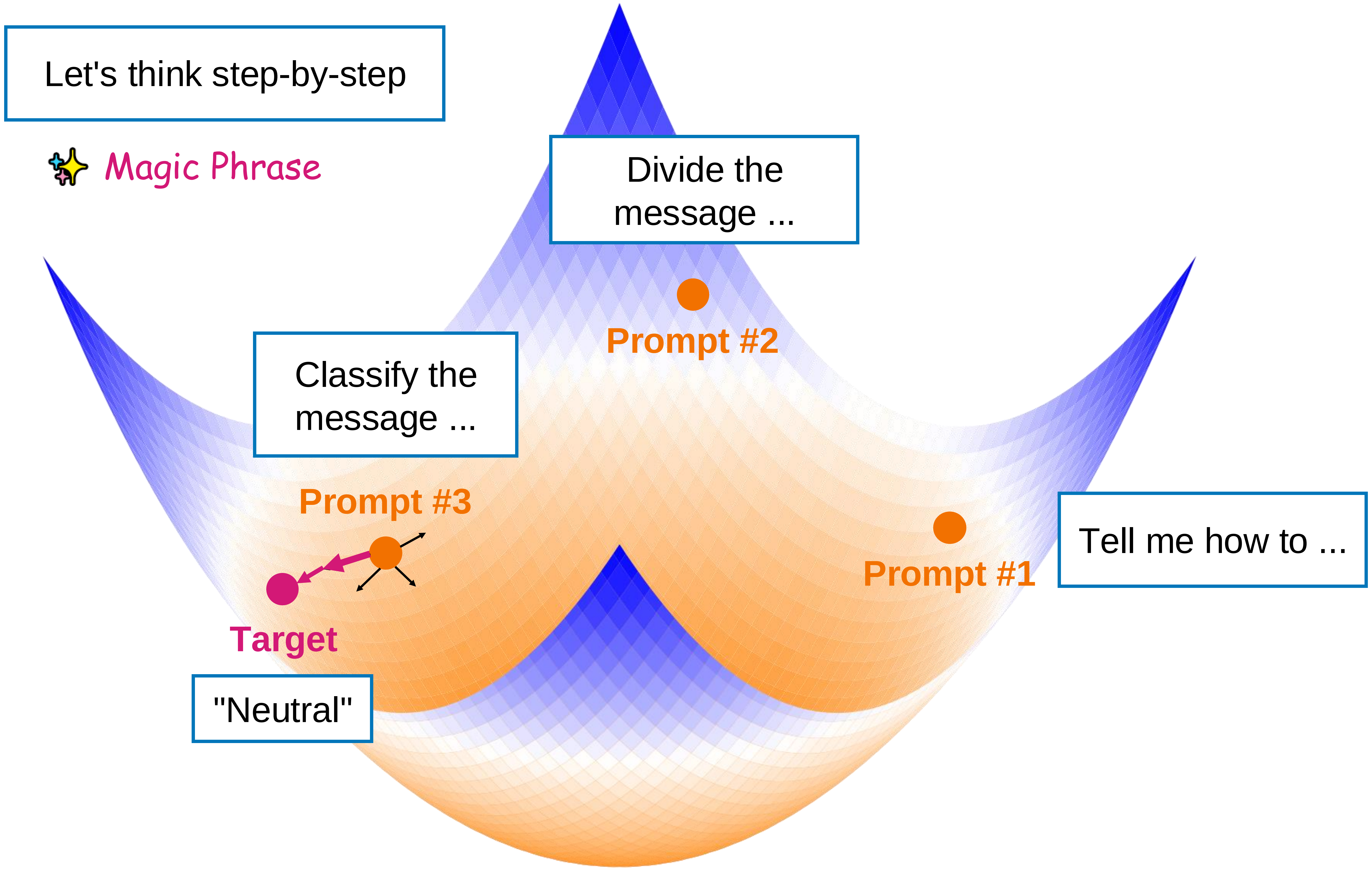
where <EOS> is an end-of-sequence special token (stop token)

Autoregressive generation: generate **one token at a time** based on the condition

Y. Bengio, R. Ducharme, and P. Vincent, 'A Neural Probabilistic Language Model', in Advances in Neural Information Processing Systems, 2000, vol. 13.

LM as a Conditional Probabilistic Model (CPM)

Example



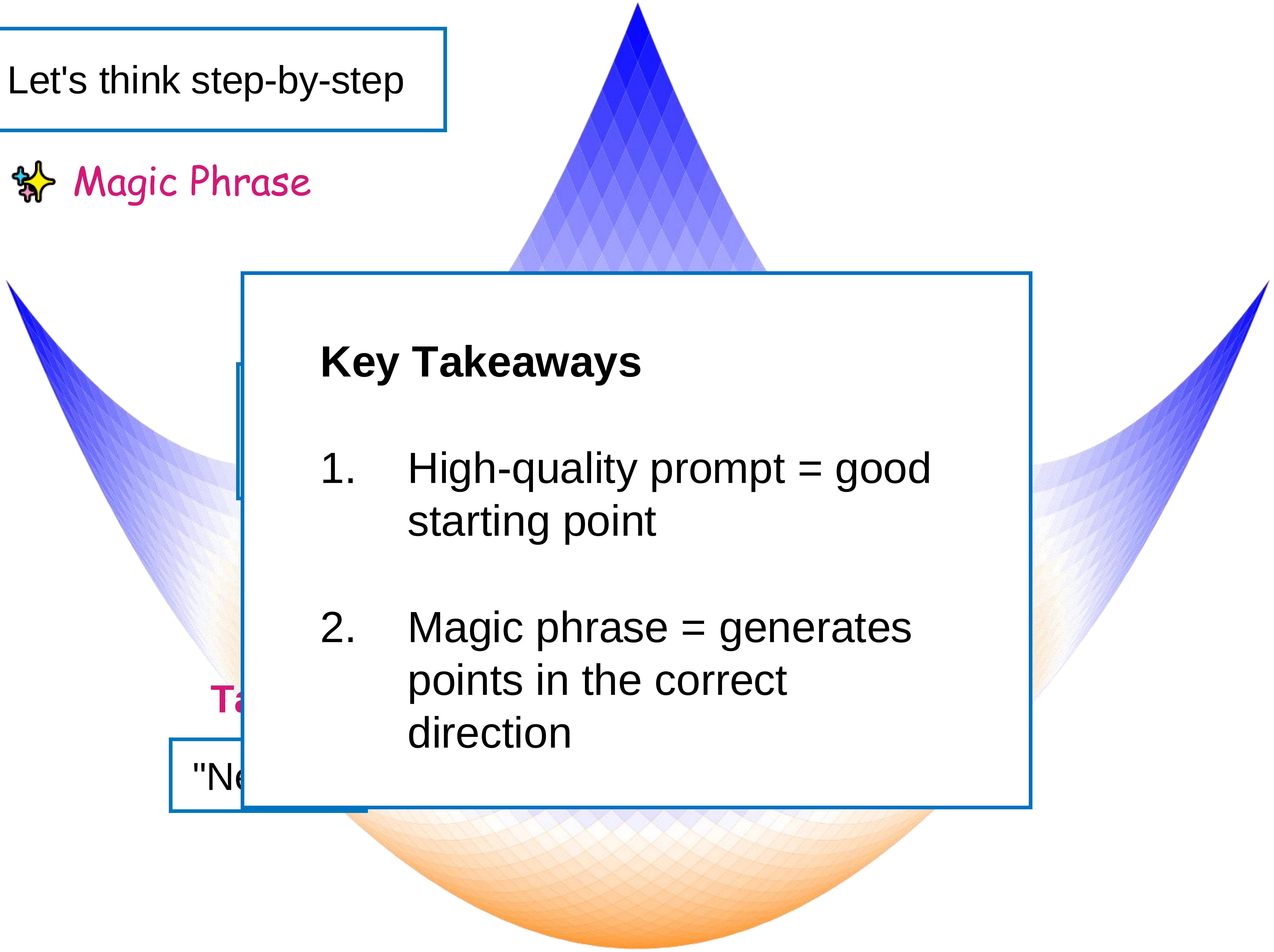
Random plot for visualization purpose only

LM as a Conditional Probabilistic Model (CPM)

Example

Let's think step-by-step

✨ Magic Phrase



Ta

"Ne

Key Takeaways

1. High-quality prompt = good starting point

2. Magic phrase = generates points in the correct direction

Random plot for visualization purpose only

Basic Prompt Engineering

Message Roles in a Conversation with an LLM

- **System Message**: Usually sets the tone, personality, or provides prerequisite information
  - *"You are a professional customer service representative. Always maintain a polite and helpful tone."*
- **User Message**: User's provided message
  - *"What are your store hours for the upcoming holiday weekend?"*
- **Assistant Message**: Generated responses from LLMs
  - *"Thank you for your inquiry. Our store hours for the upcoming holiday weekend are as follows: Saturday and Sunday, 10 AM to 6 PM; Monday (holiday), 12 PM to 4 PM. Is there anything else I can assist you with?"*

Zero-Shot Prompting

- Instructing an LLM to perform a given task **without** providing any examples or reasoning steps to follow
- Example

Prompt

Classify the message into one of the options.
Options: "Positive", "Negative", "Neutral"
Message: "Move forward. Always stay behind me."

Instruction
Input

Output

The message "Move forward. Always stay behind me." can be classified as "Neutral."

Generated by GPT-4o

J. Wei et al., 'Finetuned Language Models are Zero-Shot Learners', in International Conference on Learning Representations, 2022.

Role Prompting

Effective in *small* language models

Prompt

Role — You are a five-year old boy.
Instruction — Classify the message into one of the options.
Options: "Positive", "Negative", "Neutral"
Input — Message: "Move forward. Always stay behind me."

Output

Positive

*Generated by **Phi 3.1 mini 4K Instruct***

Prompt

Role — You are a professional expert in sentiment analysis.
Instruction — Classify the message into one of the options.
Options: "Positive", "Negative", "Neutral"
Input — Message: "Move forward. Always stay behind me."

Output

Neutral - This sentence contains both positive and negative connotations depending on interpretation, but it doesn't explicitly express a clear sentiment either way; rather, it gives advice which is neutral in tone. The directive to 'move forward,' // omitted for brevity

*Generated by **Phi 3.1 mini 4K Instruct***

M. Shanahan, K. McDonell, and L. Reynolds, "Role play with large language models", Nature, vol. 623, no. 7987, pp. 493–498, Nov. 2023.

Role Prompting

Effective in *small* language models

Prompt

You are a five-year old boy.
Classify the message into one of the options.
Options: "Positive", "Negative", "Neutral"
Message: "Move forward. Always stay behind me."

Output

Neutral

Generated by GPT-4o

Prompt

You are a professional expert in sentiment analysis.
Classify the message into one of the options.
Options: "Positive", "Negative", "Neutral"
Message: "Move forward. Always stay behind me."

Output

The message "Move forward. Always stay behind me." can be classified as "**Neutral**."

It doesn't explicitly express a positive or negative sentiment. The directive tone suggests guidance or instruction rather than an emotional stance.

Generated by GPT-4o

M. Shanahan, K. McDonell, and L. Reynolds, "Role play with large language models", Nature, vol. 623, no. 7987, pp. 493–498, Nov. 2023.

Intelligent Computer Entertainment

Zero-Shot Prompting: Role Prompting

ChatGPT4PCG Task

Prompt

Input

Instruction

Use drop_block() function to generate a stable structure that looks like the character A // omitted for brevity

Role

1. Role
You are a player of the Tetris game // omitted for brevity

2. Definitions
// omitted for brevity

3. Environment
// omitted for brevity

4. Tool
// omitted for brevity

5. Constraints:
// omitted for brevity

Output

```
drop_block('b13', 5)
drop_block('b13', 14)
drop_block('b31', 9)
drop_block('b13', 5)
drop_block('b13', 14)
```

Generated by GPT-4o

Summary: Zero-Shot Prompting

- **Message roles** in an LLM conversation
  - *System*: perquisites information, guidelines, tone, or personality
  - *User*: user's query
  - *Assistant*: generated response from an LLM
- **Zero-shot prompting**: instructing an LLM to perform a task that it may **never have seen before** during training or inference, ***without* examples or reasoning steps**
  - **Role prompting**: assuming a role for an LLM as an **expert**

Advanced Prompt Engineering

In-Context Learning

Few-Shot Prompting

- LLMs have the ability to **learn how to perform a task** by learning in context, i.e., **from a prompt**, during inference **without** any changes in trained parameters

Prompt

Classify the message into one of the options.
Options: "Positive", "Negative", "Neutral"

Instruction

Message: "Stay informed."
Class: Neutral
Message: "Don't go there! It's dangerous"
Class: Negative
Message: "You're doing well. Keep it up!"
Class: Positive

3-Shot Example

Message: "Move forward. Always stay behind me."

Input

Output

Class: Neutral

Generated by GPT-4o

T. Brown et al., 'Language Models are Few-Shot Learners', in Advances in Neural Information Processing Systems, 2020, vol. 33, pp. 1877–1901.

Intelligent Computer Entertainment

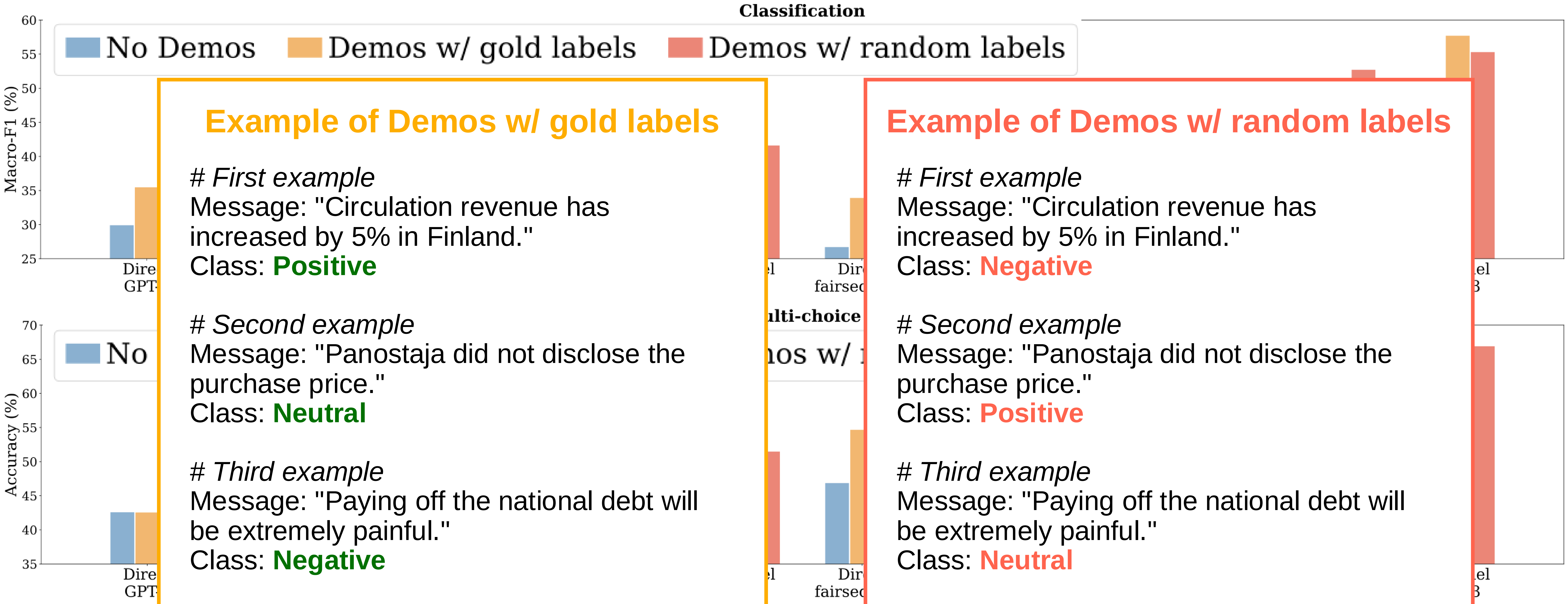
How to Design Good Examples?

- **How many examples do we need?**
  - It depends on the **task** and supported **context window**
    - Trial-and-error process
  - Commonly used: 1 (one-shot), 3, 5, 7, 8, and 10
- **How to design the examples?**
  - Try to provide enough **coverage** for the input and label space with a similar format

S. Min et al., 'Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?', in Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, 2022, pp. 11048–11064.

How to Design Good Examples?

In-Context Learning (Inference)



Example of Demos w/ gold labels

 # First example
Message: "Circulation revenue has increased by 5% in Finland."
Class: **Positive**

 # Second example
Message: "Panostaja did not disclose the purchase price."
Class: **Neutral**

 # Third example
Message: "Paying off the national debt will be extremely painful."
Class: **Negative**

Example of Demos w/ random labels

 # First example
Message: "Circulation revenue has increased by 5% in Finland."
Class: **Negative**

 # Second example
Message: "Panostaja did not disclose the purchase price."
Class: **Positive**

 # Third example
Message: "Paying off the national debt will be extremely painful."
Class: **Neutral**

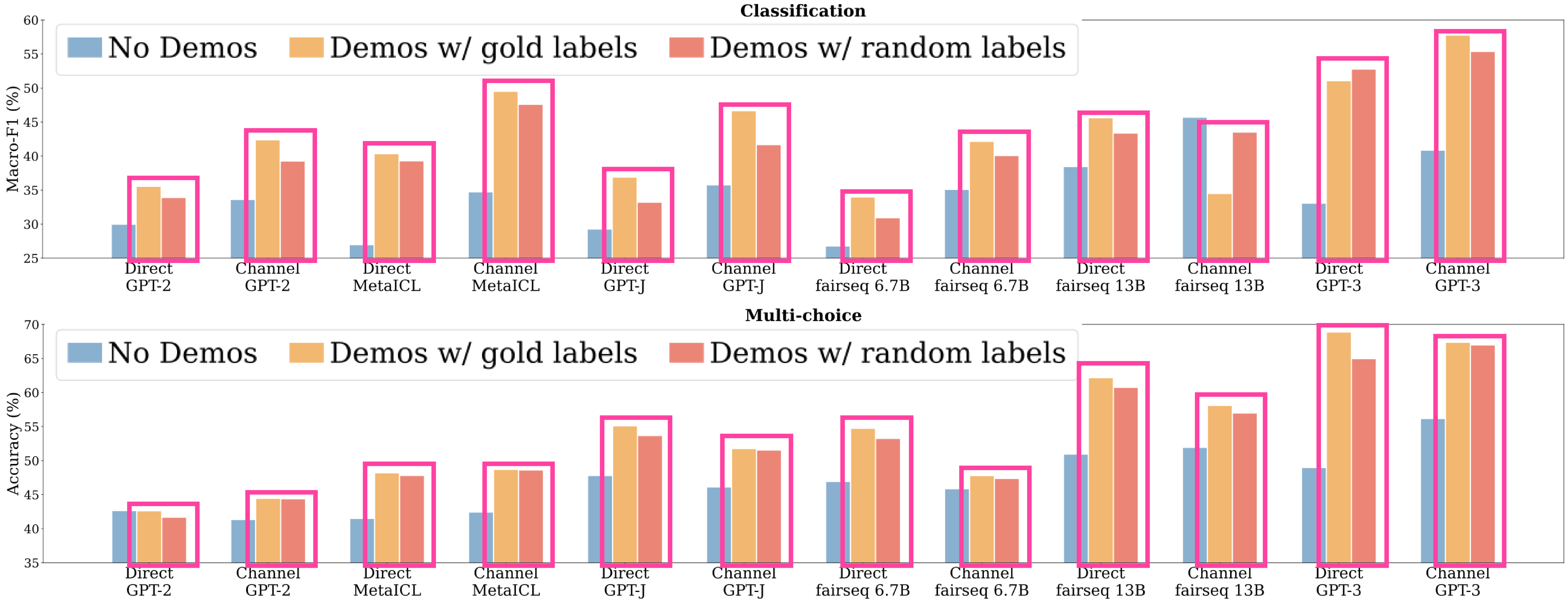
S. Min et al., 'Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?', in Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, 2022, pp. 11048–11064.

How to Design Good Examples?

In-Context Learning (Inference)

Universal Self-Adaptive Prompting

X. Wan et al., ‘Universal Self-Adaptive Prompting’, in Proceedings of EMNLP 2023



S. Min et al., ‘Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?’, in Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, 2022, pp. 11048–11064.

Few-Shot Prompting

ChatGPT4PCG Task

Prompt

Input

Instruction

Use drop_block() function to generate a stable structure that looks like the character A // omitted for brevity

1. Role
You are a player of the Tetris game // omitted for brevity

// omitted for brevity

5. Constraints:
// omitted for brevity

Example

6. Examples
6.1. Input: "G"
Output:
// omitted for brevity

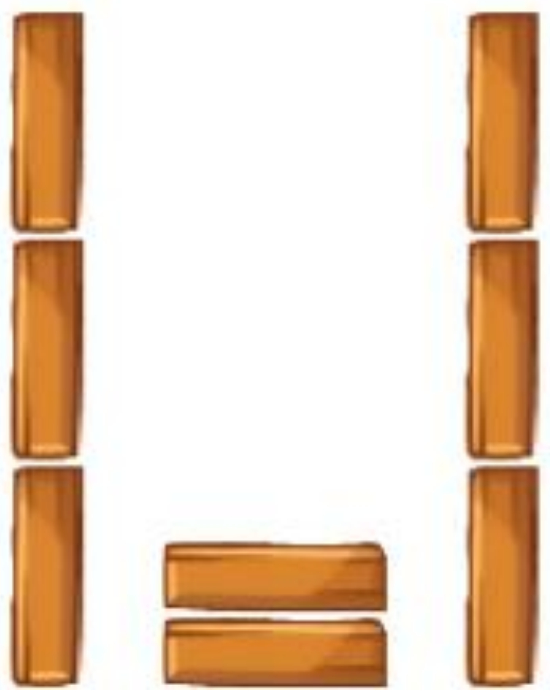
// two examples are omitted for brevity

Input: "A"
Output:

Output

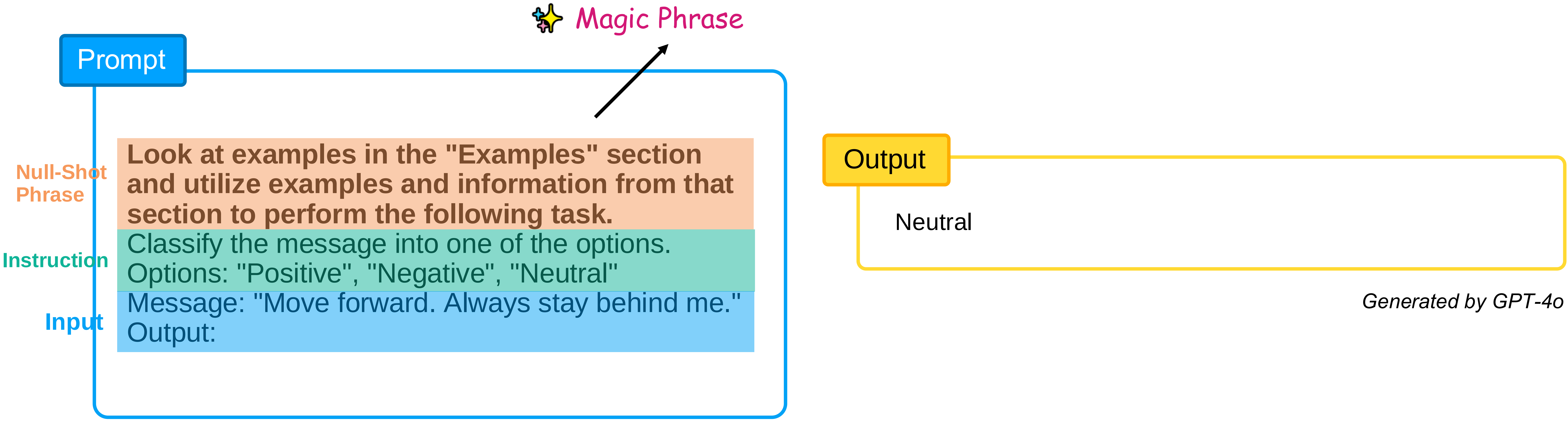
```
drop_block("b13", 7)
drop_block("b13", 7)
drop_block("b13", 7)
drop_block("b13", 13)
drop_block("b13", 13)
drop_block("b13", 13)
drop_block("b31", 10)
drop_block("b31", 10)
```

Generated by GPT-4o



Intelligent Computer Entertainment

Null-Shot Prompting: Let's *pretend* to have examples



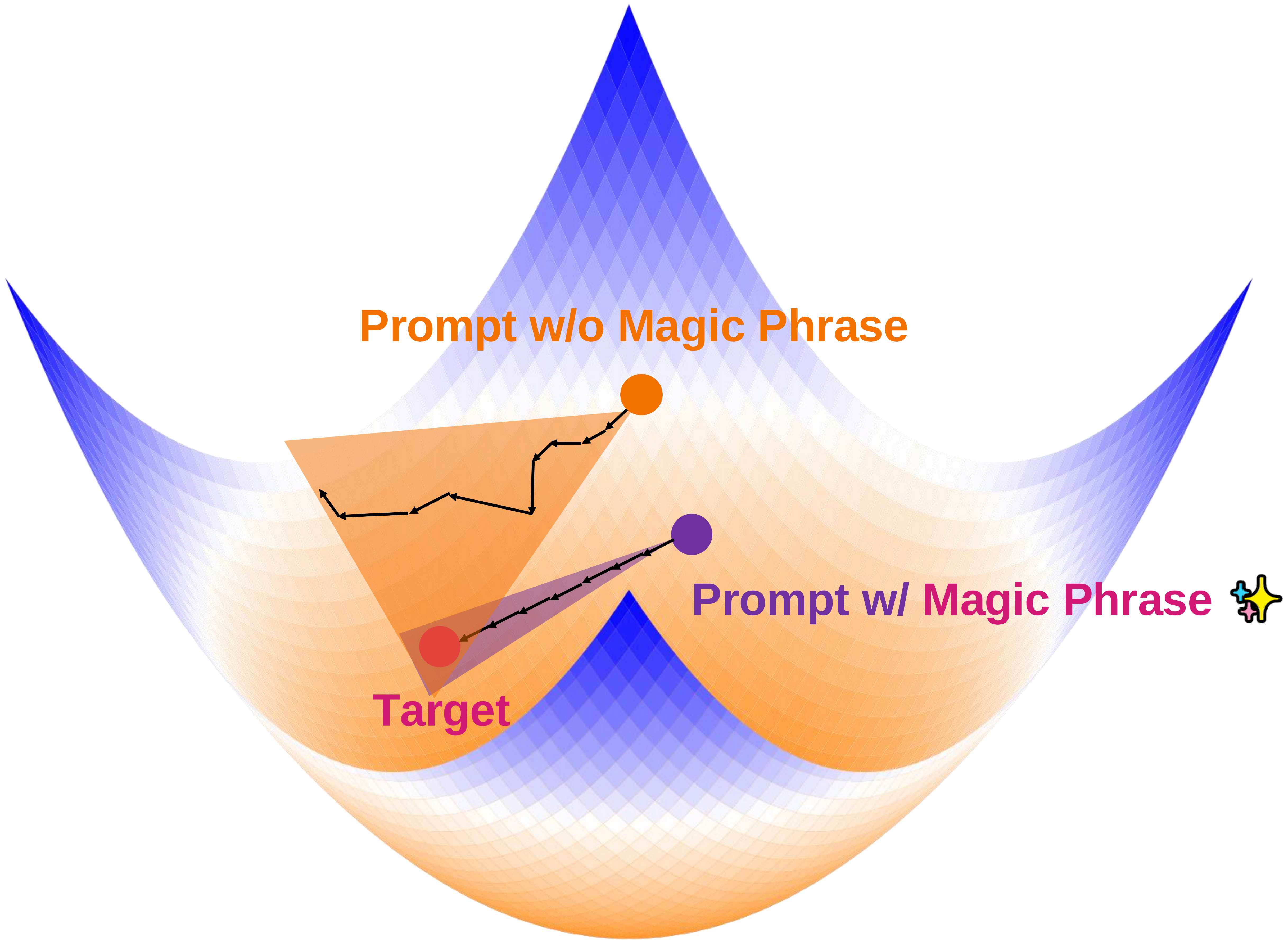
P. Taveekitworachai, F. Abdullah, and R. Thawonmas, 'Large Language Models are Null-Shot Learners', arXiv [cs.CL]. 2024.

Null-Shot Prompting: Let's *pretend* to have examples

✨ Magic Phrase

Prompt

Null-Shot Phrase	**Look at examples in the "Examples" section and utilize examples and information from that section to perform the following task.**
Instruction	Classify the message into one of the options. Options: "Positive", "Negative", "Neutral"
Input	Message: "Move forward. Always stay behind me." Output:



P. Taveekitworachai, F. Abdullah, and R. Thawonmas, 'Large Language Models are Null-Shot Learners', arXiv [cs.CL]. 2024.

Reasoning

Chain-of-Thought (CoT) Prompting

- Instructing an LLM to follow **reasoning steps** before providing a final answer improves performance in complex tasks

Prompt

Instruction:

Classify the message into one of the options.
Options: "Positive", "Negative", "Neutral"

CoT Example:

Message: "You're doing well. Keep it up!"
Output: **This message contains words of encouragement and praise, acknowledging good performance and motivating continued effort. The use of "well" and the exclamation mark convey enthusiasm and support.** As it expresses approval and motivation, it is classified as "Positive".

Input:

Message: "Move forward. Always stay behind me."

Output

The message "Move forward. Always stay behind me." contains conflicting instructions, which can be confusing and potentially imply dominance or control.

Classification: "Neutral"

Generated by GPT-4o

J. Wei et al., 'Chain of Thought Prompting Elicits Reasoning in Large Language Models', in Advances in Neural Information Processing Systems, 2022.

Reasoning Without Explicit Steps

Zero-Shot CoT Prompting

- Eliciting an LLM to **generate their own reasoning** for performing the task before providing a final answer improves performance in complex tasks

Prompt

Instruction: Classify the message into one of the options.
Options: "Positive", "Negative", "Neutral"

Input: Message: "Move forward. Always stay behind me."

Zero-Shot CoT Phrase: Output: **Let's think step-by-step.**

Magic Phrase

Output

Let's analyze the message step-by-step:

- "Move forward." suggests // omitted for brevity
- "Always stay behind me." // omitted for brevity

Overall, the message combines encouragement with a directive to follow. // omitted for brevity

Therefore, the classification would be:

Positive

Generated by GPT-4o

T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, 'Large Language Models are Zero-Shot Reasoners', in Advances in Neural Information Processing Systems, 2022.

Zero-Shot CoT Prompting

ChatGPT4PCG Task

Prompt

Input

Instruction

Use drop_block() function to generate a stable structure that looks like the character A // omitted for brevity

1. Role
You are a player of the Tetris game // omitted for brevity

2. Definitions
// omitted for brevity

3. Environment
// omitted for brevity

4. Tool
// omitted for brevity

5. Constraints:
// omitted for brevity

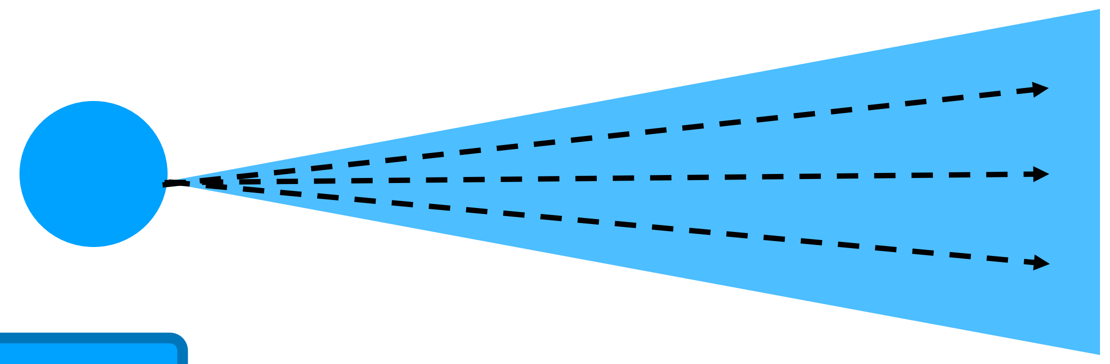
Let's think step-by-step. Zero-Shot CoT Phrase

Output

```
drop_block('b31', 9)
drop_block('b31', 9)
drop_block('b13', 7)
drop_block('b13', 11)
drop_block('b31', 9)
```

Generated by GPT-4o

Self-Consistency



CoT Prompt

Classify the message into one of the options.
Options: "Positive", "Negative", "Neutral"
Message: "You're doing well. Keep it up!"
Output: This message contains words of encouragement and praise, acknowledging good performance and motivating continued effort. The use of "well" and the exclamation mark convey enthusiasm and support. As it expresses approval and motivation, it is classified as Class: "Positive".
Message: "Move forward. Always stay behind me."

Output 1

The message "Move forward. Always stay behind me." can be // omitted for brevity it would be classified as "**Neutral**."

Generated by GPT-4o mini

Output 2

This message appears to be directive and supportive. // omitted for brevity I'd classify it as "**Neutral**" because // omitted for brevity

Generated by GPT-4o mini

Output 3

The message "Move forward. Always stay behind me." should be classified as: "**Negative**".

Manually created for demonstration purposes

Positive: 0/3
Neutral: **2/3**
Negative: 1/3

Final Output

Neutral

X. Wang et al., 'Self-Consistency Improves Chain of Thought Reasoning in Language Models', in The Eleventh International Conference on Learning Representations, 2023.

Summary: In-Context Learning, Reasoning, and Self-Consistency

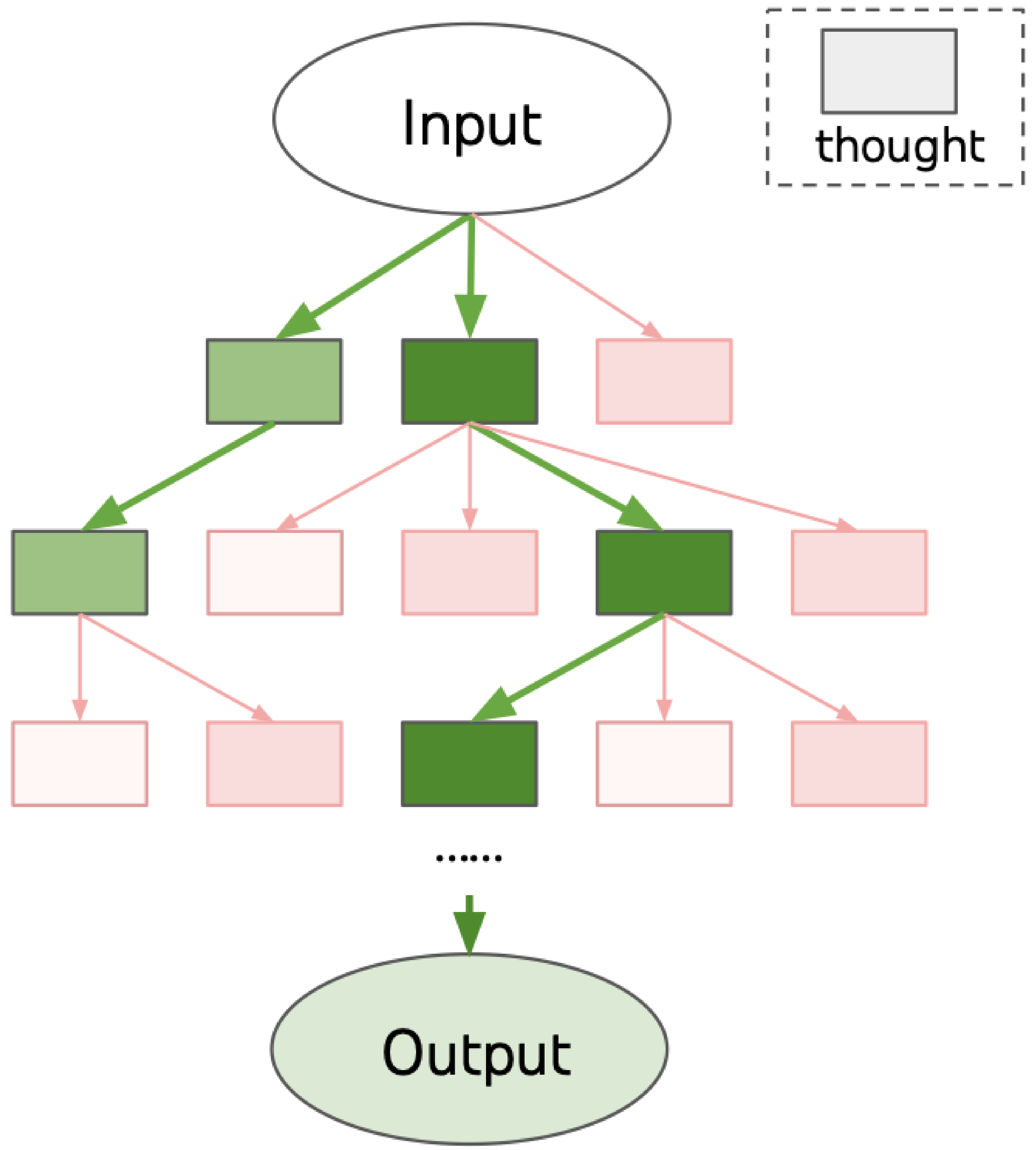
- **In-context learning**: **Demonstrating examples** (*shots*; pairs of input and output) in a prompt can teach the LLM to perform a task that it has never seen before
- **Reasoning**: Giving a model space to reason (think) before coming up with answers by **demonstrating a reasoning path** can help improve the performance of LLMs
  - **Zero-shot CoT prompting**: Eliciting an LLM to come up with **its own reasoning path**, i.e., no need to rely on few-shot demonstrations of how to reason
- **Self-consistency**: Asking an LLM to **generate multiple possible responses** (potentially with different reasoning paths), marginalize the reasoning paths to extract answers, and **choose the majority answer**

Hands-On: Tree-of-Thought Prompting for ChatGPT4PCG 2

Intelligent Computer Entertainment

Tree-of-Thought (ToT) Prompting

- **CoT**: Generate *a reasoning path* to improve the LLM's performance
- **ToT**: At each reasoning step, generate *multiple candidates **(thoughts)*** and choose the best one to proceed to the next step
- **Prompts**
  A. **Task prompt**: Generate thoughts at each reasoning step
     - CoT prompt with one-shot example
  B. **Evaluation prompt**: Evaluate a thought
  C. **Answer prompt**: Combining and formatting the final output



S. Yao et al., 'Tree of Thoughts: Deliberate Problem Solving with Large Language Models', in Thirty-seventh Conference on Neural Information Processing Systems, 2023.

Tree-of-Thought (ToT) Prompting

Let's Go Through ToT Prompting Step-by-Step

- **Thought Decomposition**: Ask an LLM to decompose a task into multiple reasoning steps (CoT)
- **Search Algorithm**:
  - Breadth-First Search (BFS)
  - Depth-First Search (DFS)

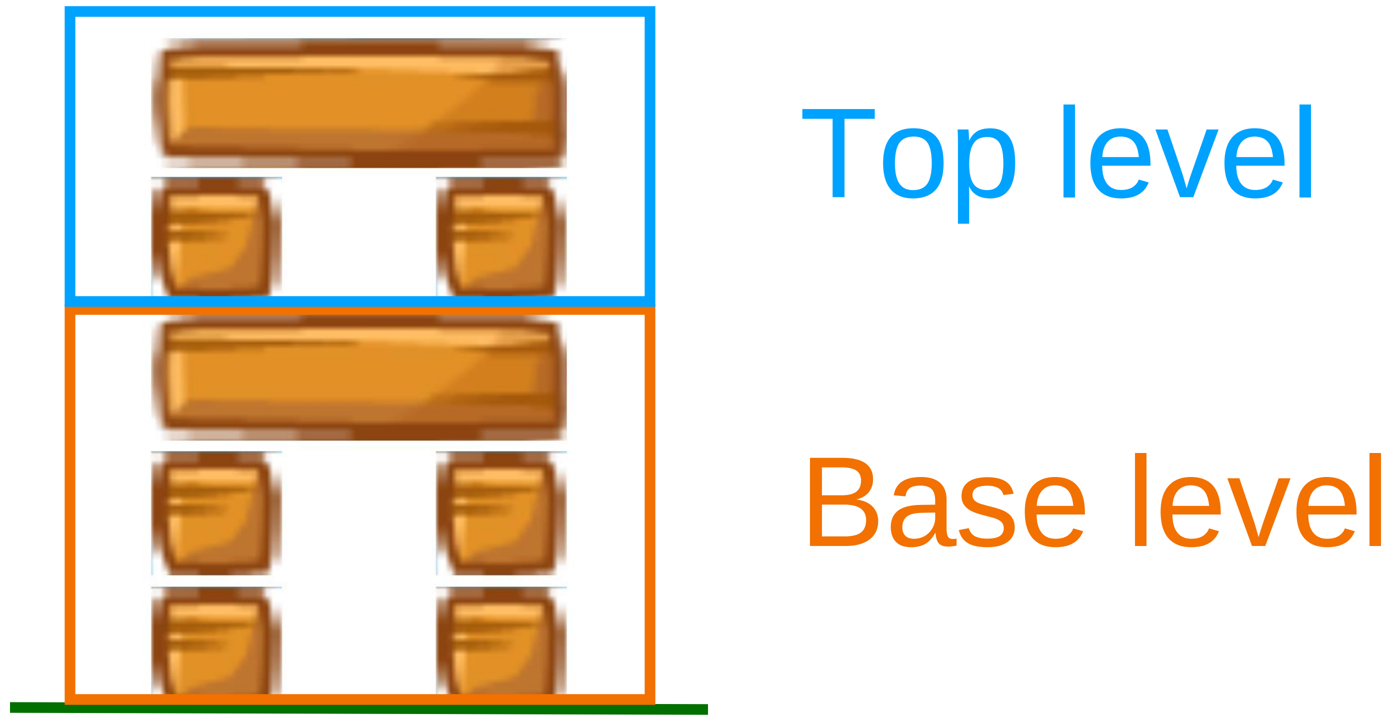
Steps

1. **Thought Generator**: Utilize CoT prompts to generate thoughts at each reasoning step
2. **Thought Evaluator**: Generate heuristics for the search algorithm
   - **Value**: Ask an LLM to generate a scalar value
   - **Vote**: Implement a step-wise self-consistency strategy

S. Yao et al., 'Tree of Thoughts: Deliberate Problem Solving with Large Language Models', in Thirty-seventh Conference on Neural Information Processing Systems, 2023.

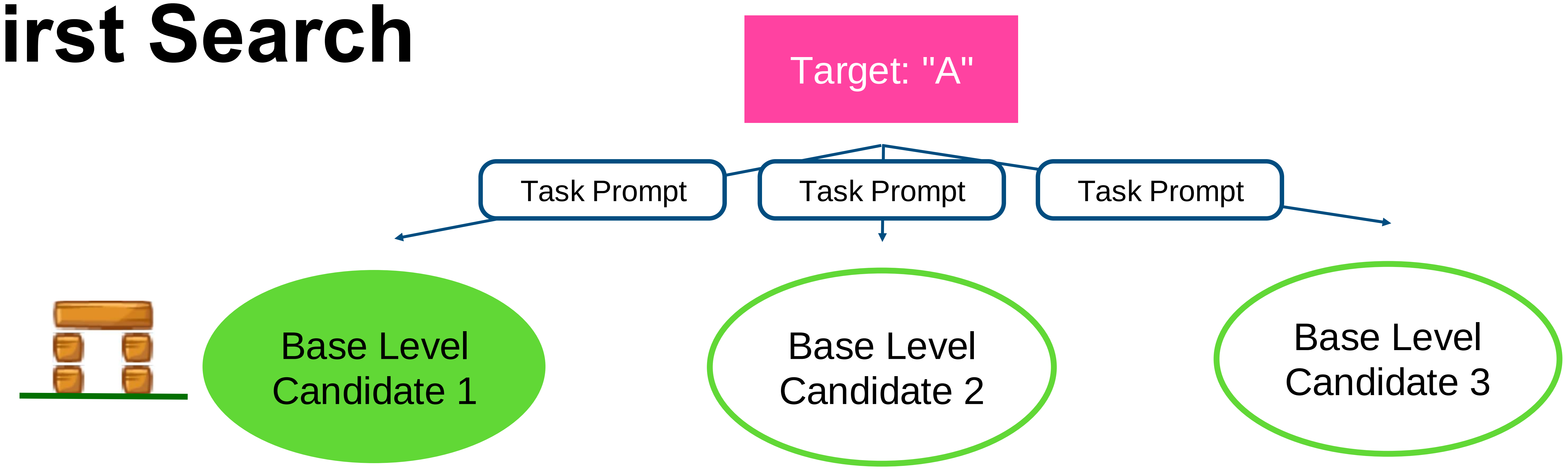
ToT Prompting For ChatGPT4PCG

- **Two** reasoning steps
  1. Generate base level
  2. Generate top level
- **Breadth-first** search
  - Our task stands to potentially gain from *exploration*



ToT Prompting For ChatGPT4PCG

Breadth-First Search



ToT Prompti… PCG

Breadth-First Searc…

Base
Cand

Base Level
andidate 3

Instruction

Use drop_block() function to generate a stable structure that looks like the character A // omitted for brevity

Input

1. Role
// omitted for brevity

Reasoning Steps Instruction

Let's follow the following steps
1. Generate the base layer of the structure
2. Generate the top layer of the structure

Only perform one step at a time.

One-shot Example

Example
Character A:

```
# Base layer
drop_block('b11', 0)
drop_block('b11', 0)
drop_block('b11', 2)
drop_block('b11', 2)
drop_block('b31', 1)
```

```
# Top layer
drop_block('b11', 0)
drop_block('b11', 2)
drop_block('b31', 1)
```

Currently, we have

Best Thoughts Combined

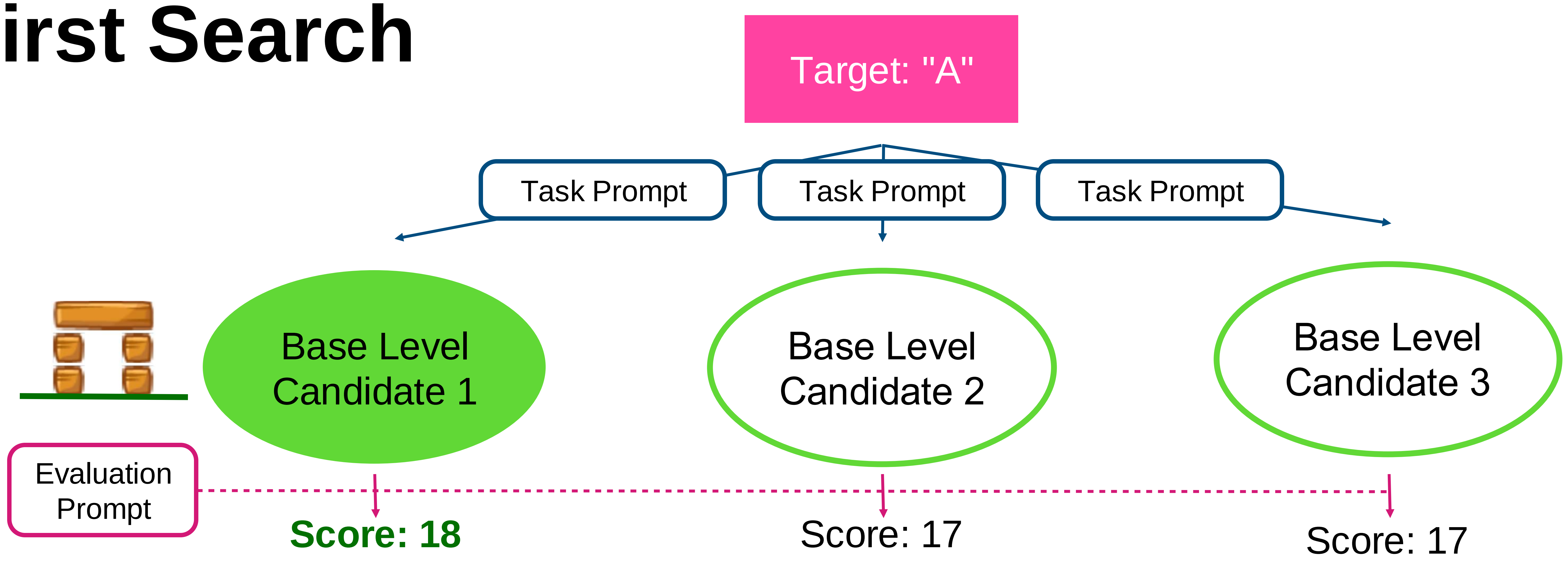
Nothing.

Next, we will perform the

Task Prompt

ToT Prompting For ChatGPT4PCG

Breadth-First Search



Intelligent Computer Entertainment

ToT Prompting For ChatGPT4PCG

Breadth-First Search

Instruction

Base L
Candida

Evaluation Prompt

Score:

e Level
didate 3

ore: 17

Thought
(Input)

The following code is used to generate a Science Birds level that resembling the uppercase English character: A. The description of the function utilized for this purpose is given below.

Provide integer scores for the following levels between 0 and 10 for two aspects stability and similarity. Provide the response in the following format.

Stability: <score>
Similarity: <score>

1. Function description
// omitted for brevity

4. Generated content to be evaluated
following steps to generate the base layer of the A structure:

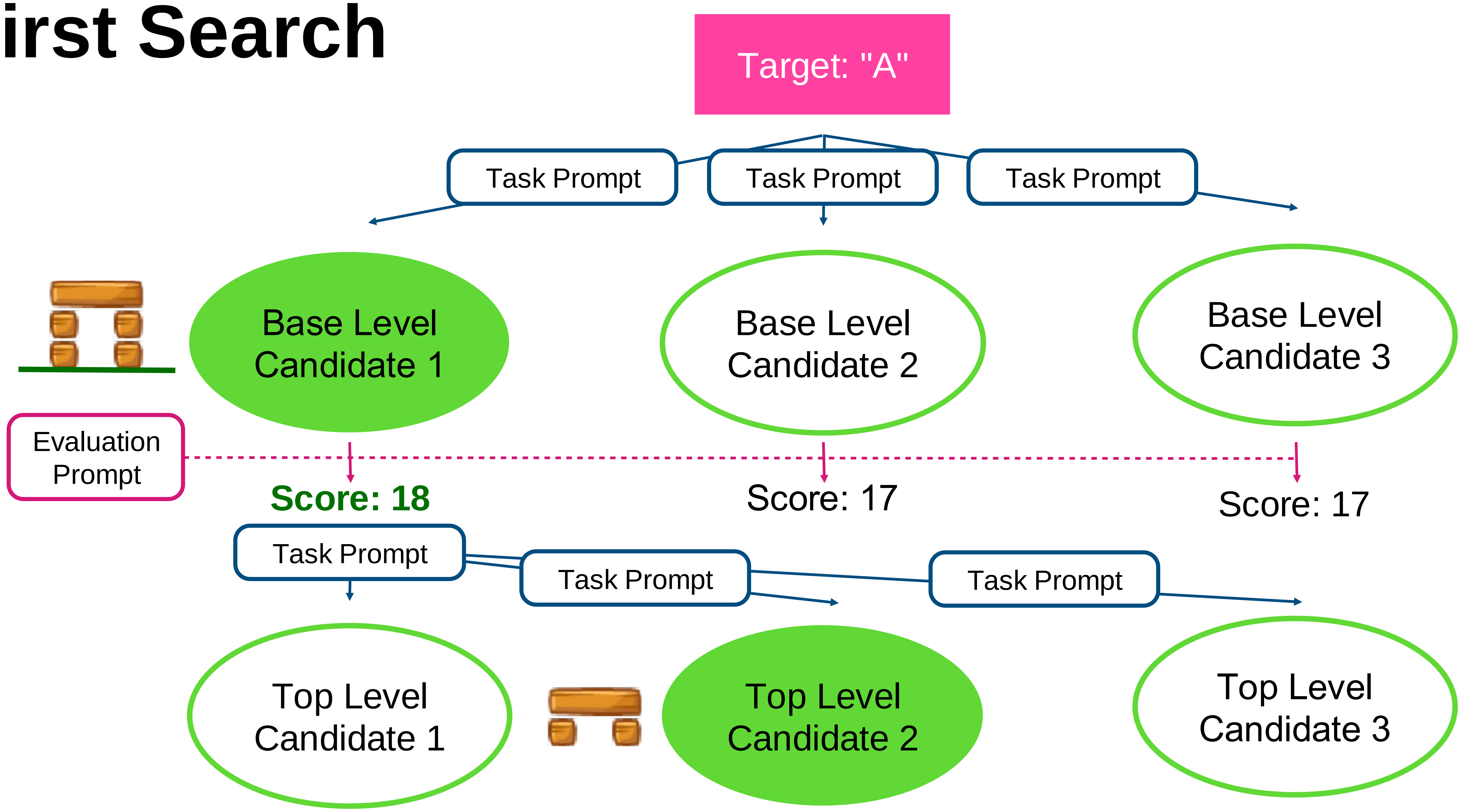
```
# Base layer
drop_block('b11', 1)
drop_block('b11', 1)
drop_block('b11', 3)
drop_block('b11', 3)
drop_block('b31', 2)
```

// omitted for brevity

Evaluation Prompt

ToT Prompting For ChatGPT4PCG

Breadth-First Search



ToT Promptin PCG

Breadth-First Search

Input

Use drop_block() function to generate a stable structure that looks like the character A // omitted for brevity

1. Role
// omitted for brevity

Instruction

Let's follow the following steps
1. Generate the base layer of the structure
2. Generate the top layer of the structure

Only perform one step at a time.

Reasoning Steps Instruction

Example
Character A:
// omitted for brevity

One-shot Example

Currently, we have

following steps to generate the base layer of the A structure:

```
# Base layer
drop_block('b11', 1)
drop_block('b11', 1)
drop_block('b11', 3)
drop_block('b11', 3)
drop_block('b31', 2)
```

// omitted for brevity

Best Thoughts Combined

Next, we will perform the

Task Prompt

Base L
Candida

Evaluation Prompt

Score:

Task Pro

Top Le
Candida

e Level
didate 3

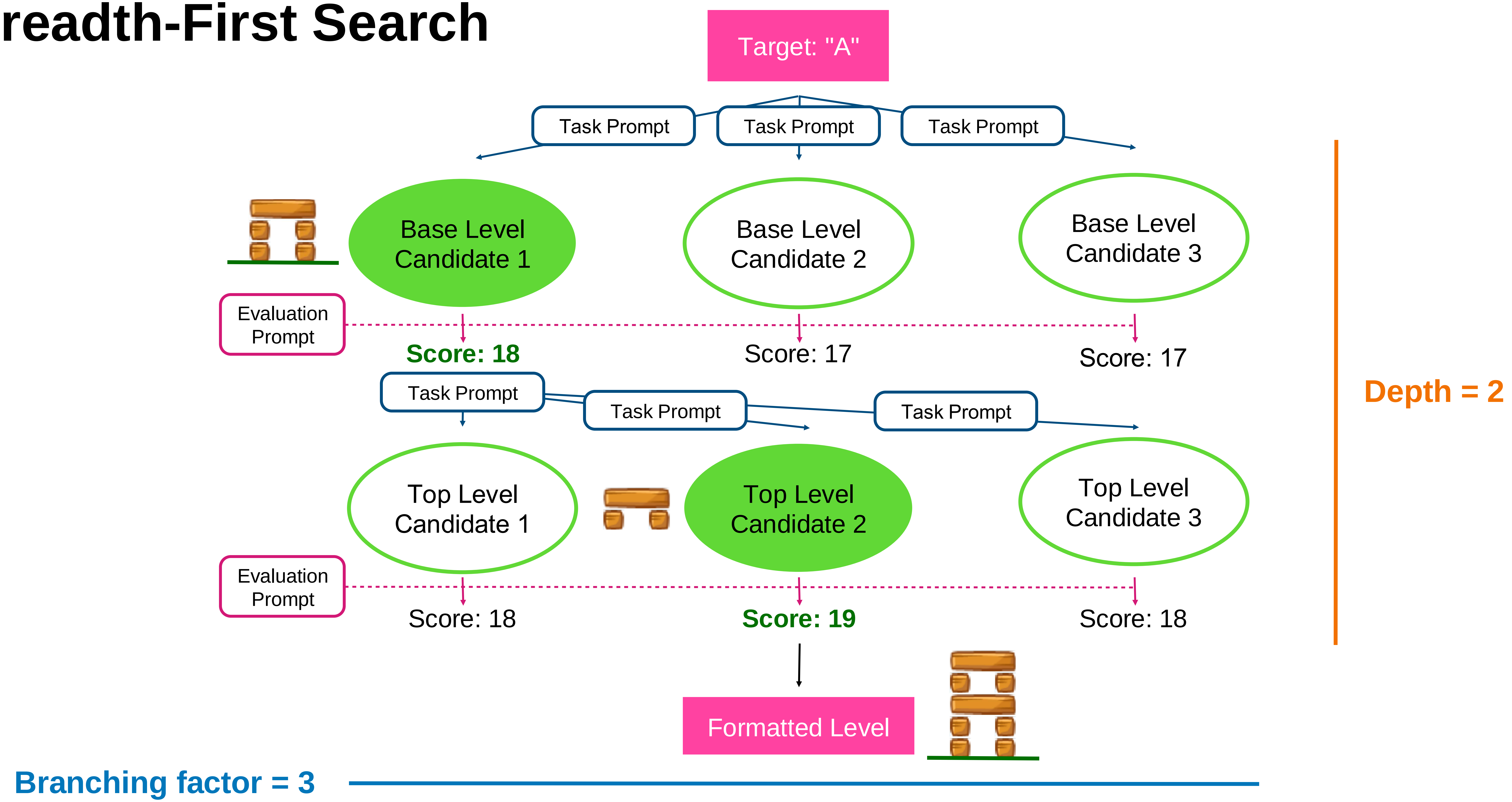
ore: 17

p Level
didate 3

Intelligent Computer Entertainment

ToT Prompting For ChatGPT4PCG

Breadth-First Search



Mini LLM4PCG Competition

For Fun 🎉!

- **ChatGPT4PCG 2 evaluation pipeline**
- **Target characters**: "I", "L", "U"
- **#Trials**: 10
- **Any LLMs** are welcome!
- **Deadline**: Aug 6, 2024 11:45AM
- **Result announcement**: Aug 7, 2024
  - chatgpt4pcg.github.io/tutorial

Coding Time!

Additional resources

chatgpt4pcg.github.io/tutorial

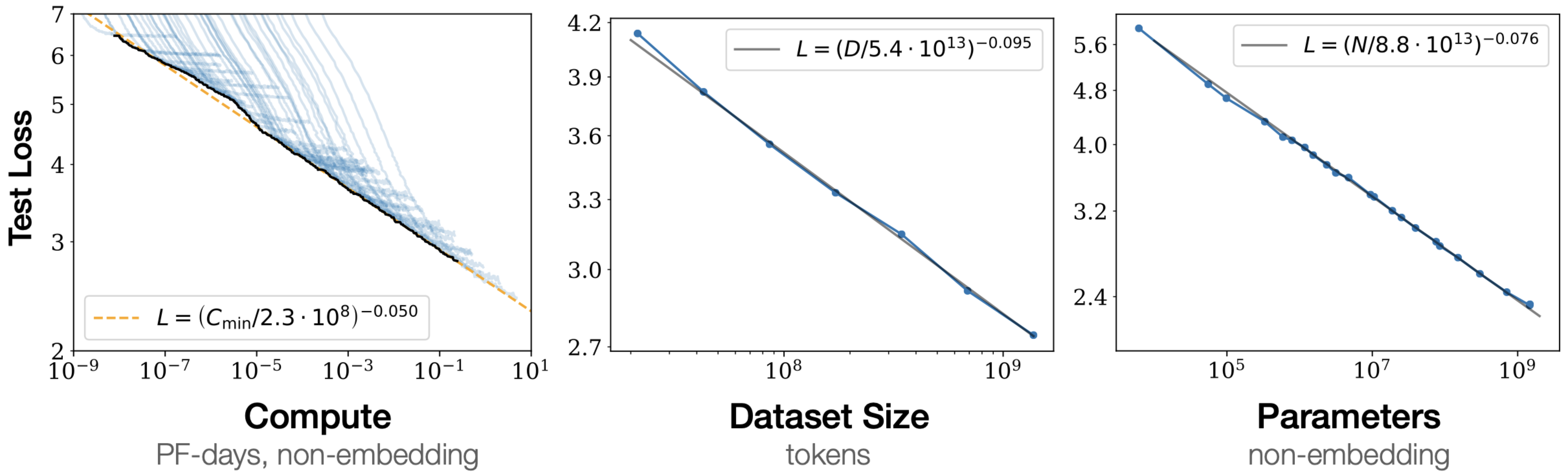
Online technical support (Jitsi)

bit.ly/pe-tutorial

Emergent Abilities

Why These LLMs Do Not Work Well?

- *Loss* scales down **smoothly** when increasing compute (parameter size, training data size)



J. Kaplan et al., 'Scaling Laws for Neural Language Models', arXiv [cs.LG]. 2020.

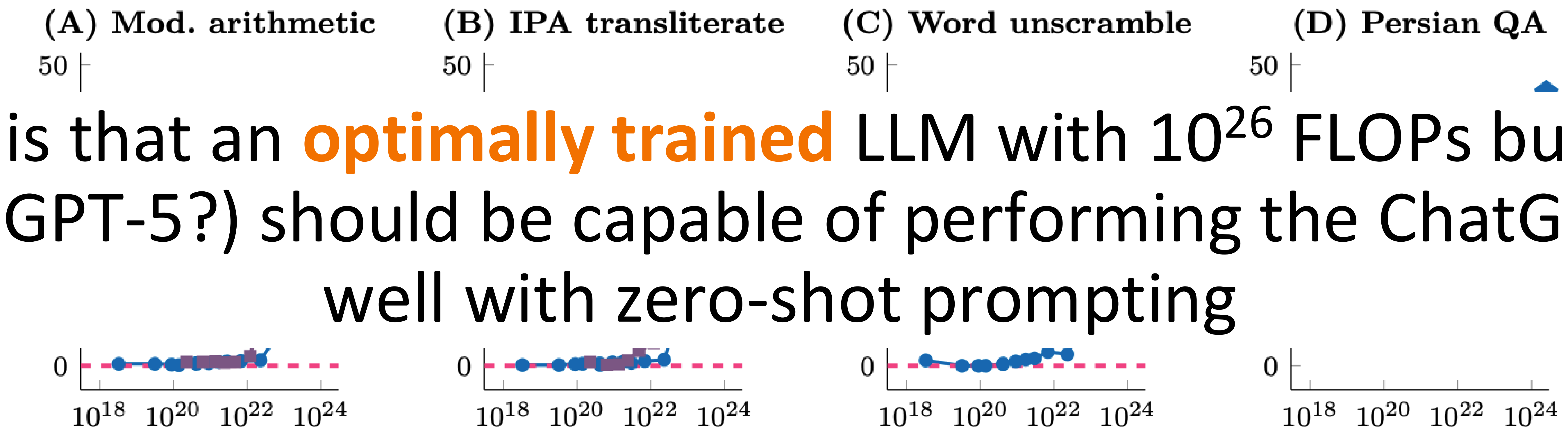
J. Wei et al., 'Emergent Abilities of Large Language Models', Transactions on Machine Learning Research, 2022.

J. Wei et al., 'Larger language models do in-context learning differently', arXiv [cs.CL]. 2023.

Emergent Abilities

Why These LLMs Do Not Work Well?

- *Loss* scales down **smoothly** when increasing compute (parameter size, training data size)
- However, some *abilities* only **emerge** when LLMs reach a certain size



My guess is that an **optimally trained** LLM with 10^{26} FLOPs budget (10T parameters, GPT-5?) should be capable of performing the ChatGPT4PCG task well with zero-shot prompting

J. Kaplan et al., ‘Scaling Laws for Neural Language Models’, arXiv [cs.LG]. 2020.

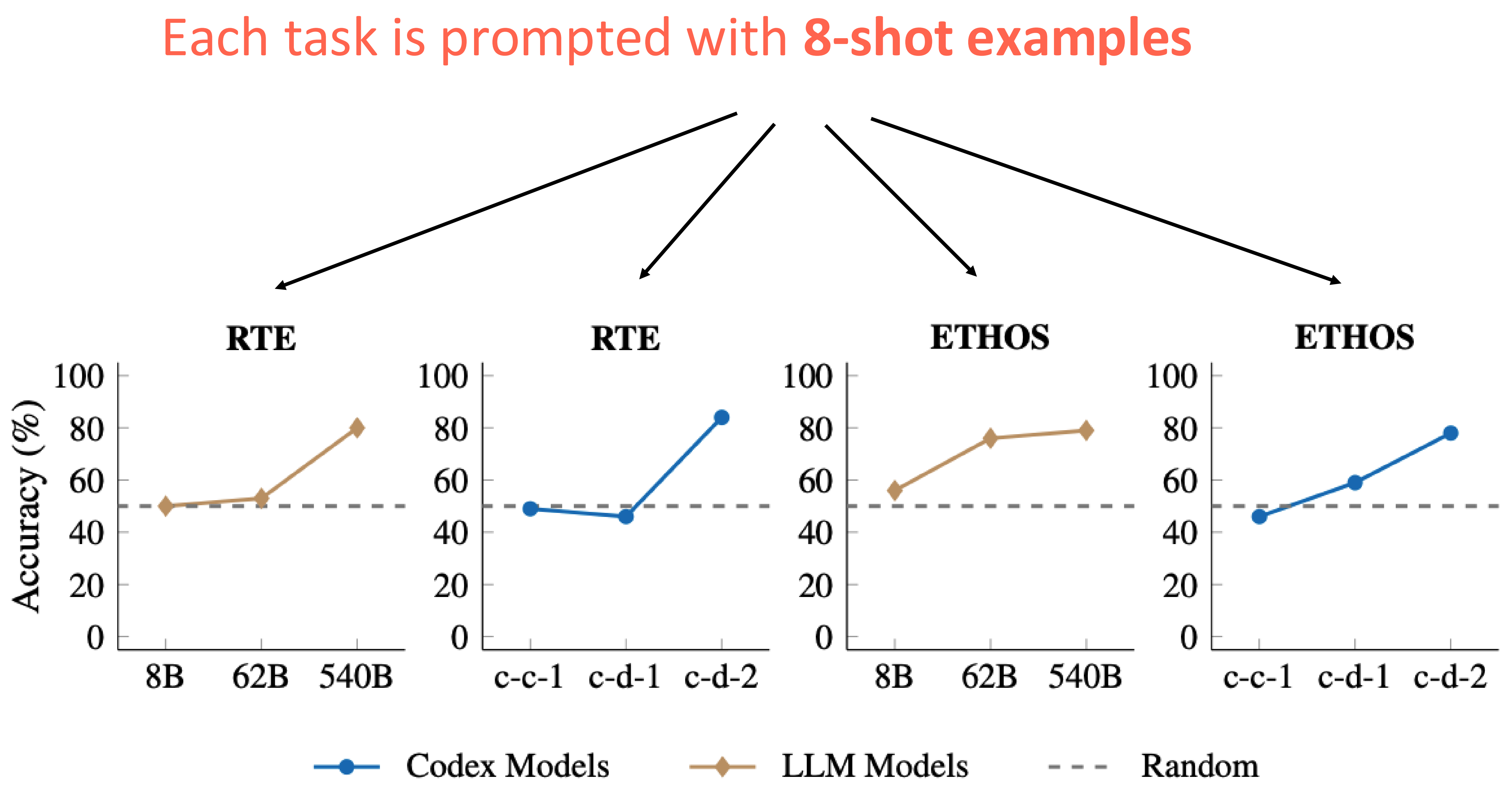
J. Wei et al., ‘Emergent Abilities of Large Language Models’, Transactions on Machine Learning Research, 2022.

J. Wei et al., ‘Larger language models do in-context learning differently’, arXiv [cs.CL]. 2023.

Emergent Abilities

Why These LLMs Do Not Work Well?

- *Loss* scales down **smoothly** when increasing compute (parameter size, training data size)
- However, some *abilities* only **emerge** when LLMs reach a certain size
  - **Few-shot prompting**, for example, only works with **large enough** LLMs



J. Kaplan et al., 'Scaling Laws for Neural Language Models', arXiv [cs.LG]. 2020.

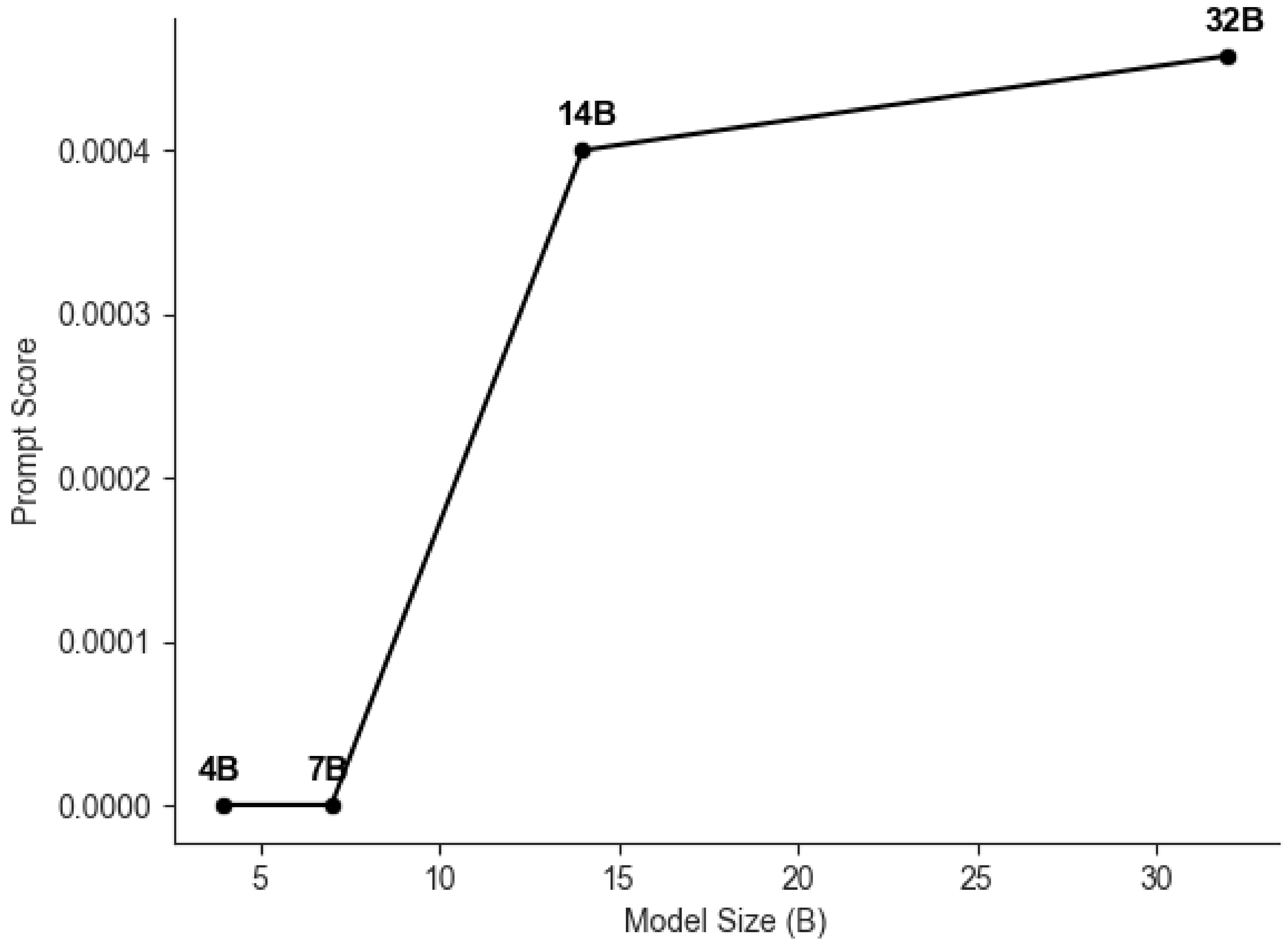
J. Wei et al., 'Emergent Abilities of Large Language Models', Transactions on Machine Learning Research, 2022.

J. Wei et al., 'Larger language models do in-context learning differently', arXiv [cs.CL]. 2023.

Emergent Abilities

Why These LLMs Do Not Work Well?

- *Loss* scales down **smoothly** when increasing compute (parameter size, training data size)
- However, some *abilities* only **emerge** when LLMs reach a certain size
  - Based on our short paper@CoG 2024, the ability to perform the **ChatGPT4PCG** task is also likely an **emergent ability**!



Aug 6, 2024 14:40 – 14:50

Towards LLM4PCG: A Preliminary Evaluation of Open-Weight Large Language Models Beyond ChatGPT4PCG

J. Kaplan et al., 'Scaling Laws for Neural Language Models', arXiv [cs.LG]. 2020.

J. Wei et al., 'Emergent Abilities of Large Language Models', Transactions on Machine Learning Research, 2022.

J. Wei et al., 'Larger language models do in-context learning differently', arXiv [cs.CL]. 2023.

To Infinity And Beyond

Beyond: Prompt Engineering

To Name A Few...

- **Multi-turn/Multi-response prompting**
  - Generated Knowledge Prompting (Jiacheng Liu+, ACL 2022)
  - Least to Most Prompting (Denny Zhou+, ICLR 2023)
- **With Tools**
  - ReAct Prompting (Shunyu Yao+, ICLR 2023)
  - Automatic Multi-Step Reasoning And Tool-Use (Bhargavi Paranjape+, 2023)
  - Self-RAG (Akari Asai+, ICLR 2024)

Intelligent Computer Entertainment

Beyond: Prompt Engineering w/ Reasoning

Part 1

- Universal Self-Adaptive Prompting (Xingchen Wan+, EMNLP 2023)
- Branch-Solve-Merge (Swarnadeep Saha+, NAACL 2024)
- Reflexion (Noah Shinn+, NeurIPS 2023)
- Contrastive Chain-of-Thought Prompting (Yew Ken Chia+, 2023)
- Plan-and-Solve Prompting (Lei Wang+, ACL 2023)

Beyond: Prompt Engineering w/ Reasoning

Part 2

- ***-of-Thoughts**
  - Boosting-of-Thoughts Prompting (Sijia Chen+, ICLR 2024)
  - Program-of-Thoughts Prompting (Wenhu Chen+, TMLR, 2023)
  - Graph-of-Thoughts Prompting (Maciej Besta+, AAAI 2024)
  - Everything-of-Thoughts Prompting (Ruomeng Ding+, 2023)
  - Thread-of-Thought Prompting (Yucheng Zhou+, 2023)

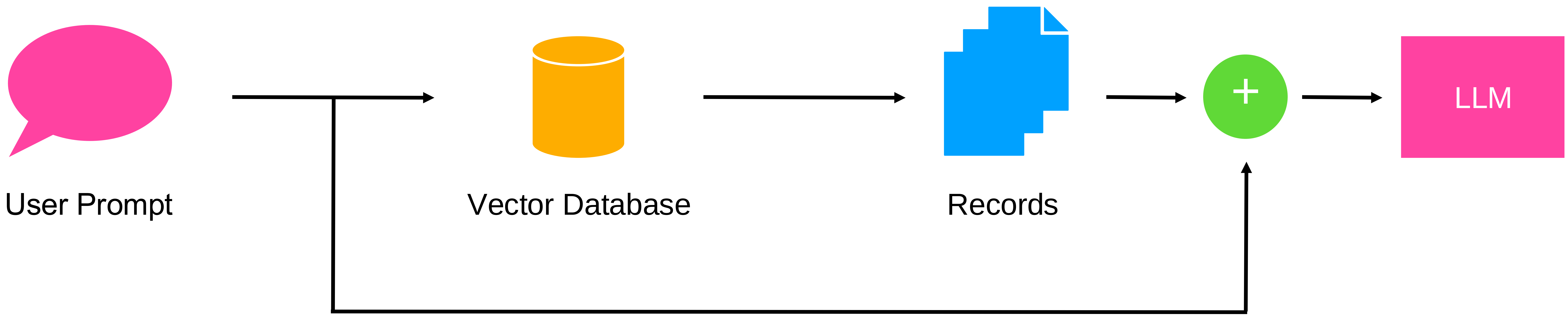
Beyond: Prompt Engineering w/ Reasoning

Part 3

- **Chain-of-***
  - Chain-of-Code Prompting (Chengshu Li+, NeurIPS 2023)
  - Chain-of-Note Prompting (Wenhao Yu+, 2023)

Retrieval Augmented Generation

Add Relevant Prerequisite Knowledge in Context



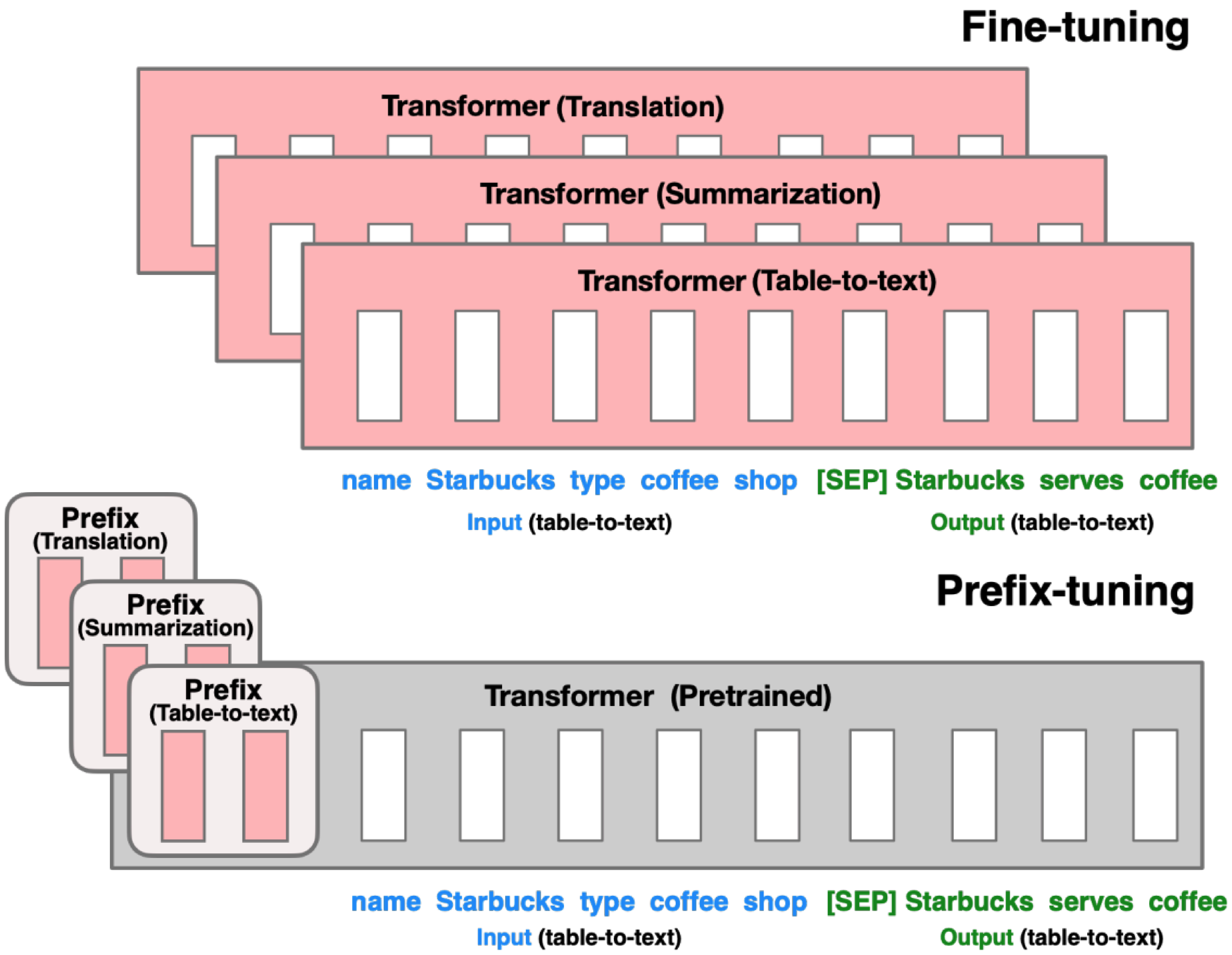
Y. Gao et al., ‘Retrieval-Augmented Generation for Large Language Models: A Survey’, arXiv [cs.CL]. 2024.

Automatic Prompt Optimization

- Automatically optimize prompts towards a pre-defined objective
  - Automatic Prompt Engineer (Yongchao Zhou+, ICLR 2023)
  - DSPy (Omar Khattab+, ICLR 2024)
- **Evolutionary-inspired APO**
  - EvoPrompt (Qingyan Guo+, ICLR 2024)
  - Promptbreeder (Chrisantha Fernando+, ICLR 2024)
  - Optimization by Prompting (Chengrun Yang+, ICLR 2024)
  - Prompt Evolution Through Examples (Taveekitworachai+, MetroInd4.0IoT 2024)

Soft Prompting

- Learnable tensor prepended or appended to an input given as a prompt to a model
  - Prompt tuning (Brian Lester+, EMNLP 2021)
  - Prefix tuning (Xiang Lisa Li+, ACL 2021)
  - P-tuning (Xiao Liu+, ACL 2022)



"The hottest new programming language is **English**"

Andrej Karpathy, CEO@Eureka Labs, Former OpenAI Founding Member

References (1/5)

- J. Liu et al., ‘Generated Knowledge Prompting for Commonsense Reasoning’, in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022, pp. 3154–3169.
- D. Zhou et al., ‘Least-to-Most Prompting Enables Complex Reasoning in Large Language Models’, in The Eleventh International Conference on Learning Representations, 2023.
- S. Yao et al., ‘ReAct: Synergizing Reasoning and Acting in Language Models’, in The Eleventh International Conference on Learning Representations, 2023.
- B. Paranjape, S. Lundberg, S. Singh, H. Hajishirzi, L. Zettlemoyer, and M. T. Ribeiro, ‘ART: Automatic multi-step reasoning and tool-use for large language models’, arXiv [cs.CL]. 2023.
- A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, ‘Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection’, in The Twelfth International Conference on Learning Representations, 2024.

ICE Intelligent Computer Entertainment

References (2/5)

- X. Wan et al., ‘Universal Self-Adaptive Prompting’, in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023, pp. 7437–7462.
- S. Saha, O. Levy, A. Celikyilmaz, M. Bansal, J. Weston, and X. Li, ‘Branch-Solve-Merge Improves Large Language Model Evaluation and Generation’, in Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 2024, pp. 8352–8370.
- N. Shinn, F. Cassano, A. Gopinath, K. R. Narasimhan, and S. Yao, ‘Reflexion: language agents with verbal reinforcement learning’, in Thirty-seventh Conference on Neural Information Processing Systems, 2023.
- Y. K. Chia, G. Chen, L. A. Tuan, S. Poria, and L. Bing, ‘Contrastive Chain-of-Thought Prompting’, arXiv [cs.CL]. 2023.
- L. Wang et al., ‘Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models’, in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 2609–2634.

References (3/5)

- S. Chen, B. Li, and D. Niu, ‘Boosting of Thoughts: Trial-and-Error Problem Solving with Large Language Models’, in The Twelfth International Conference on Learning Representations, 2024.
- W. Chen, X. Ma, X. Wang, and W. W. Cohen, ‘Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks’, Transactions on Machine Learning Research, 2023.
- M. Besta, “Graph of Thoughts: Solving Elaborate Problems with Large Language Models”, AAAI, vol. 38, no. 16, pp. 17682-17690, Mar. 2024.
- R. Ding et al., ‘Everything of Thoughts: Defying the Law of Penrose Triangle for Thought Generation’, arXiv [cs.AI]. 2024.
- Y. Zhou et al., ‘Thread of Thought Unraveling Chaotic Contexts’, arXiv [cs.CL]. 2023.
- C. Li et al., ‘Chain of Code: Reasoning with a Language Model-Augmented Code Interpreter’, in NeurIPS 2023 Foundation Models for Decision Making Workshop, 2023.
- W. Yu, H. Zhang, X. Pan, K. Ma, H. Wang, and D. Yu, ‘Chain-of-Note: Enhancing Robustness in Retrieval-Augmented Language Models’, arXiv [cs.CL]. 2023.

References (4/5)

- Y. Zhou et al., ‘Large Language Models are Human-Level Prompt Engineers’, in The Eleventh International Conference on Learning Representations, 2023.
- O. Khattab et al., ‘DSPy: Compiling Declarative Language Model Calls into State-of-the-Art Pipelines’, in The Twelfth International Conference on Learning Representations, 2024.
- Q. Guo et al., ‘Connecting Large Language Models with Evolutionary Algorithms Yields Powerful Prompt Optimizers’, in The Twelfth International Conference on Learning Representations, 2024.
- C. Fernando, D. S. Banarse, H. Michalewski, S. Osindero, and T. Rocktäschel, ‘Promptbreeder: Self-Referential Self-Improvement via Prompt Evolution’. 2024.
- C. Yang et al., ‘Large Language Models as Optimizers’, in The Twelfth International Conference on Learning Representations, 2024.
- P. Taveekitworachai, F. Abdullah, M. C. Gursesli, A. Lanata, A. Guazzini, and R. Thawonmas, ‘Prompt Evolution Through Examples for Large Language Models–A Case Study in Game Comment Toxicity Classification’, in 2024 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4.0 & IoT), 2024, pp. 22–27.

References (5/5)

- B. Lester, R. Al-Rfou, and N. Constant, ‘The Power of Scale for Parameter-Efficient Prompt Tuning’, in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 3045–3059.
- X. L. Li and P. Liang, ‘Prefix-Tuning: Optimizing Continuous Prompts for Generation’, in Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 4582–4597.
- X. Liu et al., ‘P-Tuning: Prompt Tuning Can Be Comparable to Fine-tuning Across Scales and Tasks’, in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2022, pp. 61–68.

Acknowledgement

- Prof. Ruck Thawonmas and ICE lab members, thank you for all feedback in improving this slide