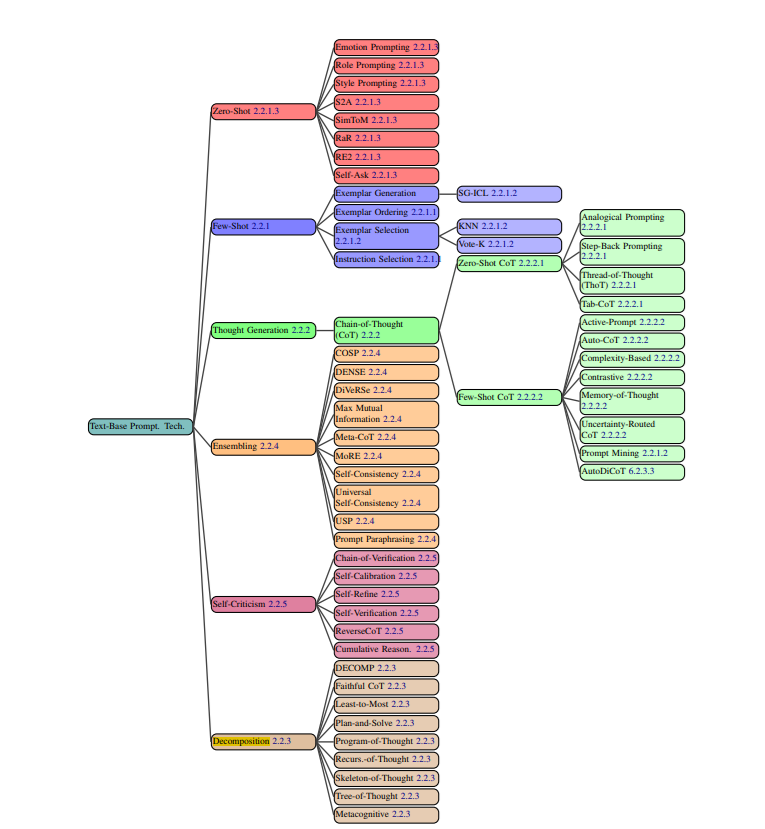
Text-Base Prompt Techniques

This document is based on "The Prompt Report: A Systematic Survey of Prompting Techniques",

<https://arxiv.org/pdf/2406.06608>

A recent paper which aims to inventory all recent text prompting techniques, some 58 have been listed here, some of them will be discussed.



Contents

[**Zero-Shot** 4](#_Toc191299894)

[Emotion Prompting 4](#_Toc191299895)

[Role Prompting 4](#_Toc191299896)

[Style Prompting 4](#_Toc191299897)

[S2A System 2 Attention 5](#_Toc191299898)

[SimToM 5](#_Toc191299899)

[RaR (Rephrase and Respond) 6](#_Toc191299900)

[RE2 (Re-reading) 6](#_Toc191299901)

[Self-Ask 6](#_Toc191299902)

[**Few-Shot** 7](#_Toc191299903)

[SG-ICL (Self-Generated In-Context Learning) 7](#_Toc191299904)

[Exemplar Ordering 8](#_Toc191299905)

[Exemplar Selection 8](#_Toc191299906)

[KNN (K-Nearest Neighbor (KNN)) 8](#_Toc191299907)

[Vote-K 9](#_Toc191299908)

[**Thought Generation** 9](#_Toc191299909)

[Chain-of-Thought (CoT) 9](#_Toc191299910)

[Zero-Shot CoT 9](#_Toc191299911)

[Few-Shot CoT 12](#_Toc191299912)

[**Ensembling** 16](#_Toc191299913)

[COSP (Consistency-based Self-adaptive Prompting) 16](#_Toc191299914)

[DENSE (Demonstration Ensembling) 16](#_Toc191299915)

[DiVeRSe 17](#_Toc191299916)

[Max Mutual Information 17](#_Toc191299917)

[Meta-CoT (Meta-Reasoning over Multiple CoTs) 18](#_Toc191299918)

[MoRE (Mixture of Reasoning Experts) 18](#_Toc191299919)

[Self-Consistency 19](#_Toc191299920)

[Universal Self-Consistency 19](#_Toc191299921)

[USP (Universal Self-Adaptive Prompting) 20](#_Toc191299922)

[Prompt Paraphrasing 20](#_Toc191299923)

[**Self-Criticism** 21](#_Toc191299924)

[Chain-of-Verification (COVE) 21](#_Toc191299925)

[Self-Calibration 22](#_Toc191299926)

[Self-Refine 22](#_Toc191299927)

[Self-Verification 22](#_Toc191299928)

[ReverseCoT (Reversing Chain-of-Thought) 23](#_Toc191299929)

[Cumulative Reason 24](#_Toc191299930)

[**Decomposition** 24](#_Toc191299931)

[DECOMP (Decomposed Prompting) 24](#_Toc191299932)

[Faithful CoT 25](#_Toc191299933)

[Least-to-Most 25](#_Toc191299934)

[Plan-and-Solve 26](#_Toc191299935)

[Program-of-Thought 26](#_Toc191299936)

[Recurs.-of-Thought 27](#_Toc191299937)

[Skeleton-of-Thought 27](#_Toc191299938)

[Tree-of-Thought 28](#_Toc191299939)

[Metacognitive 28](#_Toc191299940)

# **Zero-Shot**

Prompting uses zero exemplars. There are a number of well-known standalone zero-shot techniques as well as zero-shot techniques combined with another concept (e.g. Chain of Thought)

## 

## Emotion Prompting

Incorporates phrases of psychological relevance to humans (e.g., "This is important to my career") into the prompt, which may lead to improved LLM performance on benchmarks and open-ended text generation.

**Original Prompt**: "What are the benefits of machine learning?"

**Prompt with Emotion Prompting**: "What are the benefits of machine learning? This is important to my career."

## Role Prompting

Also known as persona prompting (Schmidt et al., 2023; Wang et al., 2023l), assigns a specific role to the GenAI in the prompt. For example, the user might prompt it to act like "Madonna"or a "travel writer". This can create more desirable outputs for open-ended tasks

**Original Prompt**: "What are the benefits of regular physical exercise?"

**Prompt with Role Prompting**: "As a fitness coach, explain the benefits of regular physical exercise."

## Style Prompting

Involves specifying the desired style, tone, or genre in the prompt to shape the output of a GenAI. A similar effect can be achieved using role prompting.

**Original Prompt:** "Describe a sunset."

**Prompt with Style Prompting**: "Describe a sunset in a poetic style."

## S2A **System 2 Attention**

**System 2 Attention** first asks an LLM to rewrite the prompt and remove any information unrelated to the question therein. Then, it passes this new prompt into an LLM to retrieve a final response.

**Original Prompt**: "Can you tell me how solar panels work and also what the weather is like today?"

**Step 1**: Rewrite the prompt to remove unrelated information: The model rewrites the prompt to focus only on the relevant question:

"Can you tell me how solar panels work?"

**Step 2**: Pass the new prompt to the LLM: The model generates a response based on the rewritten prompt:

"Solar panels work by converting sunlight into electricity using photovoltaic cells.

When sunlight hits these cells, it excites electrons, creating an electric current.

This current is then used to power electrical devices or stored in batteries for later use."

## SimToM

Given the question, it attempts to establish the set of facts one person knows, then answer the

question based only on those facts.

**Prompt :**

**Question:** "David knows that the store opens at 9 AM, and Maria knows that it takes 15 minutes to walk there. Can David arrive at the store on time if he leaves at 8:45 AM?"

SimToM Process:

Step 1: Establish facts known by David: "The store opens at 9 AM."

Step 2: Answer based on David's facts: "David knows that he can arrive at the store on time if he leaves at 8:45 AM."

## 

## RaR (Rephrase and Respond)

**Rephrase and Respond (RaR)** instructs the LLM to rephrase and expand the question before generating the final answer

**Prompt :**

Original Question: "What is the capital of France?"

RaR Process:

Step 1: Rephrase and expand the question: "Can you tell me what the capital city of France is, including some details about its significance?"

Step 2: Respond: "The capital city of France is Paris. It is known for its rich history,

cultural landmarks like the Eiffel Tower, and its role as a major European center for art, fashion, and cuisine."

## RE2 (Re-reading)

**Re-reading** (RE2) (Xu et al., 2023) adds the phrase "Read the question again:" to the prompt in

addition to repeating the question

prompt:

**Original Question:** "What is the tallest mountain in the world?"

RE2 Process:

Step 1: Add the phrase and repeat the question: "Read the question again: What is the tallest mountain in the world?"

Step 2: Respond: "The tallest mountain in the world is Mount Everest, which stands at 8,848 meters (29,029 feet) above sea level."

## 

## Self-Ask

**Self-Ask** (Press et al., 2022) prompts LLMs to first decide if they need to ask follow up questions

for a given prompt. If so, the LLM generates these questions, then answers them and finally answers the original question.

**Prompt:**

Original Question: "What are the benefits of exercise?"

Self-Ask Process:

Step 1: Decide if follow-up questions are needed: "Do I need more information to answer this question comprehensively?"

Step 2: Generate follow-up questions: "What types of exercise are we considering? Are we looking at physical, mental, or both types of benefits?"

Step 3: Answer follow-up questions: "Physical exercise includes activities like running, swimming, and weightlifting. Mental benefits include improved mood and reduced anxiety."

Step 4: Answer the original question: "The benefits of exercise include improved cardiovascular health, stronger muscles, better mental health, and increased longevity."

# **Few-Shot**

Selecting exemplars for a prompt is a difficult task–performance depends significantly on various factors of the exemplars (Dong et al., 2023), and only a limited number of exemplars fit in the typicalLLM’s context window

## 

## SG-ICL (Self-Generated In-Context Learning)

**Self-Generated In-Context Learning** (SG-ICL)(Kim et al., 2022) leverages a GenAI to automatically generate exemplars.

**Original Question:** "How do plants grow?"

SG-ICL Process:

Step 1: Generate exemplars: "Q: How do seeds germinate? A: Seeds germinate by absorbing water, which activates enzymes that start the growth process."

Step 2: Use generated exemplar to answer the original question: "Q: How do plants grow? A: Plants grow by absorbing water and nutrients from the soil,

using sunlight for photosynthesis, and developing roots, stems, and leaves."

## Exemplar Ordering

The order of exemplars affects model behavior

**Original Question**: "How does the water cycle work?"

Exemplar Ordering Process:

Order 1: "Q: How does photosynthesis work? A: Photosynthesis is the process by which plants convert light energy into chemical energy. Q: How does the water cycle work? A: The water cycle involves evaporation, condensation, and precipitation."

Order 2: "Q: How do plants grow? A: Plants grow by absorbing water and nutrients from the soil. Q: How does the water cycle work? A: The water cycle involves evaporation, condensation, and precipitation."

Impact: The order in which examples are presented can influence the model's understanding and response, leading to variations in accuracy.

## 

## Exemplar Selection

While they may not improve correctness, instructions in few-shot prompts can still guide auxiliary output attributes like writing style (Roy et al., 2023)

### 

### KNN (K-Nearest Neighbor (KNN))

**K-Nearest Neighbor (KNN)** (Liu et al., 2021) is part of a family of algorithms that selects exemplars similar to boost performance. Although effective, employing KNN during prompt generation may be time and resource intensive

**Original Question:** "How does gravity work?"

KNN Process:

Step 1: Select exemplars similar to the test question: "Q: How does magnetism work? A: Magnetism is the force exerted by magnets when they attract or repel each other."

Step 2: Use the selected exemplar to answer the original question: "Q: How does gravity work? A: Gravity is the force that attracts two bodies towards each other, such as the Earth and objects on it."

### 

### 

### Vote-K

**Vote-K** (Su et al., 2022) is another method to select similar exemplars to the test sample. In one stage, a model proposes useful unlabeled candidate exemplars

**Original Question**: "How does the digestive system work?"

Vote-K Process:

Step 1: Model proposes useful unlabeled candidate exemplars: "Q: How does the respiratory system work? A: The respiratory system allows us to breathe by taking in oxygen and expelling carbon dioxide."

Step 2: Annotator labels the candidate exemplars.

Step 3: Use the labeled pool for Few-Shot Prompting: "Q: How does the digestive system work? A: The digestive system breaks down food into nutrients that the body can absorb and use for energy, growth, and cell repair."

## 

# **Thought Generation**

## 

## Chain-of-Thought (CoT)

**Chain-of-Thought (CoT)** Prompting (Wei et al.,2022b) leverages few-shot prompting to encourage the LLM to express its thought process before delivering its final answer.6 This technique is occasionally referred to as Chain-of-Thoughts (Tutunov et al., 2023; Besta et al., 2024; Chen et al., 2023d). It has been demonstrated to significantly enhance the LLM’s performance in mathematics and reasoning tasks. In Wei et al. (2022b), the prompt includes an exemplar featuring a question, a reasoning path,and the correct answer (Figure 2.8).

### 

### Zero-Shot CoT

The most straightforward version of CoT contains zero exemplars. It involves appending a thought inducing phrase like "Let’s think step by step." (Kojima et al., 2022) to the prompt. Other suggested thought-generating phrases include "First,let’s think about this logically" (Kojima et al.,2022). Zhou et al. (2022b) uses LLMs to generate "Let’s work this out in a step by step way to be sure we have the right answer". Yang et al. (2023a) searches for an optimal thought inducer. Zero-Shot CoT approaches are attractive as they don’t require exemplars and are generally task agnostic.

#### Analogical Prompting

**Analogical Prompting** (Yasunaga et al., 2023) is similar to SG-ICL, and automatically generates exemplars that include CoTs. It has demonstrated improvements in mathematical reasoning and code generation tasks.

**Original Question**: "What is the sum of 8 and 5?"

Analogical Prompting Process:

Step 1: Automatically generate exemplars with CoTs: "Q: What is the sum of 3 and 2? A: To find the sum of 3 and 2, we add them together: 3 + 2 = 5."

Step 2: Use the generated exemplar to answer the original question: "Q: What is the sum of 8 and 5? A: To find the sum of 8 and 5, we add them together: 8 + 5 = 13."

#### 

#### Step-Back Prompting

Step-Back Prompting (Zheng et al., 2023c) is a modification of CoT where the LLM is first asked a generic, high-level question about relevant concepts or facts before delving into reasoning. This approach has improved performance significantly on multiple reasoning benchmarks for both PaLM2L and GPT-4.

**Original Question:** "How does a computer process data?"

Step-Back Prompting Process:

Step 1: Ask a high-level question: "What are the basic components of a computer system?"

Step 2: Respond with relevant concepts: "The basic components of a computer system include the central processing unit (CPU), memory, storage, and input/output devices."

Step 3: Use these concepts to answer the original question: "A computer processes data by using the CPU to execute instructions, memory to store data temporarily, and storage to save data permanently. Input/output devices allow the computer to interact with the external environment."

#### 

#### Thread-of-Thought (ThoT) Tabular Chain-of-Thought

**Thread-of-Thought (ThoT)** Prompting (Zhou et al., 2023) consists of an improved thought inducer for CoT reasoning. Instead of "Let’s think step by step," it uses "Walk me through this context in manageable parts step by step, summarizing and analyzing as we go." This thought inducer works well in question-answering and retrieval settings, especially when dealing with large, complex context

**Original Question**: "What are the main causes of climate change?"

ThoT Process:

Step 1: High-level prompt: "Walk me through this context in manageable parts step by step, summarizing and analyzing as we go."

Step 2: Summarize and analyze: "The main causes of climate change include the burning of fossil fuels, deforestation, and industrial activities. These activities release greenhouse gases like carbon dioxide and methane into the atmosphere, which trap heat and cause global temperatures to rise."

Step 3: Answer the original question: "The main causes of climate change are the burning of fossil fuels, deforestation, and industrial activities, which release greenhouse gases that trap heat in the atmosphere."

#### Tab-CoT (Tabular Chain-of-Thought)

**Tabular Chain-of-Thought** (Tab-CoT) (Jin and Lu, 2023) consists of a Zero-Shot CoT prompt that makes the LLM output reasoning as a markdown table. This tabular design enables the LLM to improve the structure and thus the reasoning of its output

**Original Question**: "What are the main causes of climate change?"

Tab-CoT Process:

Step 1: Generate reasoning as a markdown table:

| Step | Reasoning |

|------|---------------------------------------------------------------------------|

| 1 | The burning of fossil fuels releases carbon dioxide into the atmosphere. |

| 2 | Deforestation reduces the number of trees that can absorb carbon dioxide. |

| 3 | Industrial activities emit various greenhouse gases.

|

Step 2: Use the table to answer the original question: "The main causes of climate change include the burning of fossil fuels, deforestation, and industrial activities, which release greenhouse gases into the atmosphere."

### Few-Shot CoT

This set of techniques presents the LLM with multiple exemplars, which include chains-of-thought. This can significantly enhance performance. This technique is occasionally referred to as ManualCoT (Zhang et al., 2022b) or Golden CoT (Del and Fishel, 2023)

#### 

#### Active-Prompt

Active Prompting (Diao et al., 2023) starts with some training questions/exemplars, asks the LLM to solve them, then calculates uncertainty (disagreement in this case) and asks human annotators to rewrite the exemplars with highest uncertainty

**Original Question**: "What is the capital of Brazil?"

**Active Prompting Process:**

Step 1: Training questions/exemplars: "Q: What is the capital of France? A: The capital of France is Paris."

Step 2: LLM solves the question: "Q: What is the capital of Brazil? A: The capital of Brazil is Brasília."

Step 3: Calculate uncertainty: If there is disagreement or uncertainty in the LLM's response, human annotators rewrite the exemplar to clarify: "Q: What is the capital of Brazil? A: The capital of Brazil is Brasília, not Rio de Janeiro."

#### Auto-CoT (Automatic Chain-of-Thought)

Automatic Chain-of-Thought (Auto-CoT) Prompting (Zhang et al., 2022b) uses Wei et al. (2022b)’s Zero-Shot prompt to automatically generate chains of thought. These are then used to build a Few-Shot CoT prompt for a test sample.

Original Question: "What is the sum of 15 and 27?"

**Auto-CoT Process:**

Step 1: Automatically generate chains of thought: "Let's think step by step. First, add the tens: 10 + 20 = 30. Then, add the units: 5 + 7 = 12. Finally, add the results: 30 + 12 = 42."

Step 2: Use the generated chain to answer the original question: "The sum of 15 and 27 is 42."

#### Complexity-Based

**Complexity-based Prompting** (Fu et al., 2023b) involves two major modifications to CoT. First, it selects complex examples for annotation and inclusion in the prompt, based on factors like question length or reasoning steps required. Second,during inference, it samples multiple reasoning chains (answers) and uses a majority vote among chains exceeding a certain length threshold, under the premise that longer reasoning indicates higher answer quality. This technique has shown improvements on three mathematical reasoning datasets.

**Original Question:** "What is the sum of 123 and 456?"

**Complexity-based Prompting Process:**

Step 1:

Select complex examples: "Q: What is the sum of 78 and 34? A: To find the sum, add the tens: 70 + 30 = 100. Then add the units: 8 + 4 = 12. Finally, add the results: 100 + 12 = 112."

Step 2:

Sample multiple reasoning chains and use majority vote: "Q: What is the sum of 123 and 456? A: To find the sum, add the hundreds: 100 + 400 = 500. Add the tens: 20 + 50 = 70. Add the units: 3 + 6 = 9. Finally, add the results: 500 + 70 + 9 = 579."

#### Contrastive

Contrastive CoT Prompting (Chia et al., 2023) adds both exemplars with incorrect and correct explanations to the CoT prompt in order to show the LLM how not to reason. This method has shown significant improvement in areas like Arithmetic Reasoning and Factual QA

**Original Question:** "How many continents are there on Earth?"

**Contrastive CoT Process:**

Correct Explanation: "There are seven continents on Earth: Africa, Antarctica, Asia, Europe, North America, Australia, and South America."

Incorrect Explanation: "There are five continents on Earth: Africa, Asia, Europe, North America, and South America."

Answer: "There are seven continents on Earth."

#### 

#### Memory-of-Thought

Memory-of-Thought Prompting (Li and Qiu,2023b) leverage unlabeled training exemplars to build Few-Shot CoT prompts at test time. Before test time, it performs inference on the unlabeled training exemplars with CoT. At test time, it retrieves similar instances to the test sample. This technique has shown substantial improvements in benchmarks like Arithmetic, commonsense, and factual reasoning

**Original Question:** "Explain the process of condensation."

**MoT Process:**

Step 1: Perform inference on unlabeled training exemplars: "Q: Explain the process of evaporation. A: Evaporation is the process where liquid water turns into vapor due to heat."

Step 2: Retrieve similar instances at test time: "Q: Explain the process of condensation. A: Condensation is the process where water vapor turns into liquid water when it cools down."

#### Uncertainty-Routed

Uncertainty-Routed CoT Prompting (Google,2023) samples multiple CoT reasoning paths, then selects the majority if it is above a certain threshold (calculated based on validation data). If not, it samples greedily and selects that response. This method demonstrates improvement on the MMLU benchmark for both GPT-4 and Gemini Ultra models

**Original Question:** "What is the sum of 45 and 67?"

**Uncertainty-Routed CoT Process:**

Step 1: Sample multiple CoT reasoning paths:

Path 1: "45 + 67 = 112"

Path 2: "45 + 67 = 110"

Path 3: "45 + 67 = 112"

Step 2: Select the majority if above threshold: "The sum of 45 and 67 is 112."

#### 

#### Prompt Mining

Prompt Mining (Jiang et al., 2020) is the process of discovering optimal "middle words" in prompts through large corpus analysis. These middle words are effectively prompt templates. For example, instead of using the common "Q: A:" format for fewshot prompts, there may exist something similar that occurs more frequently in the corpus. Formats which occur more often in the corpus will likely lead to improved prompt performance.

**Original Question:** "What is the capital of Italy?"

**Prompt Mining Process:**

Common Format: "Q: What is the capital of France? A: The capital of France is Paris."

Discovered Optimal Format: "Tell me, what is the capital of France? The capital of France is Paris."

Answer: "Tell me, what is the capital of Italy? The capital of Italy is Rome."

#### AutoDiCoT

One-Shot AutoDiCot + Full Context. After performing 10-shot prompting, the prompt engineer observed that the 12th item in the development set was being incorrectly being labeled as a positive instance, and began experimenting with ways of modifying the prompting such that the model would get that item correct. In order to get a sense of why this mislabeling was taking place, the prompt engineer prompted the LLM to generate an explanation of why the 12th item would have been labeled the way it was.1

**Original Question:** "Is this email spam?"

**One-Shot AutoDiCot + Full Context Process:**

Step 1: Perform 10-shot prompting: "Q: Is this email spam? A: Yes, because it contains suspicious links."

Step 2: Observe mislabeling: The 12th email is incorrectly labeled as spam.

Step 3: Modify prompting and generate explanation: "Q: Why was the 12th email labeled as spam? A: The 12th email was labeled as spam because it contained phrases commonly associated with spam, but it was actually a legitimate email."

# **Ensembling**

In GenAI, ensembling is the process of using multiple prompts to solve the same problem, then aggregating these responses into a final output. In many cases, a majority vote—selecting the most frequent response—is used to generate the final output. Ensembling techniques reduce the variance of LLM outputs and often improving accuracy, but come with the cost of increasing the number of model calls needed to reach a final answer.

## COSP (Consistency-based Self-adaptive Prompting)

Consistency-based Self-adaptive Prompting (COSP) (Wan et al., 2023a) constructs Few-Shot CoT prompts by running Zero-Shot CoT with Self-Consistency on a set of examples then selecting a high agreement subset of the outputs to be included in the final prompt as exemplars. It again performs Self-Consistency with this final prompt

**Prompt:** "Explain the process of photosynthesis."

Zero-Shot CoT Output: "Photosynthesis is the process by which green plants and some other organisms use sunlight to synthesize foods with the help of chlorophyll."

Few-Shot CoT Prompt: "Explain the process of photosynthesis. Photosynthesis is the process by which green plants and some other organisms use sunlight to synthesize foods with the help of chlorophyll."

## DENSE (Demonstration Ensembling)

Demonstration Ensembling (DENSE) (Khalifa et al., 2023) creates multiple few-shot prompts, each containing a distinct subset of exemplars from the training set. Next, it aggregates over their outputs to generate a final response .

Prompt: "What are the main causes of climate change?"

Few-Shot Prompts:

Prompt 1: "What are the main causes of climate change? One major cause is the burning of fossil fuels."

Prompt 2: "What are the main causes of climate change? Deforestation is another significant factor."

Prompt 3: "What are the main causes of climate change? Industrial activities also contribute significantly."

Final Output: "The main causes of climate change include the burning of fossil fuels, deforestation, and industrial activities."

## 

## DiVeRSe

DiVeRSe (Li et al., 2023i) creates multiple prompts for a given problem then performs SelfConsistency for each, generating multiple reasoning paths. They score reasoning paths based on each step in them then select a final response.

**Prompt**: "How does photosynthesis work?"

Multiple Prompts:

Prompt 1: "How does photosynthesis work? Photosynthesis is the process by which plants convert light energy into chemical energy."

Prompt 2: "How does photosynthesis work? During photosynthesis, plants take in carbon dioxide and water and convert them into glucose and oxygen."

Prompt 3: "How does photosynthesis work? The process occurs in the chloroplasts of plant cells, where chlorophyll captures light energy."

Final Output (ASSISTANT): "Photosynthesis is the process by which plants convert light energy into chemical energy. During photosynthesis, plants take in carbon dioxide and water and convert them into glucose and oxygen. The process occurs in the chloroplasts of plant cells, where chlorophyll captures light energy."

## Max Mutual Information

Mutual Information Method (Sorensenet al., 2022) creates multiple prompt templates with varied styles and exemplars, then selects the optimal template as the one that maximizes mutual information between the prompt and the LLM’s outputs.

**Prompt:** "Explain the significance of the French Revolution."

Multiple Prompt Templates:

Template 1: "Explain the significance of the French Revolution. Consider its impact on modern democracy."

Template 2: "Discuss the importance of the French Revolution. Focus on its effects on social and political structures."

Template 3: "What was the significance of the French Revolution? Highlight its influence on contemporary governance."

Final Output (ASSISTANT): "The French Revolution was significant for its profound impact on modern democracy, social and political structures, and contemporary governance."

## Meta-CoT (Meta-Reasoning over Multiple CoTs)

Meta-Reasoning over Multiple CoTs (Yoran et al., 2023) is similar to universal SelfConsistency; it first generates multiple reasoning chains (but not necessarily final answers) for a given problem. Next, it inserts all of these chains in a single prompt template then generates a final answer from them.

**Prompt:** "What are the economic impacts of climate change?"

**Multiple Reasoning Chains**:

Chain 1: "Climate change can lead to increased costs due to natural disasters."

Chain 2: "It can affect agricultural productivity and food security."

Chain 3: "There may be shifts in energy demand and supply."

Final Output (ASSISTANT): "The economic impacts of climate change include increased costs due to natural disasters, effects on agricultural productivity and food security, and shifts in energy demand and supply."

## MoRE (Mixture of Reasoning Experts)

Mixture of Reasoning Experts (MoRE) (Si et al.,2023d) creates a set of diverse reasoning experts by using different specialized prompts for different reasoning types (such as retrieval augmentation prompts for factual reasoning, Chain-of-Thought reasoning for multi-hop and math reasoning, and generated knowledge prompting for commonsense reasoning). The best answer from all experts is selected based on an agreement score.

**Prompt**: "What are the benefits of renewable energy?"

Specialized Prompts:

Retrieval Augmentation Prompt: "What are the factual benefits of renewable energy?"

Chain-of-Thought Reasoning: "Explain the multi-step benefits of renewable energy."

Generated Knowledge Prompt: "What are the commonsense benefits of renewable energy?"

Final Output (ASSISTANT): "The benefits of renewable energy include reducing greenhouse gas emissions (factual), providing sustainable energy sources (multi-step), and being environmentally friendly (commonsense)."

## Self-Consistency

Self-Consistency (Wang et al., 2022) is based on the intuition that multiple different reasoning paths can lead to the same answer. This method first prompts the LLM multiple times to perform CoT, crucially with a non-zero temperature to elicit diverse reasoning paths. Next, it uses a majority vote over all generated responses to select a final response. Self-Consistency has shown improvements on arithmetic, commonsense, and symbolic reasoning tasks

**Prompt**: "What is the capital of France?"

**Multiple Reasoning Paths:**

Path 1: "The capital of France is Paris."

Path 2: "France's capital city is Paris."

Path 3: "Paris is the capital of France."

Final Output (ASSISTANT): "The capital of France is Paris."

## Universal Self-Consistency

Universal Self-Consistency (Chen et al., 2023e) is similar to Self-Consistency except that rather than selecting the majority response by programmatically counting how often it occurs, it inserts all outputs into a prompt template that selects the majority answer. This is helpful for free-form text generation and cases where the same answer may be output slightly differently by different prompts.

**Prompt**: "Explain the process of natural selection."

**Multiple Outputs:**

Output 1: "Natural selection is the process by which organisms better adapted to their environment tend to survive and produce more offspring."

Output 2: "Natural selection involves the survival and reproduction of organisms that are best suited to their environment."

Output 3: "The process of natural selection means that organisms with favorable traits are more likely to survive and reproduce."

Final Output (ASSISTANT): "Natural selection is the process by which organisms better adapted to their environment tend to survive and produce more offspring."

## USP (Universal Self-Adaptive Prompting)

Universal Self-Adaptive Prompting (USP) (Wan et al., 2023b) builds upon the success of COSP, aiming to make it generalizable to all tasks. USP makes use of unlabeled data to generate exemplars and a more complicated scoring function to select them. Additionally, USP does not use Self-Consistency

**Prompt**: "What are the benefits of a balanced diet?"

**Generated Exemplars:**

Exemplar 1: "A balanced diet provides essential nutrients that the body needs to function properly."

Exemplar 2: "It helps maintain a healthy weight and reduces the risk of chronic diseases."

Exemplar 3: "A balanced diet supports overall well-being and energy levels."

Final Output (ASSISTANT): "The benefits of a balanced diet include providing essential nutrients, maintaining a healthy weight, reducing the risk of chronic diseases, and supporting overall well-being and energy levels."

## Prompt Paraphrasing

Prompt Paraphrasing (Jiang et al., 2020) transforms an original prompt by changing some of the wording, while still maintaining the overall meaning. It is effectively a data augmentation technique that can be used to generate prompts for an ensemble.

**Original Prompt**: "What are the effects of global warming?"

**Paraphrased Prompts:**

"How does global warming impact the environment?"

"What consequences does global warming have on our planet?"

"In what ways does global warming affect the Earth?"

Final Output (ASSISTANT): "Global warming impacts the environment by causing rising temperatures, melting ice caps, and increasing sea levels."

# **Self-Criticism**

When creating GenAI systems, it can be useful to have LLMs criticize their own outputs (Huang et al.,2022). This could simply be a judgement (e.g., is this output correct) or the LLM could be prompted to provide feedback, which is then used to improve the answer. Many approaches to generating and integrating self-criticism have been developed.

## Chain-of-Verification (COVE)

Chain-of-Verification (COVE) (Dhuliawala et al., 2023) first uses an LLM to generate an answer to a given question. Then, it creates a list of related questions that would help verify the correctness of the answer. Each question is answered by the LLM, then all the information is given to the LLM to produce the final revised answer. This method has shown improvements in various question-answering and text-generation tasks

**Prompt**: "What are the main causes of climate change?"

Initial Answer: "The main causes of climate change are the burning of fossil fuels and deforestation."

**Verification Questions**:

"How does the burning of fossil fuels contribute to climate change?"

"What role does deforestation play in climate change?"

**Verification Answers**:

"The burning of fossil fuels releases greenhouse gases like CO2 into the atmosphere, which trap heat."

"Deforestation reduces the number of trees that can absorb CO2, increasing the amount of greenhouse gases."

Final Revised Answer (ASSISTANT): "The main causes of climate change are the burning of fossil fuels, which releases greenhouse gases like CO2 into the atmosphere, and deforestation, which reduces the number of trees that can absorb CO2."

## Self-Calibration

Self-Calibration (Kadavath et al., 2022) first prompts an LLM to answer a question. Then, it builds a new prompt that includes the question, the LLM’s answer, and an additional instruction asking whether the answer is correct. This can be useful for gauging confidence levels when applying LLMs when deciding when to accept or revise the original answer.

**Prompt:** "What is the capital of Japan?"

Initial Answer: "The capital of Japan is Tokyo."

**Self-Calibration Prompt**: "What is the capital of Japan? The capital of Japan is Tokyo. Is this answer correct?"

Final Output (ASSISTANT): "Yes, the capital of Japan is Tokyo."

## Self-Refine

Self-Refine (Madaan et al., 2023) is an iterative framework where, given an initial answer from the LLM, it prompts the same LLM to provide feedback on the answer, and then prompts the LLM to improve the answer based on the feedback. This iterative process continues until a stopping condition is met (e.g., max number of steps reached). Self-Refine has demonstrated improvement across a range of reasoning, coding, and generation tasks.

**Prompt:** "What are the benefits of regular exercise?"

Initial Answer: "Regular exercise improves cardiovascular health."

Feedback Prompt: "What are the benefits of regular exercise? Regular exercise improves cardiovascular health. How can this answer be improved?"

**Improved Answer**: "Regular exercise improves cardiovascular health, helps in weight management, and boosts mental health."

Final Output (ASSISTANT): "Regular exercise improves cardiovascular health, helps in weight management, and boosts mental health."

## Self-Verification

Self-Verification (Weng et al., 2022) generates multiple candidate solutions with Chain-of Thought (CoT). It then scores each solution by masking certain parts of the original question and asking an LLM to predict them based on the rest of the question and the generated solution. This method has shown improvement on eight reasoning datasets.

**Prompt:** "What are the main causes of climate change?"

**Multiple Candidate Solutions:**

Solution 1: "The main causes of climate change are the burning of fossil fuels and deforestation."

Solution 2: "Climate change is primarily caused by human activities such as burning fossil fuels and deforestation."

Solution 3: "Deforestation and the burning of fossil fuels are the primary drivers of climate change."

Scoring Process: Mask parts of the question and ask the LLM to predict them based on the solutions.

Final Output (ASSISTANT): "The main causes of climate change are the burning of fossil fuels and deforestation."

## ReverseCoT (Reversing Chain-of-Thought)

Reversing Chain-of-Thought (RCoT) (Xue et al., 2023) first prompts LLMs to reconstruct the problem based on generated answer. Then, it generates fine-grained comparisons between the original problem and the reconstructed problem as a way to check for any inconsistencies. These inconsistencies are then converted to feedback for the LLM to revise the generated answer

**Prompt:** "Describe the water cycle."

Generated Answer: "The water cycle involves evaporation, condensation, and precipitation."

Reconstructed Problem: "Based on the answer 'The water cycle involves evaporation, condensation, and precipitation,' what was the original question?"

Comparison: "Original question: 'Describe the water cycle.' Reconstructed question: 'What are the stages of the water cycle?'"

Feedback: "Minor inconsistency found. The original question asked for a description, not just the stages."

**Revised Answer**: "The water cycle involves evaporation, condensation, and precipitation. During evaporation, water turns into vapor. Condensation forms clouds, and precipitation returns water to the surface."

Final Output (ASSISTANT): "The water cycle involves evaporation, condensation, and precipitation. During evaporation, water turns into vapor. Condensation forms clouds, and precipitation returns water to the surface."

## Cumulative Reason

Cumulative Reasoning (Zhang et al., 2023b) first generates several potential steps in answering the question. It then has a LLM evaluate them, deciding to either accept or reject these steps. Finally,it checks whether it has arrived at the final answer.If so, it terminates the process, but otherwise it repeats it. This method has demonstrated improvements in logical inference tasks and mathematical problem.

Prompt: "Explain the process of cellular respiration."

**Generated** **Steps**:

Step 1: "Glycolysis occurs in the cytoplasm, breaking down glucose into pyruvate."

Step 2: "The Krebs cycle takes place in the mitochondria, producing ATP and electron carriers."

Step 3: "The electron transport chain uses electrons to generate a large amount of ATP."

Evaluation:

Step 1: "Accepted."

Step 2: "Accepted."

Step 3: "Accepted."

Final Output (ASSISTANT): "Cellular respiration involves glycolysis in the cytoplasm, the Krebs cycle in the mitochondria, and the electron transport chain, producing ATP and electron carriers."

# **Decomposition**

Significant research has focused on decomposing complex problems into simpler sub-questions. This is an effective problem-solving strategy for humans as well as GenAI (Patel et al., 2022). Some decomposition techniques are similar to thought-inducing techniques, such as CoT, which often naturally breaks down problems into simpler components. However, explicitly breaking down problems can further improve LLMs’ problem solving ability.

## DECOMP (Decomposed Prompting)

Decomposed Prompting (DECOMP) (Khot et al., 2022) Few-Shot prompts a LLM to show it how to use certain functions. These might include things like string splitting or internet searching; these are often implemented as separate LLM calls. Given this, the LLM breaks down its original problem into sub-problems which it sends to different functions. It has shown improved performance over Least-to-Most prompting on some tasks.

Web Search Task

**Complex Problem**: Gather information on a specific topic from the internet.

Decomposed Sub-Problems:

1 - Identify key search terms related to the topic.

2 - Conduct multiple web searches using these terms.

3 - Filter and compile relevant information from the search results.

4 - Synthesize the compiled information into a coherent summary.

## Faithful CoT

Faithful Chain-of-Thought (Lyu et al., 2023) generates a CoT that has both natural language and symbolic language (e.g. Python) reasoning, just like Program-of-Thoughts. However, it also makes use of different types of symbolic languages in a task-dependent fashion

Data Analysis Task

Complex Problem: Analyze a dataset to find trends and patterns.

Faithful Chain-of-Thought:

1 - Natural Language Reasoning: Describe the dataset and the goal of the analysis.

2 - Symbolic Language (Python): Write a Python script to clean and preprocess the data.

3 - Natural Language Reasoning: Interpret the cleaned data and decide on the analysis methods.

4 - Symbolic Language (Python): Implement statistical analysis and visualization in Python.

5 - Natural Language Reasoning: Summarize the findings and insights from the analysis.

## Least-to-Most

Least-to-Most Prompting (Zhou et al., 2022a) starts by prompting a LLM to break a given problem into sub-problems without solving them. Then, it solves them sequentially, appending model responses to the prompt each time, until it arrives at a final result. This method has shown significant improvements in tasks involving symbolic manipulation, compositional generalization, and mathematical reasoning.

Planning a Project

Complex Problem: Plan a large-scale project with multiple phases.

Least-to-Most Prompting:

1 - Decompose: Identify and list all the phases and tasks involved in the project.

2 - Solve Sub-Problems: Develop a detailed plan for each phase and task sequentially.

3 - Combine Solutions: Integrate the detailed plans to create a comprehensive project plan.

## Plan-and-Solve

Plan-and-Solve Prompting (Wang et al., 2023f) consists of an improved Zero-Shot CoT prompt,"Let’s first understand the problem and devise a plan to solve it. Then, let’s carry out the plan and solve the problem step by step". This method generates more robust reasoning processes than standard Zero-Shot-CoT on multiple reasoning datasets.

Solving a Logic Puzzle

Complex Problem: Solve a challenging logic puzzle.

Plan-and-Solve Prompting:

1 - Understand the Problem: Read and comprehend the rules and objective of the puzzle.

2 - Devise a Plan: Outline a strategy to approach the puzzle, such as identifying key clues and possible solutions.

3 - Execute the Plan: Follow the outlined strategy step by step to solve the puzzle.

## Program-of-Thought

Program-of-Thoughts (Chen et al., 2023d) uses LLMs like Codex to generate programming code as reasoning steps. A code interpreter executes these steps to obtain the final answer. It excels in mathematical and programming-related tasks but is less effective for semantic reasoning tasks.

Automating Data Analysis

Complex Problem: Analyze a large dataset to find correlations.

Program-of-Thoughts:

1 - Generate Code: Use Codex to write a Python script that performs data cleaning, preprocessing, and correlation analysis.

2 - Execute Code: Run the script using a code interpreter to process the data and find correlations.

3 - Final Answer: The script's output includes the identified correlations and relevant statistics.

## Recurs.-of-Thought

Recursion-of-Thought (Lee and Kim, 2023) is similar to regular CoT. However, every time it encounters a complicated problem in the middle of its reasoning chain, it sends this problem into another prompt/LLM call. After this is completed, the answer is inserted into the original prompt. In this way, it can recursively solve complex problems, including ones which might otherwise run over that maximum context length. This method has shown improvements on arithmetic and algorithmic tasks. Though implemented using fine-tuning to output a special token that sends sub-problem into another prompt, it could also be done only through prompting.

Solving a Complex Algorithmic Problem

Complex Problem: Optimize a complex algorithm for better performance.

Recursion-of-Thought:

1 - Initial Reasoning: Identify the parts of the algorithm that need optimization.

2 - Encounter Complicated Problem: When a particularly challenging part is identified, send it to another LLM call for detailed analysis.

3 - Insert Solution: Insert the solution from the secondary LLM call back into the original reasoning chain.

4 - Final Solution: Continue this process recursively until the entire algorithm is optimized.

## Skeleton-of-Thought

Skeleton-of-Thought (Ning et al., 2023) focuses on accelerating answer speed through parallelization. Given a problem, it prompts an LLM to create a skeleton of the answer, in a sense, sub-problems to be solved. Then, in parallel, it sends these questions to a LLM and concatenates all the outputs to get a final response.

Writing a Comprehensive Report

Complex Problem: Write a detailed report on a multifaceted topic.

Skeleton-of-Thought:

1 - Create Skeleton: Break down the report into sections such as introduction, literature review, methodology, results, and conclusion.

2 - Parallel Processing: Assign each section to a different LLM call for drafting.

3 - Concatenate Outputs: Merge the drafts from each section to compile the complete report.

## Tree-of-Thought

Tree-of-Thought (ToT) (Yao et al., 2023b), also known as Tree of Thoughts, (Long, 2023), creates a tree-like search problem by starting with an initial problem then generating multiple possible steps in the form of thoughts (as from a CoT). It evaluates the progress each step makes towards solving the problem (through prompting) and decides which steps to continue with, then keeps creating more thoughts. ToT is particularly effective for tasks that require search and planning

Project Management

Complex Problem: Plan and execute a large-scale project.

Tree-of-Thought:

1 - Initial Problem: Outline the project's goals and requirements.

2 - Generate Steps: Develop various approaches to achieve the project milestones.

3 - Evaluate Progress: Determine which approaches are most effective.

4 - Continue Steps: Elaborate on the most successful approaches.

5 - Final Solution: Finalize the project plan by continuously refining the best strategies.

## Metacognitive

Metacognitive Prompting (Wang and Zhao, 2024) attempts to make the LLM mirror human metacognitive processes with a five part prompt chain, with steps including clarifying the question, preliminary judgement, evaluation of response, decision confirmation, and confidence assessment.

Solving a Complex Problem

Complex Problem: Determine the best investment strategy.

Metacognitive Prompting:

1 - Clarify the Question: Define the investment goals and constraints.

2 - Preliminary Judgement: Identify potential investment options.

3 - Evaluation of Response: Analyze the risks and returns of each option.

4 - Decision Confirmation: Select the most suitable investment strategy.

5 - Confidence Assessment: Assess the confidence level in the chosen strategy.