Tipos de Reasoning Strategies

Reasoning Technique	Short Description
Chain-of-Thought (CoT)	Breaks down a complex problem into intermediate reasoning steps to reach a solution.
Tree-of-Thought (ToT)	Explores multiple reasoning paths in a tree structure to evaluate and select the best.
Self-Consistency	Samples multiple CoT outputs and selects the most consistent or frequent answer.
ReAct (Reason + Act)	Combines reasoning with action by letting the model think and interact with tools.
Reflexion	Uses feedback from previous answers to revise or improve reasoning in future steps.
Least-to-Most Prompting	Starts with easier subtasks and gradually solves harder ones using prior results.
Step-by-Step Prompting	Encourages the model to reason one step at a time before arriving at an answer.
Scratchpad	Allows the model to write and store intermediate computations or thoughts.
Active Prompting	Guides the model through a sequence of questions to stimulate deeper reasoning.

Generated Knowledge Prompting	Injects relevant background knowledge generated by the LLM before reasoning begins.
Answer-Then-Explain	Asks the model to give an answer first, then explain the reasoning behind it.
Explain-Then-Answer	Encourages reasoning first, followed by the final answer for more reliable results.
Multi-Prompt Decoding	Uses diverse prompts to generate multiple answers and aggregates them for consensus.
Verification and Critique	One model (or pass) critiques or verifies another's reasoning for reliability.
Tool-Augmented Reasoning	Integrates external tools (e.g., calculator, search engine) into reasoning workflows.
Deliberation Networks	Emulates a back-and-forth internal debate to refine reasoning and decisions.

Pesquisa dos reasonings dados em aula - Quais são e não são estratégias de reasoning

✅ Categorized Table

Reference	Type	Definition / Explanation
Fully Supervised Fine-Tuning	❌ *Training Method*	Adapts a pre-trained LLM to a task using labeled data. Not a reasoning strategy — it's part of model adaptation.
Prompt & In-Context Learning	✅ *Reasoning Strategy*	Uses task examples or instructions directly in the prompt to guide the model's reasoning and output.
Rationale Engineering – Chain-of-Thought	✅ *Reasoning Strategy*	Designs prompts to elicit step-by-step reasoning (Chain-of-Thought, or CoT), improving logical accuracy.
Self-Consistency Sampling	✅ *Reasoning Strategy*	Generates multiple reasoning paths (e.g., CoT) and selects the most frequent or consistent answer.
Decomposição de Problemas *(Problem Decomposition)*	✅ *Reasoning Strategy*	Breaks complex tasks into simpler subproblems to solve sequentially — a core principle in structured reasoning.
Tool-Augmented Reasoning	✅ *Reasoning Strategy*	Combines reasoning with external tools (e.g., calculator, web search) during inference to improve problem-solving.
Memory and Contextual Reasoning	❌ *Capability / Mechanism*	Refers to the model's use of past turns (memory) or large prompt context. Enables reasoning but is not itself a strategy.
MCP – Protocol	❌ *Framework / Methodology*	Likely refers to a structured **prompting or evaluation protocol**. Not a standard reasoning strategy — needs clarification, but likely a **meta-approach** or **framework** for task execution.

🔍 Summary

✅ Are Reasoning Strategies

- Prompt & In-Context Learning
- Rationale Engineering / Chain-of-Thought
- Self-Consistency Sampling
- Decomposição de problemas (Problem Decomposition)
- Tool-Augmented Reasoning

❌ Are Not Reasoning Strategies

- **Fully Supervised Fine-Tuning** → *Training Method*
- **Memory and Contextual Reasoning** → *Underlying Capability*
- **MCP – Protocol** → *Likely a Framework*, not a direct reasoning technique (needs precise definition)

📚 References Table

Concept	Type	Reference / Source	Notes
Fully Supervised Fine-Tuning	Training Method	Howard & Gugger (2020); Hugging Face Docs	Model training with labeled data; not used during inference.
Prompt & In-Context Learning	Reasoning Strategy	Brown et al. (2020), *GPT-3 Paper*	Few-shot learning by providing examples in the prompt.
Rationale Engineering / CoT	Reasoning Strategy	Wei et al. (2022), *Chain of Thought Prompting*	Improves reasoning by prompting step-by-step thought.

Self-Consistency Sampling	Reasoning Strategy	Wang et al. (2022), *Self-Consistency Improves Chain of Thought*	Samples multiple CoT paths and chooses the most consistent answer.
Decomposição de Problemas	Reasoning Strategy	Zhou et al. (2022), *Least-to-Most Prompting*; Cobbe et al. (2021)	Breaks down complex tasks into simpler subtasks.
Tool-Augmented Reasoning	Reasoning Strategy	Yao et al. (2022), *ReAct: Reason + Act*	Combines reasoning with external tools (e.g., calculator, search engine).
Memory & Contextual Reasoning	Model Capability / Mechanism	OpenAI (GPT-4 Memory); LangChain docs	Enables multi-turn understanding and long-context reasoning; not a reasoning strategy.
MCP – Protocol	Framework / Not Standardized	*No formal academic source found*	Possibly internal or domain-specific method; not recognized as a formal reasoning strategy.

1 - In-Context Learning (ICL) reasoning strategy

Creating an effective prompt for In-Context Learning (ICL) involves giving a language model examples of the task you want it to perform — directly within the prompt — without updating the model weights. This is commonly used in zero-shot, one-shot, or few-shot learning setups.

🧠 Principles of Effective In-Context Learning Prompts

Best Practice	Explanation
✅ Include task instructions	Give clear, concise instructions.
✅ Use structured examples	Include 1–5 high-quality input/output pairs in the format you want.
✅ Keep examples relevant	Use examples similar in type/format to your target input.
✅ Be consistent in formatting	Use the same phrasing, structure, and spacing in all examples.
❌ Avoid ambiguous examples	Don't include unclear or mixed-format samples — they confuse the model.

✨ Example: Sentiment Classification (Few-Shot Prompt)

Goal: Classify whether a review is *positive* or *negative*

text
```
Classify the sentiment of the following reviews as either Positive or
Negative.

Review: "I absolutely loved the new phone. Battery life is great!"
Sentiment: Positive

Review: "The food was cold and the service was terrible."
Sentiment: Negative
```

```
Review: "I'm really happy with my purchase. Everything works
perfectly."
Sentiment: Positive

Review: "The product broke after two days of use. I'm very
disappointed."
Sentiment:
```

🧠 **What happens: The model sees 3 examples and is expected to complete the 4th with `Negative`.**

⚙️ Prompt Variants for Different Tasks

Task Type	Prompt Structure
Text Classification	**Show labeled examples with clear "Input → Label" format.**
QA (Closed Book)	**Provide Q&A examples; end with a new question.**
Text Generation	**Provide story/completion snippets and let the model continue.**
Math Word Problems	**Use CoT-style examples with reasoning + answer.**

🛠️ Tips for Optimization

- Try more examples if the model is underperforming (up to 4–6 for GPT-3.5, more for GPT-4).
- Use Chain-of-Thought reasoning if the task requires step-by-step logic.
- For GPT-4 or similar, prefix your prompt with instructions (e.g., "You are a helpful assistant…").

Effective Prompt & In-Context Learning setup tailored for the fiscal domain — specifically, a tax deduction classification task

Example: Tax Deduction Classification (Few-Shot Prompt)

text

```
You are a helpful fiscal assistant. Classify each expense description
as either "Deductible" or "Non-Deductible" for tax purposes.

Expense: "Office supplies like pens and paper."
Classification: Deductible

Expense: "Client entertainment at a restaurant."
Classification: Deductible

Expense: "Personal clothing purchases."
Classification: Non-Deductible

Expense: "Travel expenses for business meetings."
Classification: Deductible

Expense: "Home electricity bill."
Classification: Non-Deductible

Expense: "Conference registration fees for employees."
Classification:
```

Result:

Classification: Deductible

Explanation

- **The prompt gives clear instructions.**
- **It shows examples relevant to fiscal domain.**
- **The format is consistent and easy to follow.**
- **The model will predict the classification for the last expense.**

Complex prompt example for the fiscal domain that combines explanations with classification — this helps the model reason about why an expense is deductible or not, improving accuracy and transparency

Example: Tax Deduction Classification with Explanation (Few-Shot Prompt)

text

```
You are a fiscal assistant. For each expense description, determine if
it is "Deductible" or "Non-Deductible" for tax purposes. Also, provide
a brief explanation supporting your classification.

Expense: "Office supplies like pens and paper."
Classification: Deductible
Explanation: These are necessary business expenses directly related to
office operations.

Expense: "Client entertainment at a restaurant."
Classification: Deductible
Explanation: Entertainment expenses for clients are deductible when
they are directly related to business activities.

Expense: "Personal clothing purchases."
Classification: Non-Deductible
Explanation: Personal clothing is a personal expense and not related
to business operations.

Expense: "Travel expenses for business meetings."
Classification: Deductible
Explanation: Travel costs for business purposes are deductible as they
are necessary for company operations.

Expense: "Home electricity bill."
Classification: Non-Deductible
Explanation: Home utilities are personal expenses unless a specific
business portion is documented.

Expense: "Conference registration fees for employees."
Classification:
Explanation:
```

Result :
`Classification`: **Deductible**
`Explanation`: **Conference registration fees are considered a business-related expense that supports employee training and professional development, making them deductible under tax regulations.**

How this helps:

- **The explanation forces the model to “think” and justify the classification.**
- **It improves transparency and can be used for audit or review.**
- **Encourages more accurate and confident predictions.**

Chain-of-Thought Prompt for Tax Deduction Classification

text

```
You are a fiscal assistant helping classify expenses for tax
deduction. For each expense, think through the reasoning step-by-step
before giving the final classification as "Deductible" or
"Non-Deductible."

Expense: "Office supplies like pens and paper."
Step 1: These supplies are used directly in the business.
Step 2: Business-related office supplies are usually deductible.
Final Classification: Deductible

Expense: "Client entertainment at a restaurant."
Step 1: Entertaining clients is related to business activities.
Step 2: Such entertainment expenses are generally deductible under tax
laws.
Final Classification: Deductible

Expense: "Personal clothing purchases."
Step 1: This expense is for personal items, not business use.
Step 2: Personal expenses are not deductible.
Final Classification: Non-Deductible

Expense: "Travel expenses for business meetings."
Step 1: Travel is necessary for attending business meetings.
Step 2: Business travel costs are deductible.
Final Classification: Deductible

Expense: "Home electricity bill."
Step 1: Electricity is a utility expense for the home.
Step 2: Unless a portion is explicitly for business, it is personal.
Final Classification: Non-Deductible

Expense: "Conference registration fees for employees."
Step 1: This is a business-related professional development expense.
Step 2: Such fees are deductible as they support work.
Final Classification:
```

Result :

```
Final Classification: Deductible
```

Benefits of this approach:

- **The model explicitly *walks through* the reasoning.**
- **Helps handle complex or borderline cases.**
- **Great for auditability and explanations.**

2 - Self-Consistency Sampling reasoning strategy

Creating an **effective Self-Consistency Sampling** reasoning strategy involves using **Chain-of-Thought (CoT) prompting** combined with **multiple diverse outputs** for the same input. You then **aggregate the final answers** to choose the most consistent or likely correct outcome.

🧠 What is Self-Consistency?

Self-Consistency is a reasoning strategy where a language model:

1. Is **prompted with a Chain-of-Thought** format (i.e., explains step-by-step reasoning),
2. Is **sampled multiple times** (e.g., using temperature ≈ 0.7–1.0),
3. Then **selects the most frequent or consistent answer** from those multiple generations.

🔍 It was introduced by [Wang et al., 2022] to improve reasoning on complex tasks like math and logic.

🛠️ Steps to Apply Self-Consistency

Step	What You Do
1. ✅ Use CoT prompting	Ask the model to "think step-by-step."
2. 🔁 Sample multiple outputs	Use **temperature > 0** and generate several completions (e.g., 5–20).
3. 📊 Aggregate results	Choose the **most common final answer** across all completions.

📘 Example: Arithmetic Reasoning

Prompt (Chain-of-Thought style):

```text
Q: If there are 3 cars and each car has 4 wheels, how many wheels are there in total?
A: Let's think step-by-step.
```

Sample Outputs:

Output 1:

```
Each car has 4 wheels. There are 3 cars.
So, 3 × 4 = 12 wheels.
Answer: 12
```

Output 2:

```
There are 3 cars. Each has 4 wheels.
3 times 4 is 12.
Answer: 12
```

Output 3:

```
Each car has 4 wheels, but including a spare tire, that's 5 per car.
5 × 3 = 15.
Answer: 15
```

Output 4:

```makefile
3 cars × 4 wheels = 12.
Answer: 12
```

Output 5:

```
Thinking step-by-step: 3 × 4 = 12.
Answer: 12
```

✅ Final Self-Consistent Answer:

From 5 samples:

- **12 appears 4 times**,
- **15 appears once**.

> **Final Answer: 12** (by majority vote = self-consistent outcome).

When to Use Self-Consistency?

- Complex **math or logic** problems
- Reasoning over **multi-step decisions**
- Cases where **single-shot CoT is unreliable**

Great! Let's apply **Self-Consistency Sampling** to the **fiscal domain**, specifically for reasoning about **whether an expense is tax-deductible**, using Chain-of-Thought (CoT) style reasoning + multiple sampled outputs.

🧠 Objective: Tax Deduction Classification with Self-Consistency Sampling

We'll:

1. Prompt the model to **think step-by-step**.
2. Generate **multiple completions** (using `temperature > 0`).
3. **Aggregate the answers** and select the most common as the final classification.

📝 Example Prompt (CoT Style in Fiscal Domain)

text
CopiarEditar

```
You are a fiscal assistant helping determine whether a business
expense is tax-deductible. Think through the classification
step-by-step and then give the final answer as "Deductible" or
"Non-Deductible".

Expense: "Monthly coworking space membership used to meet clients and
work remotely."
A: Let's think step-by-step.
```

🔁 Sample Outputs from Multiple Runs (via LLM):

Output 1

rust
CopiarEditar

```
The coworking space is used for business operations and client
meetings.
Business-use office spaces are considered deductible.
Final Answer: Deductible
```

Output 2

sql
CopiarEditar
```sql
The coworking membership supports business work and client
interactions.
Such spaces fall under office expenses, which are deductible.
Final Answer: Deductible
```

Output 3

rust
CopiarEditar
```rust
If the space is used primarily for business, then it's deductible.
However, if also used for personal purposes, it may be partially
deductible.
Final Answer: Deductible
```

Output 4

vbnet
CopiarEditar
```vbnet
Since the coworking space is used to meet clients and conduct business
activities, it qualifies as a business expense.
Final Answer: Deductible
```

Output 5

vbnet
CopiarEditar
```vbnet
Because the space is rented for work and client interaction, it's
aligned with deductible business costs.
Final Answer: Deductible
```

✅ Self-Consistent Result:

All 5 responses → **"Deductible"**
→ **Final classification:** `Deductible`

Use Cases in Fiscal Domain

Use Case	Why Self-Consistency Helps
Grey areas (e.g., home office expenses)	Helps surface the dominant interpretation
Deduction eligibility with nuanced criteria	Allows reasoning diversity before consensus
Audit preparation (documenting logic)	Outputs rationales + agreement improves traceability

3 - Rationale Engineering - Chain of Thought reasoning strategy

Rationale Engineering using the **Chain-of-Thought (CoT)** strategy is one of the most effective techniques for guiding large language models (LLMs) through **complex reasoning tasks**.

🧠 What Is Rationale Engineering (Chain-of-Thought)?

Rationale Engineering is the practice of designing **intermediate reasoning steps (rationales)** that help LLMs arrive at more accurate, robust, or explainable answers.

Chain-of-Thought (CoT) is a specific type of rationale engineering that:

- Instructs the model to **think step-by-step**,
- Mimics how humans break down problems,
- Helps handle **multi-step logical, mathematical, or classification** tasks.

✅ When to Use It?

Use CoT when:

- The task involves **multi-step reasoning**.
- Answers depend on **conditions or rules**.
- You want **explainability or traceability** in the model's reasoning.

🎯 Steps to Create Effective Rationale Engineering (CoT)

Step	Description
1. 📝 Choose a task that benefits from reasoning	e.g., tax deduction, legal classification, fraud detection

2. 🧩 Break it into **logical steps**	Define how a human would reason to the answer
3. 🧠 Craft the prompt to **think step-by-step**	Include "Let's think step-by-step" or similar
4. 🧪 Use **examples** (few-shot or zero-shot)	To show the model *how* to reason
5. 🔁 Test with **edge cases**	See if the rationale generalizes

---

📘 Example: Fiscal Domain (Tax Deduction – Chain-of-Thought Prompt)

Task: Classify whether an expense is deductible or not and explain the rationale.

text
CopiarEditar

```
You are a fiscal assistant. For each expense, think step-by-step and
determine if it is tax-deductible. Explain the reasoning and give the
final classification.

Expense: "Lunch with a potential client to discuss a business
proposal."

Step 1: The lunch was with a potential client, indicating a business
purpose.
Step 2: Business meals related to client discussions are generally
deductible under tax rules.
Step 3: As long as the meal was not lavish and documentation exists,
it qualifies.
Final Classification: Deductible
```

💡 Design Tips for Better CoT Rationale Engineering

Technique	How It Helps

Use consistent formatting (`Step 1:`, `Step 2:`)	Makes parsing and logic clearer
Include both positive and negative examples	Helps the model generalize reasoning
Add uncertainty conditions (e.g., “unless...”)	Encourages deeper reasoning
Use domain-specific terms (e.g., “Section 162”)	Anchors logic to real-world rules

---

🔍 Comparison: Without vs. With CoT

Without CoT

text
CopiarEditar
```
Expense: Lunch with a client
Classification: Deductible
```

With CoT (Rationale Engineering)

text
CopiarEditar
```
Step 1: The lunch is with a client, so it's related to business.
Step 2: The IRS allows deductions for meals related to business
discussions.
Step 3: As long as it is not lavish and properly documented, it
qualifies.
Final Classification: Deductible
```

> 🧠 Outcome: With CoT, the model **performs better** on ambiguous or complex inputs.

4 - Problem Decomposition reasoning strategy

Problem Decomposition is a powerful reasoning strategy for large language models (LLMs), especially when tackling complex tasks. It involves **breaking a large, difficult problem into smaller, manageable subproblems**, solving each one, and combining the results for a final answer.

🧠 What Is Problem Decomposition?

> **Problem Decomposition** is a reasoning technique where a task is split into smaller subtasks that are easier to solve individually.

This mirrors how humans approach complexity: divide → conquer → combine.

✅ When to Use Problem Decomposition

Use it when:

- The original task is **too broad or ambiguous**.
- Multiple **decision steps or types of knowledge** are involved.
- You want to **improve accuracy** by isolating logic steps.

🧰 Steps to Create an Effective Problem Decomposition Strategy

Step	Description
1. 🔍 Identify the main problem	Define the overall question or task
2. 🧩 Break it into subproblems	Divide it into simpler questions or logic chunks

3. 💡 Solve each subproblem independently	Each should be clearly answerable
4. 🔗 Combine sub-results	Synthesize final conclusion based on prior steps
5. ✅ Justify the final answer	Optional but helps with explainability

---

📘 Example: Fiscal Domain – Expense Deductibility (Problem Decomposition)

Task: Determine if a home internet expense is deductible for tax purposes.

Prompt Using Problem Decomposition:

text
CopiarEditar

```
You are a fiscal assistant. For each expense, break down the problem
into sub-questions to decide if it's tax-deductible.

Expense: "Home internet service used partially for remote work."

Step 1: Is the expense related to business activity?
→ Yes, part of the internet usage is for remote work.

Step 2: Is the expense mixed (personal and business)?
→ Yes, it is used both for personal and business purposes.

Step 3: Is there a method to separate the business portion?
→ If a reasonable method (e.g., percentage based on time or data) is
used, the business portion can be isolated.

Step 4: Are mixed-use expenses partially deductible?
→ Yes, when documented properly, the business-use portion is
deductible.

✅ Final Classification: Partially Deductible (business portion only)
```

🧠 Why This Works

Problem Decomposition helps the model:

- **Avoid oversimplified yes/no answers**.
- Handle **gray areas**, like mixed-use expenses.
- Explain **reasoning transparently**, ideal for audits or legal justifications.

5 - Tool-Augmented reasoning strategy

Tool-Augmented Reasoning is a powerful strategy where a Large Language Model (LLM) uses external tools—such as calculators, search engines, APIs, databases, or even other models—to enhance its reasoning capabilities.

🧠 What Is Tool-Augmented Reasoning?

> **Tool-Augmented Reasoning is the process of delegating subtasks (like calculations, lookups, or verifications) to external tools, while the LLM handles coordination, logic, and synthesis.**

The LLM acts as a controller that uses tools to extend its abilities beyond text generation alone.

✅ When to Use It

Use this strategy when:

- **The task requires real-time data (e.g., currency rates, tax thresholds).**
- **You need high accuracy (e.g., math, financial computation).**
- **You want to automate workflows (e.g., document classification, data extraction).**
- **The model alone may hallucinate or be unsure.**

🧰 Steps to Create an Effective Tool-Augmented Reasoning Strategy

Step	Description
1. 🎯 Define the problem	What is the user trying to solve?

2. 🧩 Identify which parts need tools	E.g., calculation, database query, search
3. 🔧 Choose the tools	Calculator, tax API, web search, spreadsheet, etc.
4. 🧠 Let LLM orchestrate	Use the model to break down the problem, call tools, and reason
5. 📌 Synthesize result	Combine tool results with reasoning to form a final answer

---

📘 Example: Fiscal Domain – Deduction Calculation (Tool-Augmented)

💼 Task:

Calculate the deductible portion of a home internet bill used partially for business purposes.

🧠 LLM Reasoning Flow:

text

CopiarEditar

```
Task: "Is the internet bill of $80/month deductible if 40% is used for business?"

Step 1: Determine the percentage used for business.
→ 40%

Step 2 (tool call): Calculate 40% of $80.
→ Use calculator tool → $32

Step 3: Verify if partial internet use qualifies for deduction.
→ Yes, based on IRS rules, mixed-use expenses are deductible for the business portion.

✅ Final Answer: $32 is deductible per month.
```

🛠️ Tool Used:

- A calculator to compute `$80 × 0.4 = $32`.

In practice, this could be handled by:

- A Python or Excel formula.
- A built-in calculator tool in an LLM agent system.
- A call to a fiscal rule API or database.

🧪 Example (Pseudocode in Python)

python
CopiarEditar

```
internet_cost = 80
business_percent = 40

deductible = internet_cost * (business_percent / 100)
print(f"Deductible portion: ${deductible:.2f}")
```

LLM might generate the reasoning and then delegate the computation to code like the above.

🔄 Tool-Augmented Reasoning in Systems (Examples)

Tool	Purpose
Calculator API	Financial math / percentages
Tax Rule API	Validate eligibility / limits

Web Search	Check current deduction limits
SQL/Database	Lookup expense records
Spreadsheet	Visualize or audit breakdowns

🧠 Why This Strategy Matters

Without tools, LLMs:

- **May hallucinate tax rates or make math errors.**
- **Struggle with up-to-date or structured data.**

With tools:

- **Outputs are more reliable and precise.**
- **LLMs behave like fiscal analysts, not just writers.**

🔚 Summary

Component	Role
LLM	Reasoning, decomposition, orchestration
External Tool (e.g., calculator)	Performs the accurate subtask (math, lookup)
Final Output	Combines tool results + explanation