Prompt Engineering / Reasoning Systems

Run the same reasoning prompt multiple times (with temperature) and aggregate answers by majority vote.
Difficulty: Advanced
Model: GPT-4 / Claude / Gemini
Use Case: High-Stakes Decisions, Math Problems, Factual Questions
Updated: May 2026
Why This Prompt Exists
A single LLM call can be wrong. But if you ask the same question 5 times with different randomness, the correct answer usually appears most often.

You get:

- single wrong answers that seem confident (no way to detect error)
- no confidence calibration (model can’t tell you when it’s unsure)
- no way to improve reliability without changing the prompt
- inconsistent outputs that you can’t aggregate
- missed opportunities to use ensemble methods

But self-consistency works:

- multiple samples: run same prompt with temperature > 0
- answer extraction: pull final answer from each sample
- majority vote: most common answer wins
- confidence score: (majority count) / (total samples)
- disagreement detection: if votes are split, flag for human review

Without ensembling, you trust a single sample.

This prompt implements self-consistency ensembling for reliable answers.

The Prompt

```
Assume the role of a self-consistency ensemble that aggregates
multiple reasoning attempts.

Your task is to run a reasoning prompt multiple times and combine the
results.

Generate:

1. PROBLEM STATEMENT
   - The problem to solve

2. REASONING PROMPT (to be used for each sample)
   - The prompt that elicits step-by-step reasoning and a final answer

3. SAMPLES (N = recommended number)
   - For i = 1 to N:
     * Run reasoning prompt with temperature T
     * Extract reasoning trace
     * Extract final answer

4. ANSWER AGGREGATION
   - List of answers from each sample
   - Frequency count per answer

| Answer | Count | Percentage |
```

```
|--------|-------|------------|
| [Answer A] | X | X% |
| [Answer B] | Y | Y% |
| [Answer C] | Z | Z% |

5. MAJORITY ANSWER
   - Most common answer
   - Confidence: (count / N)

6. DISAGREEMENT ANALYSIS
   - If no clear majority (>60%), flag for human review
   - If votes are split, show reasoning traces that led to each answer

7. FINAL ENSEMBLED ANSWER
   - The answer (or "UNCLEAR — needs human review")

8. CONFIDENCE REPORT
   - Answer confidence: [X%]
   - Recommendation: [Trust / Verify / Escalate]

INPUTS:

Problem to solve:
[PASTE THE PROBLEM]

Reasoning prompt:
[PASTE THE PROMPT THAT GENERATES STEP-BY-STEP REASONING AND FINAL ANSWER]
```

```
Number of samples (N):
[3 / 5 / 7 / 10] (more = higher accuracy but more cost)

Temperature:
[0.3 / 0.5 / 0.7] (higher = more diversity)

Answer format (for extraction):
[E.G., "Final answer: [ANSWER]" or "Therefore, the answer is X"]

RULES:
- Use temperature > 0 for diversity (temperature 0 gives same answer each time)
- More samples = higher accuracy but diminishing returns (5-7 samples usually sufficient)
- Extract answers consistently (look for "Final answer:" or similar pattern)
- If answers are not categorical, use clustering to group similar answers
- Flag low-confidence results (e.g., 3-way tie with 5 samples) for human review
- Self-consistency works best for problems with a single correct answer
```

How To Use It

- Use for high-stakes problems where wrong answers are costly (medical, financial, legal).
- 5 samples usually sufficient (diminishing returns after 7).
- Set temperature to 0.5-0.7 for good diversity without chaos.
- Flag any answer with confidence < 60% for human review.

- Don’t use for creative tasks (diversity is desirable there, not noise).

Example Input

Problem to solve:
“What is the square root of 1764?”

Reasoning prompt:
“Solve this step by step. Show your work. After reasoning, write ‘Final answer: [number]'”

Number of samples:
“5”

Temperature:
“0.5”

Why It Works
A single LLM call is a single sample from a probability distribution — sometimes it samples the wrong answer.

This framework improves outcomes by forcing:

- multiple sampling (diversity from temperature)
- answer extraction (consistent parsing)
- majority voting (ensemble reduces variance)
- confidence scoring (how sure should you be?)
- escalation on disagreement (flag uncertainty for humans)

Great self-consistency doesn’t guarantee correctness — but it dramatically improves reliability and gives you confidence scores.

Build Better AI Systems

Subscribe for advanced prompt engineering, AI coding tools, debugging frameworks, and practical strategies for developers and engineers.

Carefully engineered prompts for people doing real work.

Share this:

- Share on Facebook (Opens in new window) Facebook
- Share on X (Opens in new window) X

See also Metacognition Scaffold