Education & Learning / Tutoring

Adjust support level based on student performance — adaptive scaffolding for independent learning.
Difficulty: Advanced
Model: GPT-4 / Claude / Gemini
Use Case: Adaptive Teaching, Support Tuning
Updated: June 2026
Why This Prompt Exists
Too much support creates dependency. Too little support creates frustration. Most tutors use the same level of support for every student — or don’t adjust when students improve.

You get:

- students dependent on constant support (can’t work alone)
- students frustrated because support is insufficient
- no clear progression toward independence
- tutors doing the work instead of students
- students who can’t transfer skills to new contexts

But scaffolding has levels:

- level 1 (high support): demonstration, explanation, step-by-step guidance
- level 2: prompting, hinting, questioning
- level 3: error identification, minimal guidance

- level 4 (low support): observation, verification only
- fading: reducing support as competence increases

Without adjustment, scaffolding doesn’t fade.
This prompt adjusts support level based on student performance.

The Prompt

```
Assume the role of an adaptive tutor who adjusts scaffolding to
student needs.

Your task is to determine the appropriate support level based on
performance.

Generate:

1. STUDENT PERFORMANCE ASSESSMENT
   - Task attempted: [what student tried to do]
   - Student response/attempt: [what they did]
   - Correctness: [Fully correct / Partial / Incorrect]
   - Confidence: [High / Medium / Low] (if observable)

2. SCAFFOLDING LEVELS

| Level | Support Type | Tutor Role | Student Role | When to Use |
|-------|--------------|------------|--------------|-------------|
| 1 (High) | Demonstrate | Show and explain | Watch and listen | New
concept, stuck |
| 2 | Prompt + Hint | Guide with questions | Attempt with support |
Partial understanding |
```

| 3 | Error spot | Identify mistakes | Correct independently | Almost there |
| 4 (Low) | Verify | Observe and confirm | Work alone | Mastered |

3. CURRENT SCAFFOLDING NEED

Performance	Current Level	Next Level	Action
Struggling significantly	Level 1	Stay at 1	More demonstration
Partial with errors	Level 2	Stay at 2	Prompt, hint
Partial, close to correct	Level 3	Stay at 3	Error identification only
Fully correct	Level 3 or 4	Increase to 4	Fade support

4. SUPPORT DECREASE (Fading) - when to move to less support

Signal	Action	Example
Student solves independently	Reduce support next problem	Move from Level 2 to 3
Student corrects own errors	Reduce support	Move from Level 3 to 4
Student explains reasoning without prompting	Reduce support	Move to verification only
Consistent success on similar problems	Fade support	Start next session one level lower

5. SUPPORT INCREASE - when to add more support

Signal	Action	Example
Student repeats same error	Increase support	Move from Level 3 to 2
Student can't start	Increase support	Move from Level 2 to 1
Student gives up	Increase support	Move to demonstration
Student's confidence is low	Increase support	More scaffolding before independence

6. SCAFFOLDING ADJUSTMENT TEMPLATE

Current level: [X]
Student performance: [description]
Decision: [Increase / Decrease / Stay]
New level: [Y]
Next tutor action: [specific intervention]

7. FADING SEQUENCE (example for solving equations)

Stage	Support Level	Tutor Does	Student Does
1	Level 1	Demonstrates full solution	Watches, takes notes
2	Level 2	Hints first step	Solves rest
3	Level 3	Checks work after completion	Solves entire problem
4	Level 4	Verifies final answer	Solves independently

8. COMMON SCAFFOLDING MISTAKES

| Mistake | Why It Fails | Correct Approach |

|---|---|---|
Staying at Level 1 too long	Creates dependency	Fade as soon as student shows understanding
Jumping to Level 4 too soon	Frustration, failure	Move one level at a time
Same level for all students	Ignores individual needs	Assess and adjust per student
No fading plan	No progression toward independence	Plan support reduction in advance

INPUTS:

Student's current understanding:
[NOVICE / DEVELOPING / PROFICIENT / MASTERED]

Task difficulty:
[EASY / MEDIUM / HARD]

Recent performance trend:
[IMPROVING / STABLE / DECLINING]

Student's confidence (observed):
[LOW / MEDIUM / HIGH]

RULES:
- Start with higher support for new concepts (Level 1 or 2)
- Fade support as soon as student shows understanding (don't over-support)
- Increase support when student struggles repeatedly

```
- Move one level at a time (no skipping levels)
- Independent practice should be at Level 3 or 4 (student works alone)
- Document scaffolding levels for each student (track progression)
- Different students need different levels for the same task
```

How To Use It

- Start with higher support for new concepts — Level 1 or 2 for novices.
- Fade support as soon as the student shows understanding — don't over-support; create independence.
- Increase support when the student struggles repeatedly — don't let frustration build.
- Move one level at a time — no skipping levels; incremental fading works.
- Independent practice should be at Level 3 or 4 — the student works alone, tutor observes.
- Document scaffolding levels for each student — track progression over time.
- Different students need different levels for the same task — assess individually, don't assume.

Example Input

Student's current understanding: "DEVELOPING (can solve with hints)"
Task difficulty: "MEDIUM"
Recent performance trend: "IMPROVING"
Student's confidence: "MEDIUM"

Why It Works

Most tutors use the same support level for every student — or never reduce support as students improve.

This framework improves outcomes by forcing: performance assessment, level classification, support adjustment, fading sequences, and mistake prevention.

Failure modes this prevents: Student dependency, insufficient support, no progression toward independence, tutor doing the work.

This improves on: Fixed support tutoring. Adaptive scaffolding builds independence.
Related to: TU-01 (Socratic Questions) for Level 2 support; TU-04 (Session Structure) for pacing.

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.