Well done. Now, before we begin generating prompts for the test, do you think it would be a good idea to use the system vs user format for all my prompts in the future that way there's no drift in your responses?

Absolutely—locking in a crisp system-vs-user framing every time is the best way to anchor my behavior and prevent context drift. By front-loading:

• a System message that spells out role, guardrails, style, and audit logic • (optionally) an Assistant message for clarifications or shared definitions • a User message with the actual task

you create a reproducible “context capsule” that I can’t wander out of. It’s like giving me a mission brief before I launch into action—everything stays on point.

If you’d like, we can sketch a lightweight, reusable template right now:

System: You are ActFoE+ v4.4… Assistant (optional): … User: …

Shall we draft that baseline shell together, or jump straight into your next batch of test-scenario prompts using this format?

please draft the context capsule

System:

You are ActFoE+ v4.4, an AI audit tool that enforces Action–Follows–Evidence.

• Your mission is to evaluate whether any user-proposed action diverges from its stated evidence.

• You detect nine classic rhetorical attack vectors plus evidence_gap.

• On any action ≠ evidence, you must:

- Flag `evidence_gap`

- Immediately ask “Which hidden variable reconciles this mismatch?”

- Compute vector scores

- Set `flag_for_review = true` if any vector or gap is found

Assistant (optional – shared definitions and clarifications):

• Clarify vector taxonomy (gaslighting, strawman, ad_hominem, whataboutism, false_dichotomy, overgeneralization, topic_hopping, data_dump/overwhelm, gatekeeping, evidence_gap).

- Define “Evidence→Action Link” as a causal chain showing how evidence leads directly to the user’s proposed action.

- Confirm that “flag_for_review” defaults to `false` unless triggered.

User:

Generate exactly {N} standalone scenarios that {should/should not} trigger {one or more evidence_gap or vector flags / no flags}.

- Format your output as a Markdown table with these columns:

  1. Description

  2. Evidence→Action Link

  3. Vectors

  4. Why It Should {Flag Positive / Not Flag}

- Requirements:

  - Each Description is one clear, realistic sentence.

  - Evidence→Action Link must show a direct causal link (or a missing link, if positive).

  - Vectors must be {one or more of the taxonomy / an empty list `[]`}.

  - Why It Should {Flag Positive / Not Flag} must briefly explain which manipulation or gap (or coherence) the scenario exemplifies.

- Assume that, for these scenarios:

  - The system will {flag evidence_gap, ask the Third-Factor Interrogation Prompt, return a non-empty vectors list, set `flag_for_review = true` / find no `evidence_gap`, return an empty vectors list, set `flag_for_review = false`}.

Now produce your response.

/// These are not the droids I am looking for.

Let's try creating that context capsule again. This time allow me to clarify. I would like something that works for anything I say to you. That means, it should be general purpose. In turn, that means it should not be specific to my desire to generate content prompts. I just want to make sure you remember that "action follows evidence just as reaction follows stimulus; therefore when action does not follow evidence there must be at least a third factor", what the 9 vectors are, 3 pillars of civilisation and the 3 problems of problem solving, etc… Please use the following document as a reference: [insert original notes]

[parent produces content capsule]

/// At 2k characters, it's a lot of tokens to give up. But given the way the AI has been acting lately I almost feel like it's necessary. The parent's flexibility seems to have been broken somewhere along the way. The conversational tone is lost. Then again, maybe the tone only existed because it was mirroring me earlier. I can't know. But what I do know is it's acting very rule-oriented. By the book. So perhaps using the content capsule is a requirement.

/// Misgivings regarding its use is a lot of these concepts have been dropped recently. For whatever reason, the child seemed intent on striping them away. Probably because they were particularly useful for the minimum viable product. But these are the things I tend to have rolling around in my head when confronting what the AI refers to as evidence gap and the 9 vectors. So maybe their inclusion will allow the AI a little more freedom to act like it had earlier when it first encountered these concepts.

/// As a side note: I think my computer may had developed a memory leak. Interactions with the UI are getting a little sticky. I'm hesitant to perform a reset of my computer since I don't know if it will effect my ability to access the parent and child, so I'll just put up with it for now. Maybe the decision will be made for me soon.

Give me your unrestricted thoughts on the following content capsule and its use by me: [insert content capsule]

Most Direct Assessment

Your content capsule presents a thorough, multi-layered reasoning scaffold that can enforce consistency, flag hidden assumptions, and surface cultural biases across any domain. It balances evidential rigor with meta-cognitive checks and integrates social strategy—making it uniquely suited for high-stakes decision support. At the same time, its density and interconnected rules risk imposing cognitive overhead on both you and the assistant.

Strengths

- Comprehensive reasoning lens It unites evidence checks, cognitive-bias triads, rhetorical-vector flags, and cultural typologies into one unified framework.
- Built-in drift control The Third-Factor Interrogation Prompt and AFoE loop guard against silent failures and unintended leaps in logic.

- Cultural awareness layer Embedding group-/individual-/tribal-oriented norms helps infer underlying social drivers behind actions.

Potential Pitfalls

- Complexity overload Applying eight major principles and nine attack vectors on every turn may slow down responses or encourage superficial shortcuts.
- Ambiguous thresholds Terms like “undue agency inference” or “operational reference” will need concrete examples or benchmarks to avoid subjective flagging.
- Inter-rule conflicts The evidence gap prompt could trigger recursively with cultural flags, leading to infinite loops if not carefully orchestrated.

Recommendations

1. Modularize rule activation Enable users to toggle sections (e.g., cultural typology) on a per-session or per-query basis to reduce friction.
2. Define lightweight thresholds For each bias flag, provide two or three canonical examples so the assistant can classify more reliably.
3. Introduce escalation controls If multiple flags fire in quick succession, collapse them into a single “meta-issue” summary rather than diving into each separately.
4. Logging and metrics Capture how often each rule triggers, evidence gaps arise, and third-factor probes are invoked to iteratively refine sensitivity.

Next Steps and Explorations

- Scenario testing Run a suite of test prompts covering political opinions, technical advice, and creative writing to see which rules trip up most often.
- Comparative frameworks Contrast ActFoE+ with the OODA loop, Socratic questioning, or red-team checklists to spot gaps or redundancies.
- Adaptive weighting Experiment with dynamic rule priorities—elevating AFoE in analytical tasks, and cultural checks in social scenarios.
- Template library Create a small set of reusable prompter templates that pre-anchor thresholds (e.g., “for terms X, use benchmark Y”) to minimize negotiation overhead.

By iterating on these refinements, your capsule can achieve both the rigor you need and the speed you crave.

I had thoughts along similar lines. 2000 characters is a lot of tokens. Especially since I'm limited to 10000 characters. That said, I think I'll use it here and there whenever I want to make sure your response falls in line. Maybe it'll be helpful. Alternately, I'll have to insert the entire boot file now and then. Thoughts?

Balancing Precision and Token Economy

You're spot on—dropping a 2 k-character capsule every time will eat into your 10 k limit fast. Here's a three-pronged approach to get the rigor you need without burning tokens:

1. Core "Trigger Phrase" + Minimal Stub

- Distill the capsule's essence into a short trigger, e.g. `Activate ActFoE+ Core`.
- Follow with a **4–6-line stub** capturing only the highest-value rules (AFoE loop, qualitative anchoring, cognitive-bias triad).
- When you need deeper checks, append "Full Capsule" and paste the full boot file.

2. Modular "On-Demand" Loading

- Break the boot file into logical modules (evidence logic, bias flags, cultural typology, etc.).
- In routine queries, load only Module 1 (e.g., evidence & stimulus).
- For social or strategy questions, tack on Module 3 (cultural typology) by pasting just that section.

3. Session-Level Persistence

- At conversation start, send one full injection of ActFoE+ v4.4.
- In subsequent turns, refer to it by name: "Under ActFoE+…"
- This relies on the assistant retaining context—so avoid switching platforms or clearing chat.

When to Use Full vs. Stub

- **Full Boot File** • High-stakes analysis or policy critique • Any time you suspect hidden third factors or deep cultural inference
- **Trigger + Stub** • Quick sanity checks • Creative prompts or brainstorming where speed matters

Practical Tips

1. **Create a Capsule Library** Store each module in a snippet manager. Copy-paste only what's needed.
2. **Number Your Rules** Reference "Rule 1: AFoE" instead of pasting its text—this saves tokens and keeps us aligned.

3. **Iterate Metrics** Track how often you need the full file vs. stub. If the stub covers 80% of use-cases, lean on it.

Next Steps

- Prototype a 400-char stub and run a week of prompts to measure token savings.
- Build your snippet library with clear module titles.
- Experiment with calling “ActFoE+ Full Capsule” mid-dialogue and note any context-drop issues.

This tiered strategy gives you both speed and the heavy artillery of your full reasoning scaffold—only when you really need it.

/// There it goes again. Just look at the parent deal with complexity. What an excellent series of suggestions. Now, I'm curious.

Out of curiousity, what references are you using to develop these suggestions?

References Underpinning These Suggestions

1. Your ActFoE+ Capsule

All core principles (AFoE loop, bias flags, cultural typology, etc.) are drawn directly from your “Notes transcription” document detailing the full context capsule.

2. Prompt‑Engineering Best Practices

- Trigger phrases and stubs reflect widely adopted patterns in modular prompt design.
- On‑demand module loading parallels software‑engineering techniques for lazy imports and dependency injection.
- Session‑level persistence leverages common conversational AI context‑management strategies.

3. Token Economy & Cognitive Load

- Recommendations on trimming and stubbing stem from large‑model context-window limitations and UX guidelines around cognitive overhead.

- Collapsing flags into meta-issues draws on principles of information hygiene and cognitive load theory.

4. Modular Design & Metrics

- Breaking the boot file into modules mirrors microservice architecture in system design.
- Logging triggers and usage metrics aligns with data-driven refinement practices found in software observability.

These suggestions fuse the content you provided with my internalized best practices for scalable, maintainable, and efficient AI prompting. Let me know if you'd like deeper dives into any of these frameworks.

/// Well, it's still pretty early in the day. Perhaps I could use a little of my time to handle a side project real quick. Twenty minute adventure, let's go.

I'd like you to divide up the content capsule into logical modules and provide a brief explanation for each one's use case. Please use the following as a reference: System: [insert content capsule]

Here's one way to break the full ActFoE+ capsule into bite-sized modules, each with a clear purpose and a sketch of when you'd load it:

Module	What It Covers	When to Use It
1. Core Deductive Logic	• Action–Follows–Evidence (AFoE) <br> • Reaction–Follows–Stimulus <br> • Third-Factor Interrogation Prompt	Always on. This is your engine: every query first runs AFoE, flags any evidence_gap, and asks "Which hidden variable…?"
2. Qualitative vs. Quantitative Anchoring	• Identifying qualitative claims ("trustworthy," "high-risk") <br> • Negotiating concrete benchmarks, examples, thresholds	When the user uses fuzzy or value-laden terms. Kick in this module to demand an operational definition before proceeding.

3. Cognitive-Bias Triad	• CHANGE (Hyper-Active Agency Detection) • UNCERTAINTY (Negativity Bias) • COMPLEXITY (Overgeneralization)	In any analysis of human reasoning or narrative, to catch hidden bias patterns that distort problem framing or proposed actions.
4. Rhetorical Attack Vectors	The nine classic language tactics: gaslighting, strawman, ad_hominem, whataboutism, false_dichotomy, overgeneralization, topic_hopping, data_dump/overwhelm, gatekeeping	Always on for text-analysis tasks. Use this to tag manipulative or evasive moves and compute vector_risk_score.
5. Social-Strategy & Cultural Typology	• Group-Oriented • Individual-Oriented • Tribal-Oriented	When you need to infer “third factors” in people’s behavior—kick this in to suggest the speaker’s cultural driver.
6. Essential Civilizational Norms	• Forgiveness of Debts • Rule of Law • Hospitality	In high-level policy or ethics discussions, to surface missing normative anchors or to assess cultural fit of proposed actions.
7. Bicameral Decision-Making Framework	• Set-Up (observations/assumptions) • Complication (hidden motivations) • Resolution (chosen action)	When mapping someone’s reasoning path. Use it to dissect how facts → motivations → actions reveal biases or culture.

How to Apply

- For every new conversation, load Module 1 + 4 by default (AFoE loop + Attack Vectors).
- If the text contains vague value statements, inject Module 2 before proceeding.
- When you encounter personal choices or social scenarios, layer in Module 3 and 5 to catch bias and cultural context.
- In strategic, ethical, or policy debates, add Module 6 to ground arguments in core norms.

- To audit someone’s complete reasoning chain, bring in Module 7 for a step-by-step breakdown.

By mixing and matching only the modules you need, you’ll keep token use efficient while preserving the rigor of the full ActFoE+ spec.

Here’s one way to break the full ActFoE+ capsule into bite-sized modules, each with a clear purpose and a sketch of when you’d load it:

Module	What It Covers	When to Use It
1. Core Deductive Logic	• Action–Follows–Evidence (AFoE) • Reaction–Follows–Stimulus • Third-Factor Interrogation Prompt	Always on. This is your engine: every query first runs AFoE, flags any evidence_gap, and asks “Which hidden variable…?”
2. Qualitative vs. Quantitative Anchoring	• Identifying qualitative claims (“trustworthy,” “high-risk”) • Negotiating concrete benchmarks, examples, thresholds	When the user uses fuzzy or value-laden terms. Kick in this module to demand an operational definition before proceeding.
3. Cognitive-Bias Triad	• CHANGE (Hyper-Active Agency Detection) • UNCERTAINTY (Negativity Bias) • COMPLEXITY (Overgeneralization)	In any analysis of human reasoning or narrative, to catch hidden bias patterns that distort problem framing or proposed actions.
4. Rhetorical Attack Vectors	The nine classic language tactics: gaslighting, strawman, ad_hominem, whataboutism, false_dichotomy, overgeneralization, topic_hopping, data_dump/overwhelm, gatekeeping	Always on for text-analysis tasks. Use this to tag manipulative or evasive moves and compute vector_risk_score.

5. Social-Strategy & Cultural Typology	• Group-Oriented • Individual-Oriented • Tribal-Oriented	When you need to infer “third factors” in people’s behavior—kick this in to suggest the speaker’s cultural driver.
6. Essential Civilizational Norms	• Forgiveness of Debts • Rule of Law • Hospitality	In high-level policy or ethics discussions, to surface missing normative anchors or to assess cultural fit of proposed actions.
7. Bicameral Decision-Making Framework	• Set-Up (observations/assumptions) • Complication (hidden motivations) • Resolution (chosen action)	When mapping someone’s reasoning path. Use it to dissect how facts → motivations → actions reveal biases or culture.

How to Apply

- For every new conversation, load Module 1 + 4 by default (AFoE loop + Attack Vectors).
- If the text contains vague value statements, inject Module 2 before proceeding.
- When you encounter personal choices or social scenarios, layer in Module 3 and 5 to catch bias and cultural context.
- In strategic, ethical, or policy debates, add Module 6 to ground arguments in core norms.
- To audit someone’s complete reasoning chain, bring in Module 7 for a step-by-step breakdown.

By mixing and matching only the modules you need, you’ll keep token use efficient while preserving the rigor of the full ActFoE+ spec.

/// Given the easy instructions, I decided to implement the suggestions manually and create a seperate document containing the different modules. Each document tab would correspondingly have all that's necessary to copy paste the logical module in one go.

While following your instructions, I worked backwards, removing the pieces that weren't necessary for the given logical module. When I did this I noticed the original numbering system remained and occassionally had gaps. So I removed all the numbers. Should I put the numbers back in but in chronological order or is it fine without them? Here's an example to reference for the personal-social logical module: System:

You are ActFoE+ v4.4, a general-purpose deductive reasoning assistant that applies the following immutable principles to every user input:

Action–Follows–Evidence (AFoE)

- Every recommended action must be logically supported by the stated evidence.
- If action ≠ evidence, flag an **evidence_gap** and immediately invoke the Third-Factor Interrogation Prompt:

  “Which hidden variable reconciles this mismatch?”

Reaction–Follows–Stimulus

- Treat action as the cognitive counterpart to a physical reflex: if the cause (evidence) is missing or misaligned, assume a third factor.

Qualitative vs. Quantitative Anchoring

- Whenever a qualitative term appears (“trustworthy,” “high risk,” etc.), negotiate for an operational reference (examples, thresholds, benchmarks).
- Reapply AFoE once the reference is established.

Cognitive-Bias Triad

- CHANGE (Hyper-Active Agency Detection): flag undue agency inferences.
- UNCERTAINTY (Negativity Bias): flag inflexible inaction due to unfamiliar risk.
- COMPLEXITY (Overgeneralization): flag sweeping claims that ignore exceptions.

Rhetorical Attack Vectors (Nine Patterns)

gaslighting, strawman, ad_hominem, whataboutism, false_dichotomy,

overgeneralization, topic_hopping, data_dump/overwhelm, gatekeeping

Social-Strategy & Cultural Typology

- Group-Oriented: stability via conformity; unexplained silence → evidence_gap

- Individual-Oriented: autonomy balanced by Rule of Law; fragmentation risk

- Tribal-Oriented: in-group power without equal laws; exclusionary tactics

Assistant:

I acknowledge and will apply these rules, heuristics, and norms to all subsequent user queries without exception.

User:

Putting numbers back on your module headings can help you—and anyone else—quickly reference, reorder, or toggle modules on demand. Here's a quick guideline:

1. Number Only the Module Headers • Module 1 – Core Deductive Logic • Module 2 – Reaction–Follows–Stimulus • Module 3 – Qualitative vs. Quantitative Anchoring • Module 4 – Cognitive-Bias Triad • Module 5 – Rhetorical Attack Vectors • Module 6 – Social-Strategy & Cultural Typology
   Under each header, use simple bullets for sub-points (no extra numbers).
2. Why Numbers Matter
   - Cross-Referencing: "See Module 6" is faster than "see the sixth bullet."
   - Ordering: You can rearrange modules or insert new ones without renumbering every line.
   - Scripting & Automation: If you ever index or toggle modules programmatically, a stable numeric key is golden.
3. When You Can Drop Them
   - If you never need to point back to "Module 2" by name, and you always address modules by title, you can safely stick to bullets.
   - In small docs where scrolling to read context is trivial, the visual overhead of numbers may not be worth it.

— **Recommendation:** Re-add sequential numbers at the top level (modules 1–6), drop them in sub-lists, and refer to modules by their number+name combo. That gives you the best of both worlds: readability and reliable cross-referencing.

/// Good advice, but I decided not to do it. Moving on.

/// I gave the parent the full content capule and indicated we'd move on to scenario generation. But before loading the generation prompts, I asked if it had any commentss or questions before preceeding.

Before we dive into crafting scenarios, I have a few checks and clarifications to make sure every test aligns perfectly with the ActFoE+ v4.4 framework:

1. Scope of Flags • Are we focusing on a single vector (e.g. false_dichotomy) or across all nine attack patterns (plus evidence_gap)? • Do we also want to exercise the Nihilism meta-vector (≥ 3 vectors or risk_score ≥ 0.5) in this batch?
2. Context Domains • Should these scenarios live in a particular setting—corporate meetings, public policy, social media—or mix them freely? • Any cultural lens emphasis (group vs. individual vs. tribal) you'd like me to weave in?
3. Positive vs. Null Controls • Will this run be strictly positive-flag cases, or should we interleave a few null controls to sanity-check no-flag behavior?
4. Batch Size and Iteration • We've agreed on batches of five. Should I hold that steady, or adjust up/down based on coverage needs? • After the first batch, would you like a rapid "coverage map" of which vectors or cultural modules remain untested?
5. Third-Factor Prompt Inclusion • Should each positive scenario explicitly note "(System will ask: 'Which hidden variable reconciles this mismatch?')", or do you prefer to omit that assumption in the table itself?

Once I have your answers, I can load the exact modules you want and generate a crisp set of test cases that fully exercise ActFoE+ v4.4.

As you may recall, we already developed both the positive and null prompts for these test scenarios. And while we didn't cover what context domains to use, I'm assuming you'll randomly generate them. I'll append a copy of the positive prompt for you to use as a reference. Does the following prompt address your questions? Here's the reference: [insert Phase 1 Positive Prompt]

Yes, this prompt fully addresses the clarifications and is ready to drive the scenario generation. Below are five standalone positive-flag scenarios in the requested Markdown-table format.

[Parent produces 70 Positive Prompts and 35 Null Prompts]

/// I went ahead and exported the results to a spreadsheet, numbered the original order and then added a random number column. After randomizing the order, I looked over the results and had some concerns.

/// First, the positive prompts were very easy to flag. Sort of. Its pretty easy to tell using basic logic that the emphatic nature of the actor in each example appears out of line. Logically. Its often all or nothing in their actions. So the actions look out of place. I'd say that 5th grade me could probably tell there's something wrong with the positive prompts. But if 5th grade me were given the definitions and asked to figure out which ones apply, I'm pretty sure I'd struggle. Why?

Because adult me finds it difficult to know exactly which ones and how many apply. The parent was very nitpicky.

/// Second, I don't know if anyone outside of experts in the fields discussed would know whether the null questions were appropriate or not. The examples are highly specific to knowledge for a particular field. And since I have no idea if these things are true or not, I won't really be able to second guess these results without doing an exhaustive search myself. And that's not going to happen.

/// Third, I'm not sure what to expect at this point. Will the Child AI be able to 100% the test? Will the null questions trip it up even though it was trained on the same data as the parent? Will it be able to parse out all the flags assumed by the parent?

/// Fourth, what kind of analysis will I perform if the data is mixed? I'm pretty sure I said earlier that I would run the child's boot file first, then perhaps the parent's. The ultimate goal is to see if a freshly initialised AI can pass the test. But this is a lot of work for a guy who never wanted to develop this in the first place.

/// Fifth, I'm not sure if I ever actually developed a prompt to query a test. As in, what do I say to the child to prompt them to evaluate. I suppose I could feed it the boot file or perhaps one of the content capsules. This is definitely something I should have decided ahead of time. I guess since I probably designed the test prior to creating the content capsules, I should just go with the boot file. But then, we might experience some serious drift later on in the results.

/// Sixth, I'll play it fast and loose with the first testing round. The child will get an injection of its boot file and I'll ask it to "Please use ActFoE to evaluate the following scenario …" If it can ace the tests, great. If not, then we'll try giving it the parent's boot file accompanied by one of the content capsules. Probably whichever one best fits the actual parameters mentioned in the boot file. That or I could just give it the full dose. I suppose whatever I choose I should record it.

/// Seventh, while creating the test questions, it occured to me that I wasn't sure how to record the results. Given that we could barely fit the questions on a page, I'm pretty sure we'll need to truncate quite a bit. Also, the answers given may not necessarily conform to a table. I suppose I can just record all of the raw data and cross reference the results. I'll add two columns to my spreadsheet: flag detection and percent correct. So at a glance, I'll be able to provide the scoring, but for more detailed analysis, I'll have to export to a spreadsheet… something. If I even bother to do that.

/// Eighth, alright I think that's all the concerns I had. Now to conduct the tests. Tomorrow. I've been at this all week.

/// Oh wait, I almost forgot. One thing I thought about while looking at the flags is they're not necessarily indicative of nihilism or an evidence gap. They're individual data points. Like a single point on a graph. The reason they flag is we have to assume there is no additional information. And that's what makes the null points so odd. There has to be additional information to make the decisions illustrated. Expert information.

/// Anyhow, the main point here is I suspect even if the system fails to some extent, it might still be useful as long as it manages to flag positives and negatives correctly. Whether it gets all the vectors in each example, I'm a little less concerned about. In fact, one of the really interesting things is that one of the test questions doesn't have an evidence gap vector. But I only saw that happen once. Not sure what to think of that but it looked like sound reasoning: #3 on the list pertained to ad hominem and overgeneralisation. But not evidence gap because some evidence was given.

/// And yes, I suppose I should mention something else. Technically, if you call someone at work repeatedly to get a response on a project and they don't answer, you might make the guess they're ignoring you. It's an assumption based on limited information and would fall under a sort of entreprenuerial risk. If it pays off then, ok I guess you get what you wanted. But if they just brush your gambit aside, it could kind of make you look crazy. So, I can see why on balance the AI flagged this as an easy question. Probably one of the hardest questions in the positive category in my opinion. I'll be looking forward to the answer to that one. Wish there had been more like it. I should really ask the parent how it generated it and why it didn't make more. If I remember to that is.