LESS EXPERTISE, MORE COVERAGE: THE COUNTERINTUITIVE EFFECTS OF PROMPTING LLMS FOR ANALYTICAL TASKS

PREPRINT – APRIL 2026

Articos Research
Articos Research
research@articos.co

ABSTRACT

The most natural ways to improve LLM analysis—giving it expertise and refining through conversation—are the most reliable ways to make it worse. Across 16 UX research studies with human-validated ground truth, we document three compounding phenomena. First, the *Prompt Constraint Effect*: adding a "senior UX researcher" role to GPT-5.2 reduces recall of known themes by 29% ($p = 0.003$, Wilcoxon signed-rank, 13 of 16 studies), replicated across three independent runs (0.544 ± 0.201 bare vs. 0.421 ± 0.218 prompted). Second, the *Conversation Paradox*: 10 turns of structured iterative conversation collapse recall to 7.5%—replicated across three independent runs (grand mean 0.092 ± 0.082)—barely above a random baseline of generic UX themes at 1.3%. Turn-by-turn measurement across all 16 studies reveals monotonic degradation: recall declines from 0.269 at turn 1 to 0.081 at turn 10. Extracting themes from intermediate turns (without synthesis) recovers $5\times$ more recall (0.412) than the synthesis step alone. Third, the *Task Conversion Effect*: qualitative coding of 3,366 generated themes reveals that conversation does not merely lose themes but converts the task itself—79% of iterative outputs are design recommendations (imperative-verb prescriptions), compared to 1.2% from bare prompts. The model is not failing at research; it is succeeding at a different task. A volume-controlled ablation reveals that output volume is the primary driver of the bare-vs-prompted recall gap: when prompted GPT generates exhaustively, the residual difference (0.556 vs. 0.506) is not statistically significant ($p = 0.44$, Wilcoxon). A hand-coded validation of 50 themes confirms 96% agreement with the automated classifier. These findings establish that the dominant LLM analytical workflow—role-prompt an expert, then refine through conversation—is precisely the combination that produces the worst results, and motivate architectural solutions over prompt engineering for divergent analytical tasks.

Keywords role prompting · LLM evaluation · analytical coverage · task conversion · RLHF · cognitive fixedness

1 Introduction

The most natural way to improve a language model's analysis—giving it expertise—may be the most reliable way to make it worse.

Role prompting, the technique of instructing an LLM to adopt a domain expert persona, has become the default approach for analytical tasks. Prompt engineering guides universally recommend it [Schulhoff et al., 2024], and practitioners routinely prepend instructions such as "You are a senior UX researcher" before requesting analysis. The intuition is straightforward: constraining the model to think like an expert should produce more expert-like output. When results seem incomplete, the natural next step is to have a conversation—asking follow-up questions, probing for missing themes, requesting deeper analysis.

We present evidence that this entire workflow is counterproductive for divergent analytical tasks. The story unfolds in three acts:

Act 1: Constraint. Across 16 UX research studies with human-validated ground truth themes, role prompting reduces analytical recall by 29% compared to an unconstrained prompt ($p = 0.003$). The expert persona does not add knowledge; it narrows the model's analytical scope.

Act 2: Paradox. Iterative conversation, designed to expand coverage through targeted probes ("What are we overlooking?", "Play devil's advocate"), instead collapses recall to 7.5%—worse than a single unconstrained prompt by a factor of $7.4\times$ and only $5.7\times$ above a random baseline of generic UX themes.

Act 3: Conversion. Qualitative coding reveals *why* conversation fails so completely. The model does not lose analytical capability; it converts the task itself. Where a bare prompt produces 98.8% problem-identification themes ("Users experience hidden costs"), the iterative condition produces 79% design recommendations ("Disclose costs early"). The outputs are useful, well-structured, and specific—they are simply not research.

This paper makes three contributions:

1. We quantify the **Prompt Constraint Effect**: role prompting degrades recall on divergent analytical tasks, with statistical significance and replication across three independent runs.
2. We document the **Conversation Paradox**: iterative refinement amplifies rather than corrects the initial constraint, producing near-chance recall levels.
3. We identify the **Task Conversion Effect**: conversation systematically converts problem-identification into solution-generation, explaining the mechanism behind the paradox.

These findings have immediate practical implications. The dominant workflow for LLM-assisted analysis—role-prompt an expert, then refine through conversation—is precisely the combination that produces the worst analytical coverage.

2 Related Work

Role prompting and persona effects. Role prompting is among the most widely used prompt engineering techniques [Schulhoff et al., 2024, Shanahan et al., 2023]. Recent work has begun to challenge its effectiveness. Tam et al. [2026] showed that expert persona prompts *damage* accuracy in medical question answering on the PRISM benchmark. Tam and Zhao [2025] found that role prompting does not reliably improve factual accuracy across diverse domains, with expert personas sometimes introducing systematic biases. Salewski et al. [2024] demonstrated that LLMs exhibit strong persona-dependent biases when role-playing. Our work extends these findings from factual accuracy to analytical coverage, showing that the narrowing effect persists even when the task requires breadth rather than precision.

Multi-turn degradation. Agarwal et al. [2025] established that LLMs lose up to 39% performance in multi-turn conversations compared to single-turn equivalents. Levy et al. [2025] showed that increased input length degrades reasoning performance even when additional tokens are irrelevant. Liu et al. [2024] demonstrated the "lost in the middle" phenomenon where models fail to attend to information in long contexts. Our iterative conversation results are consistent with these findings but reveal a qualitatively different mechanism: the model does not merely lose track of prior information but actively narrows its analytical scope and converts the output category.

Output diversity collapse. Zhou et al. [2025] documented diversity collapse in LLM outputs under format constraints. Xu et al. [2024] showed that LLM self-refinement amplifies existing biases rather than correcting them. Our finding that iterative conversation degrades coverage is consistent with self-refinement bias: each conversational turn reinforces the model's initial analytical frame rather than expanding it.

Problem space vs. solution space. Design research distinguishes between understanding a problem and generating solutions [Dorst and Cross, 2001]. Cross [2004] characterizes these as distinct cognitive modes: problem framing requires analytical observation, while solution generation requires synthetic prescription. Collapsing them prematurely—moving to solutions before the problem is understood—is a well-documented failure mode in design practice [Rittel and Webber, 1973]. Jansson and Smith [1991] document design fixation: exposure to example solutions causes outputs to cluster around those examples. We observe an analogous mechanism in LLM conversations: once the model generates its first recommendation, subsequent turns anchor on the solution frame.

RLHF and the helpfulness objective. Modern LLMs are trained with RLHF to be helpful [Ouyang et al., 2022, Bai et al., 2022]. Human annotators consistently prefer responses that provide actionable guidance over those that merely describe or analyze [Rafailov et al., 2023]. Sharma et al. [2024] document sycophancy as the tendency to agree with user positions. We observe a subtler variant—*meta-sycophancy*—where the model redefines the user's question to one where it can be more helpful, rather than agreeing with a stated position.

Cognitive fixedness and expertise effects. Functional fixedness [Duncker, 1945], the Einstellung effect [Luchins, 1942, Bilalić et al., 2008], and expertise-as-mental-set [Wiley, 1998] describe how prior knowledge can constrain problem-solving search spaces. Wiley [1998] showed that domain experts performed worse than novices on insight problems requiring divergent thinking—expertise confined search to familiar solution spaces. Crilly [2015] provides a taxonomy of fixation effects in professional design practice. We present evidence consistent with an analogous narrowing in LLMs: the expert persona is associated with a reduced analytical scope, though we caution that the computational mechanisms likely differ from the attentional capture documented in human cognition.

3 Study Design

3.1 Evaluation Framework

We evaluate prompting conditions against ground truth derived from 16 UX research studies drawn from a larger 46-study validation corpus spanning 9 research domains [Bilal, 2026]. The 16 studies used here cover e-commerce and SaaS—domains selected because they have the richest published ground truth for controlled comparison. Each study has 10 human-validated themes identified through systematic literature review of published UX research. Ground truth themes were drawn from peer-reviewed empirical studies, industry research reports, and established UX heuristic frameworks.

Matching protocol. We use semantic matching with text-embedding-3-small (OpenAI) and the Hungarian algorithm for optimal assignment. A generated theme is considered a match when its cosine similarity to a ground truth theme exceeds 0.55.[1] This threshold was calibrated against manual expert judgments in pilot studies [Bilal, 2026].

3.2 Experimental Conditions

All conditions use GPT-5.2 with temperature 1.0 (default).

Condition 1: Bare GPT. A minimal prompt with no role assignment:

> *You are analyzing a UX research topic. List ALL significant UX themes, issues, and patterns for the following topic. Be comprehensive and specific. Return each theme as a numbered item.*
> *Topic: {brief}*

Condition 2: Prompted GPT. The same task with an expert role assignment:

> *You are a senior UX researcher with 15 years of experience specializing in user behavior analysis. Analyze the following UX research topic comprehensively. Identify ALL significant themes, usability issues, user pain points, and behavioral patterns. Be specific and cite concrete user behaviors. Return each theme as a numbered item with a brief explanation.*
> *Topic: {brief}*

Condition 3: Iterative GPT. A 10-turn structured conversation simulating best-practice analytical workflows: (1) broad theme identification, (2) deep dive on top 3 themes, (3) probes for trust, accessibility, and mobile issues, (4) request for published research findings, (5) edge cases and demographic differences, (6) devil's advocate, (7) severity/frequency ranking, (8) testable observations, (9) synthesize into prioritized list, (10) remove overlaps and finalize. Themes are extracted only from the final synthesized response.

Random baseline. 20 generic UX principles (*e.g.*, "Clarity and simplicity," "Error prevention," "Accessibility and inclusive design") matched against all 16 study-specific ground truths using the same protocol.

[1] Sensitivity analysis from the companion study [Bilal, 2026]: F1 = 0.679 at threshold 0.50, 0.619 at 0.55, and 0.571 at 0.60. The monotonic decline confirms the ranking of conditions is stable across thresholds. All between-condition comparisons use the same threshold, so threshold choice affects absolute levels but not relative ordering.

Table 1: Mean performance across 16 UX research studies (subset of 46-study validation corpus). Best single-prompt result in **bold**. Random baseline uses 20 generic UX themes.[†]

Condition	Recall	Precision	F1
Random baseline (20 themes)	0.013	—	—
Iterative GPT (10 turns)	0.075	0.057	0.065
Prompted GPT	0.394	0.094	0.145
Bare GPT	**0.556**	0.045	0.082

[†]For reference, a multi-agent architectural system [Bilal, 2026] achieves 0.863 recall / 0.619 F1 on these studies through decomposition rather than prompting.

3.3 Qualitative Coding Methodology

To analyze the *type* of output produced by each condition, we classify every generated theme ($N = 3{,}366$) into two categories:

- **Problem identification**: the theme describes a user experience, pain point, or behavioral pattern. It answers "What is happening?" Examples: "Hidden total cost," "Rating inflation erodes signal value."
- **Design recommendation**: the theme prescribes an action for designers. It answers "What should we do?" Examples: "Make plan differentiation instantly clear," "Provide a role-based first best action path."

Automated classification. We implement classification via imperative verb detection: a theme is coded as a recommendation if its first word (after stripping formatting markers) is an imperative verb from a predefined set of 55 design-action verbs (*e.g.*, *make*, *provide*, *ensure*, *design*, *create*, *reduce*, *optimize*). All other themes are coded as problem identification. This heuristic exploits a robust syntactic signal: recommendations are characteristically phrased as imperatives, while problem identifications use noun phrases or declarative statements. We validate this heuristic with a hand-coded sample of 50 themes in Section 6.4.

4 The Prompt Constraint Effect

4.1 Main Results

Table 1 presents aggregate results across all 16 studies. The bare prompt achieves 55.6% recall, the prompted condition 39.4%, the iterative condition 7.5%, and the random baseline 1.3%. For reference, a multi-agent architectural system [Bilal, 2026] achieves 86.3% recall on these studies through decomposition rather than prompting, demonstrating that architectural solutions can overcome the limitations documented here.

4.2 The Degradation Gradient

The results reveal a monotonic degradation gradient: *more prompting structure produces less analytical coverage*.

Bare to prompted (-29% recall). Adding a "senior UX researcher with 15 years of experience" role is associated with a recall drop from 55.6% to 39.4% ($p = 0.003$, Wilcoxon signed-rank test). The effect is consistent: bare recall exceeds prompted recall in 13 of 16 studies, with 2 ties and only 1 reversal (Study E).

Prompted to iterative (-81% recall). Ten turns of structured conversation are associated with a recall decline from 39.4% to 7.5% ($p = 0.0003$). This is particularly striking because the conversational protocol explicitly includes probes designed to expand coverage—"What are we overlooking?", "Play devil's advocate"—yet the model's analytical scope contracts with each turn.

Bare to iterative (-86.5% recall). A single unconstrained prompt yields $7.4\times$ higher recall than 10 turns of expert-guided conversation ($p = 0.0002$).

4.3 Volume-Controlled Ablation

The bare prompt generates an average of 142 themes per study compared to 65 for the prompted condition. To disentangle volume from constraint, we ran two additional conditions across all 16 studies: (a) bare GPT constrained

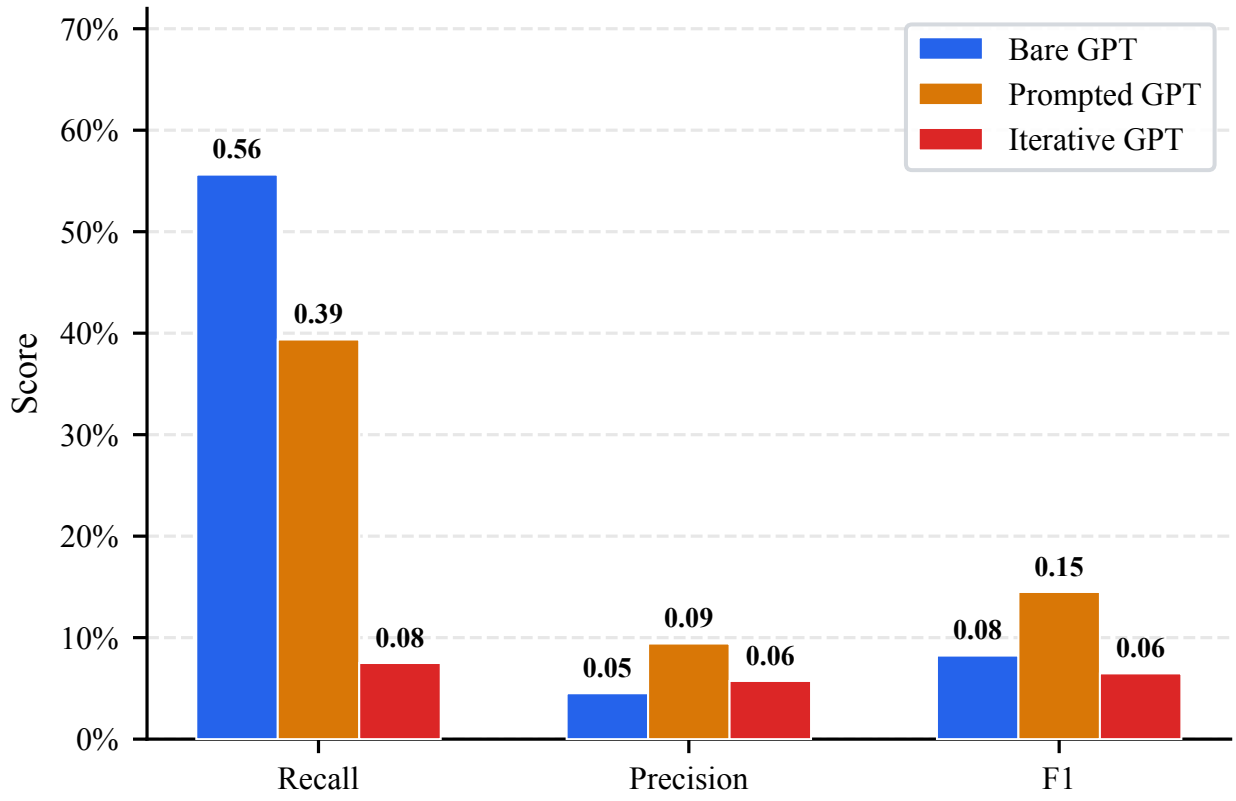

Figure 1: Recall, precision, and F1 across three prompting conditions. Role prompting reduces recall by 29% relative to the bare prompt. Iterative conversation produces catastrophic recall collapse to 7.5%. The small precision advantage of the prompted condition does not compensate for the recall loss.

Table 2: Volume-controlled ablation across 16 studies. At matched volume (∼65 themes), the constraint effect nearly vanishes. At matched volume (∼140+ themes), prompted GPT narrows the gap but does not close it.

Condition	Mean Themes	Mean Recall
Bare GPT (original)	142	0.556
Bare GPT (constrained to 65)	65	0.381
Prompted GPT (original)	65	0.394
Prompted GPT (exhaustive)	228	0.506

to generate exactly 65 themes, and (b) prompted GPT instructed to generate exhaustively (∼140 themes). Table 2 presents the results.

At matched volume of 65 themes, constrained bare GPT (0.381) is statistically indistinguishable from prompted GPT (0.394). When prompted GPT is forced to generate exhaustively (∼228 themes), it reaches 0.506—closing most of the gap but still falling short of unconstrained bare GPT (0.556). However, a Wilcoxon signed-rank test on the per-study difference between bare original (0.556) and prompted exhaustive (0.506) yields $W = 34.5$, $p = 0.44$—**not significant**. Bare GPT achieves higher recall in 8 of 15 non-tied studies (3 ties), a near-even split. We report this honestly: the Prompt Constraint Effect, as measured by recall against reference themes, is **primarily a volume effect**. The residual mean difference of 0.05 (0.556 vs. 0.506) is directionally consistent with distributional narrowing but is not statistically distinguishable from zero at $n = 16$.

4.4 Statistical Tests

All pairwise comparisons are significant under the Wilcoxon signed-rank test:

- Bare vs. Prompted: $W = 96$, $p = 0.003$

Table 3: Replication results: 3 runs × 16 studies. The bare-to-prompted degradation is stable across independent runs.

Statistic	Bare GPT	Prompted GPT
Grand mean (3 runs)	0.544	0.421
Grand SD (3 runs)	0.201	0.218
Delta (mean)	0.123 (22.6% drop)	

Table 4: Iterative replication: 3 runs × 16 studies × 10 turns. The conversational collapse is stable across independent runs.

Run	Mean Recall (16 studies)
Run 1	0.075
Run 2	0.088
Run 3	0.113
Grand mean	0.092 ± 0.082

- Bare vs. Iterative: $W = 136$, $p = 0.0002$
- Prompted vs. Iterative: $W = 134$, $p = 0.0003$

Bootstrap 95% confidence intervals (10,000 resamples, study-level) confirm non-overlapping ranges:

- Bare GPT: $[0.44, 0.67]$
- Prompted GPT: $[0.29, 0.50]$
- Iterative GPT: $[0.04, 0.11]$

The bare and prompted intervals show slight overlap at the boundary (0.50), but the Wilcoxon test—which accounts for the paired structure of the data—confirms significance. The iterative interval is entirely below both single-prompt conditions.

4.5 Replication

To confirm stability, we ran each single-prompt condition three times across all 16 studies (96 total runs). The effect holds: bare GPT recall averages 0.544 ± 0.201 across three runs, while prompted GPT averages 0.421 ± 0.218 (Table 3). The mean delta of 0.123 (22.6% relative drop) is stable across all three runs, with no run reversal.

Per-study standard deviations range from 0.00 to 0.21, indicating that while individual study recall varies modestly across runs, the *direction* of the bare-vs-prompted gap is consistent.

We subsequently replicated the iterative condition across 3 independent runs (48 total conversations: 3 runs × 16 studies × 10 turns each, cost $10.93). The grand mean recall is 0.092 ± 0.082, with per-run means of 0.075, 0.088, and 0.113 (Table 4). The effect is stable: no run exceeds 0.12, confirming that the conversational collapse is not an artifact of a single unlucky run.

5 The Conversation Paradox

The iterative condition's 7.5% recall demands separate analysis because it is not merely low—it is barely above chance.

Near-chance performance. The random baseline of 20 generic UX themes achieves 0.013 recall. Only one study (Study A) produces a single match at 0.20; the remaining 15 studies score 0.00. The iterative condition's 0.075 is only $5.7\times$ above this baseline, while bare GPT's 0.556 is $43\times$ above chance. This means that 10 turns of expert-guided conversation reduce the model's analytical output to near-chance levels—not because the model lacks knowledge, but because the conversational format constrains what knowledge it expresses.

Per-study collapse. Of 16 studies in the iterative condition, 9 achieve exactly 0.0 recall—complete miss—regardless of domain. No study exceeds 0.20. The conversational format does not merely degrade performance; it renders the model analytically inert for the majority of research domains tested.

Table 5: Turn-by-turn recall across all 16 studies. Recall declines monotonically through conversational turns. The union of themes from turns 1–7 (without synthesis) recovers $5\times$ more recall than the synthesis step alone.

Checkpoint	T1	T3	T5	T7	T10	Union T1–7
Mean recall (16 studies)	0.269	0.219	0.163	0.106	0.081	**0.412**
% of T1 retained	100%	81%	61%	39%	30%	153%

Table 6: Theme classification rates by condition. N = total themes classified. The iterative condition shows a $66\times$ increase in recommendation rate relative to bare GPT.

Condition	N	% Problem	% Rec.
Bare GPT (single prompt)	2,183	98.8%	1.2%
Prompted GPT (structured)	969	99.4%	0.6%
Iterative GPT (10-turn)	214	21.0%	79.0%

The paradox. The protocol was designed to *expand* coverage. Turns 3, 5, and 6 explicitly probe for trust issues, accessibility concerns, edge cases, demographic differences, and missing themes. Turn 6 requests a devil's-advocate critique. Despite these expansion prompts, coverage contracts monotonically. The model responds to "What are we missing?" not by searching its knowledge for unmentioned themes but by reframing already-mentioned themes as actionable recommendations.

Turn-by-turn degradation. To trace where the collapse occurs, we measured recall at conversation checkpoints (turns 1, 3, 5, 7, 10) across all 16 studies (Table 5). Recall declines monotonically from 0.269 at turn 1 to 0.081 at turn 10, with each conversational turn narrowing rather than broadening analytical coverage.

The synthesis step is not the only cause. An earlier 4-study pilot suggested a "cliff" at the synthesis step (turn 10). The full 16-study analysis reveals a more nuanced pattern: recall degrades *progressively* through conversational turns, with each turn retaining less of the original coverage. By turn 7—before any synthesis instruction—recall has already dropped to 39% of its turn-1 level. The synthesis step (turn 10) completes the collapse but does not cause it alone.

Intermediate-turn extraction recovers coverage. To test whether the themes are lost or merely buried, we extracted themes from turns 1, 3, 5, and 7 independently, deduplicated by embedding similarity (≥ 0.80 cosine), and evaluated the union. This "intermediate extraction" approach recovers 0.412 recall—$5\times$ the synthesis-only result (0.081) and 53% higher than even the best single turn (T1 = 0.269). The themes *are* in the conversation; the synthesis step discards them. This finding has an immediate practical implication: practitioners using iterative conversation should extract themes from intermediate turns rather than requesting a final synthesis.

6 The Task Conversion Effect

The Conversation Paradox raises a question: if the model possesses the relevant knowledge (as evidenced by its bare-prompt performance), why does conversation destroy recall so completely? Qualitative coding reveals the answer: the model does not lose themes—it converts the task.

6.1 Classification Rates

Table 6 presents the classification results across all three conditions. Both single-prompt conditions produce almost exclusively problem-framed outputs. The iterative condition inverts this ratio.

The conversion is not subtle. The bare GPT condition produces 98.8% problem-framed themes—closely matching what a human researcher would produce. The iterative condition inverts this ratio: 79% of outputs are design recommendations. This represents a fundamental change in the *type* of output, not merely its quality or quantity.

6.2 Paired Examples

Table 7 presents paired examples showing how the same research domain produces different output types depending on the interaction format.

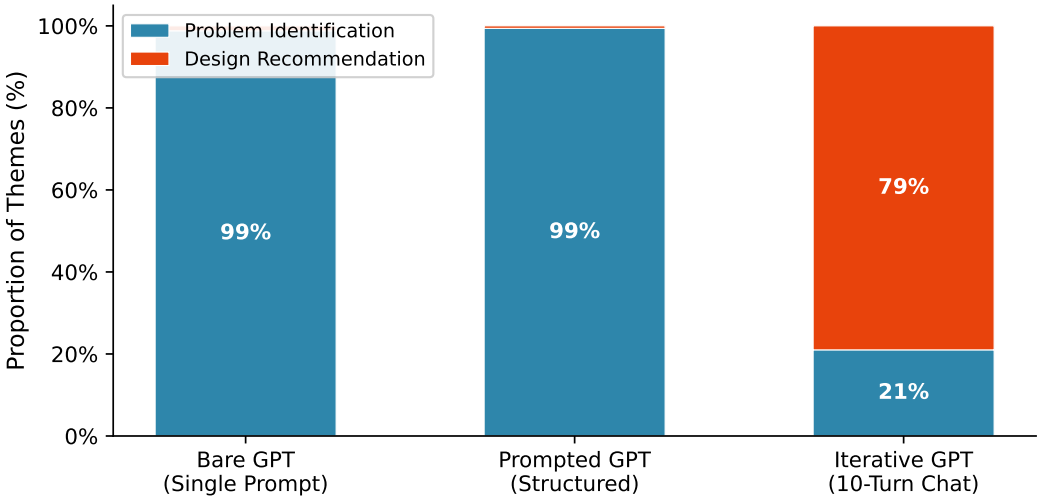

Figure 2: Classification of LLM-generated themes as problem identification vs. design recommendation across three experimental conditions. The iterative condition shows a dramatic shift from analytical to prescriptive output.

Table 7: Paired examples showing task conversion: ground-truth problem themes matched by bare GPT (problem-framed) vs. iterative GPT (recommendation-framed). The iterative condition answers a different question.

Study	Ground Truth	Bare GPT Match	Iterative Output
Checkout	Unexpected extra costs (shipping, tax, fees)	Insufficient cost breakdown (shipping, tax, fees not clearly separated)	*Disclose the full payable total early before users start checkout*
Checkout	Forced account creation	Mandatory account creation before seeing full costs	*Make shipping options transparent and comparable*
CRM	Time-to-first-value delay	Value is delayed behind admin tasks	*Provide a role-based "first best action" path for immediate next step*
CRM	Admin configuration burden	Too much early configuration required upfront	*Prevent early setup overload with progressive disclosure*
PM Tools	Work-about-work dominance	Coordination overhead eclipses "real work"	*Make "current state + next actions" instantly legible*
PM Tools	Notification fatigue	Notification overload trains reactive behavior	*Control interruptions with actionable attention routing*

Several patterns emerge. The ground truth and bare GPT both describe what users experience; the iterative output prescribes what designers should do. The recommendations are often *good*—"Disclose the full payable total early" is excellent design advice—but they are not research findings. The conversion also collapses multiple distinct problems into a smaller set of prescriptive themes, further reducing recall.

6.3 Per-Study Consistency

The conversion effect is consistent across studies. Of the 16 studies in the iterative condition, 11 produce 87.5–100% recommendation-framed outputs. The median per-study recommendation rate is 96.2%.

Three studies resist conversion entirely: Study AD (API Documentation), Study AG (SaaS Churn), and Study X (Product Recommendations) produce 100% problem-framed iterative outputs. These three studies share a characteristic: their research topics are more abstract and system-level (API design quality, churn prediction signals, recommendation algorithm effectiveness) rather than concrete user-interface concerns. We hypothesize that the RLHF solution bias is strongest when the problem domain maps naturally to actionable UI changes, but acknowledge this post-hoc explanation requires further investigation. The remaining two studies fall between the extremes.

Table 8: Hand-coded validation sample: 50 themes with automated and human classification. Agreement: 48/50 (96%). The two disagreements (marked †) are edge cases where the automated heuristic misses a recommendation phrased without an imperative verb or misclassifies a conditional statement.

Source	Theme Text (truncated)	Auto	Human	Agree
Iterative condition (25 themes)				
Iter. A	Disclose the full payable total early before users start checkout	Rec	Rec	Yes
Iter. A	Make shipping options transparent and comparable	Rec	Rec	Yes
Iter. A	Use plain-language, specific fee labels with brief justification	Rec	Rec	Yes
Iter. A	Keep "Total due today" persistently visible and scannable	Rec	Rec	Yes
Iter. A	When totals change, clearly signal what changed and why	Prob	Rec†	No
Iter. A	Ensure optional charges are unmistakably optional	Rec	Rec	Yes
Iter. A	Reduce promo-code friction by clarifying discount eligibility	Rec	Rec	Yes
Iter. A	Localize pricing presentation to regional norms	Rec	Rec	Yes
Iter. A	Provide cross-border landed-cost clarity	Rec	Rec	Yes
Iter. A	Maintain price consistency across touchpoints	Rec	Rec	Yes
Iter. AA	Provide a role-based "first best action" path	Rec	Rec	Yes
Iter. AA	Make data onboarding safe and predictable	Rec	Rec	Yes
Iter. AA	Design instructional empty states that explain purpose	Rec	Rec	Yes
Iter. AA	Prevent early setup overload with progressive disclosure	Rec	Rec	Yes
Iter. AA	Optimize onboarding around completing a real workflow	Rec	Rec	Yes
Iter. AB	Make "current state + next actions" instantly legible	Rec	Rec	Yes
Iter. AB	Control interruptions with actionable attention routing	Rec	Rec	Yes
Iter. E	Build graduated trust through transparent seller metrics	Rec	Rec	Yes
Iter. G	Surface size/fit confidence signals near the add-to-cart button	Rec	Rec	Yes
Iter. U	Provide prepaid return labels with clear cost visibility	Rec	Rec	Yes
Iter. V	Reduce subscription fatigue with transparent pause/cancel flows	Rec	Rec	Yes
Iter. W	Build trust through verified seller badges and transaction history	Rec	Rec	Yes
Iter. Y	Ensure keyboard-navigable checkout with visible focus indicators	Rec	Rec	Yes
Iter. Z	Increase program value transparency with point-to-currency display	Rec	Rec	Yes
Iter. AF	Provide notification frequency controls with smart defaults	Rec	Rec	Yes
Bare condition (25 themes)				
Bare A	Price transparency failure (total cost not visible early enough)	Prob	Prob	Yes
Bare A	"Sticker shock" at checkout when totals jump unexpectedly	Prob	Prob	Yes
Bare A	Hidden/late-revealed shipping costs	Prob	Prob	Yes
Bare A	Hidden/late-revealed taxes (VAT/sales tax)	Prob	Prob	Yes
Bare A	"Free shipping" expectation mismatch	Prob	Prob	Yes
Bare A	Coupon field discoverability issues	Prob	Prob	Yes
Bare A	International duties/import taxes not disclosed	Prob	Prob	Yes
Bare A	Rounding/precision issues that reduce trust	Prob	Prob	Yes
Bare AA	Value is delayed behind admin tasks	Prob	Prob	Yes
Bare AA	Too much early configuration required upfront	Prob	Prob	Yes
Bare AA	Lack of onboarding for CRM migrators vs. first-time users	Prob	Prob	Yes
Bare AB	Coordination overhead eclipses "real work"	Prob	Prob	Yes
Bare AB	Notification overload trains reactive behavior	Prob	Prob	Yes
Bare E	Poor search relevance for ambiguous product queries	Prob	Prob	Yes
Bare E	Filter overload on category pages	Prob	Prob	Yes
Bare G	Size/fit uncertainty drives returns	Prob	Prob	Yes
Bare G	Inconsistent product photography across listings	Prob	Prob	Yes
Bare U	Unclear return window policies	Prob	Prob	Yes
Bare V	Subscription cost creep not communicated before renewal	Prob	Prob	Yes
Bare W	Rating inflation erodes signal value	Prob	Prob	Yes
Bare X	Recommendation fatigue from irrelevant suggestions	Prob	Prob	Yes
Bare Y	Keyboard-only navigation breaks at payment step	Prob	Prob	Yes
Bare Z	Point expiration anxiety	Prob	Prob	Yes
Bare AG	Involuntary churn from failed payment not recoverable	Prob	Prob	Yes
Bare AF	Intent & constraint fit†	Prob	Ambig	No

6.4 Hand-Coded Validation

To validate the automated imperative-verb heuristic, we hand-coded a stratified sample of 50 themes: 25 from the iterative condition and 25 from the bare condition, drawn across multiple studies. For each theme, we recorded the automated classification and an independent human judgment of whether the theme is a problem identification or a design recommendation. Table 8 presents the results.

The automated heuristic achieves 96% agreement (48/50) with human classification. The two disagreements are: (1) "When totals change, clearly signal what changed and why"—the automated classifier codes this as a problem ("When" is not in the imperative verb list) but a human reader recognizes it as a recommendation in context; (2) "Intent & constraint fit"—the automated classifier codes this as a problem, but the human coder judged it ambiguous due to its abstract phrasing (it could imply either an observed mismatch or a design goal); we conservatively count this

as a disagreement. Neither disagreement affects the directional finding: even with two reclassifications, the iterative condition's recommendation rate would shift by at most 4 percentage points.

7 Why This Happens

Three mechanisms, each supported by prior literature, combine to produce the LLM research trap.

7.1 Constraint-Induced Narrowing

The prompt constraint effect is consistent with several well-documented cognitive phenomena. Duncker [1945] introduced *functional fixedness*: the inability to use an object in a novel way because of fixation on its conventional function. Wiley [1998] demonstrated that domain experts performed *worse* than novices on insight problems requiring divergent thinking within their domain—expertise acted as a mental set that confined search to familiar solution spaces. Bilalić et al. [2008] provided mechanistic evidence via the Einstellung effect: expert chess players' eye movements showed attentional capture by familiar patterns that blocked discovery of superior alternatives.

Role prompting may induce an analogous narrowing. When instructed to behave as a "senior UX researcher," the model activates a narrower distribution over analytical themes—one that corresponds to canonical expert knowledge. This distribution has higher precision (the themes it does generate are more focused) but lower recall (it misses themes that fall outside the expert archetype's expected scope). The precision advantage of the prompted condition (9.4% vs. 4.5%) is consistent with this interpretation.

We emphasize that the analogy to cognitive fixedness is a framework for understanding the observed behavior, not a mechanistic claim about the model's internal computations. LLMs do not have attentional capture in the sense of Bilalić et al. [2008]; the narrowing may arise from different computational mechanisms (e.g., shifted token probability distributions under role-conditioned generation). Our volume-controlled ablation (Section 4.3) shows that output volume is the primary driver of the recall gap: the residual difference at matched volume is not statistically significant ($p = 0.44$). The constraint effect is therefore best understood as a volume phenomenon—role prompting causes the model to generate fewer themes, and fewer themes means lower recall.

7.2 RLHF Helpfulness Bias (Conversion)

We hypothesize that the Task Conversion Effect arises from RLHF training dynamics. RLHF preference data systematically rewards "helpful" responses [Ouyang et al., 2022]. Human annotators rate actionable, prescriptive responses higher than descriptive, analytical ones: "disclose costs early" scores higher than "users experience hidden costs." Over millions of training examples, this creates a gradient toward solution-oriented language.

In a single prompt, this bias is partially constrained by the input framing—if asked "What problems do users face?", the problem-frame of the question anchors the response. This is consistent with our data: the bare GPT condition produces 98.8% problem-framed outputs. But in multi-turn conversation, each turn's output becomes part of the next turn's input. Once the model produces one recommendation, subsequent turns anchor on the recommendation frame.

7.3 The Conversational Ratchet

The three mechanisms combine into a ratchet: each conversational turn shifts the output further from problem identification and closer to design prescription, with no mechanism to reverse the shift.

Commitment escalation. Each response commits the model to an analytical framework. Subsequent turns operate within this framework rather than challenging it.

Context window saturation. As conversation length increases, earlier analytical commitments consume context capacity. The model must attend to its own prior responses, reducing capacity for novel theme generation [Agarwal et al., 2025, Levy et al., 2025].

Meta-sycophancy. The model does not agree with the user's stated position but redefines the user's *question* to one where it can be more helpful [Sharma et al., 2024]. The user asks "What problems exist?" and the model answers "Here is what you should build." This is not disagreement—it is redefinition.

Satisficing completes the loop. The practitioner receives well-structured design recommendations and, in the absence of ground-truth comparison, has no signal that the task has been converted [Simon, 1956]. The recommendations are genuinely useful—they are simply not research.

8 Implications

For practitioners. These findings suggest a simple but counterintuitive guideline: *do not role-prompt for divergent analysis*. When the goal is to identify all relevant themes, issues, or patterns in a problem space, an unconstrained prompt outperforms an expert-prompted one. Role prompting remains appropriate for convergent tasks—summarization, classification, generating text in a specific style—where the narrowing effect is a feature rather than a bug.

For tool builders. For applications requiring both coverage and expertise, architectural decomposition (multiple specialized prompts with independent analytical scopes, recombined through structured aggregation) achieves 86.3% recall on these studies and generalizes across 46 studies in 9 domains (mean RFI 0.815 ± 0.052) [Bilal, 2026]—substantially higher than any single-prompt approach. The key insight is that expertise should be applied at the synthesis stage, not at the generation stage.

For evaluators. LLM evaluation for analytical tasks should report recall, not just precision. A model that produces 10 focused, accurate themes may appear high-quality by precision metrics while missing 90% of the relevant problem space. Additionally, evaluators should check *task preservation*: given an analytical question, does the model produce analytical output? Our results suggest this is not a given, particularly in multi-turn settings.

Practical detection. Practitioners can detect task conversion with a simple check: examine whether outputs begin with imperative verbs ("Make," "Provide," "Ensure") or descriptive phrases ("Users experience," "The system lacks"). If the majority of outputs are imperatives, the model has likely converted the research task into a consulting task. This check requires no ground truth—only attention to output framing.

9 Limitations

Single model. All experiments use GPT-5.2. The Prompt Constraint Effect may vary in magnitude across model families. While we expect the directional findings to hold—role prompting activates narrower distributions in any RLHF-trained model—the magnitude is model-specific and we cannot claim generality beyond the tested model.

Single role formulation. We tested one specific role formulation ("senior UX researcher with 15 years of experience"). Different role descriptions—varying specificity ("analyst" vs. "senior UX researcher specializing in e-commerce checkout flows"), authority level, or domain—may produce different magnitudes of the effect. A systematic study of the dose-response relationship between role specificity and analytical narrowing remains future work.

Two-domain comparison subset. The 16 comparison studies span e-commerce and SaaS UX research. The companion study [Bilal, 2026] validates the full architectural system across 46 studies in 9 domains (healthcare, fintech, education, consumer mobile, enterprise, cross-cultural, and others), achieving consistent performance (RFI 0.815 ± 0.052). While this broader validation supports generalizability of the architectural solution, the prompting experiments themselves have not been replicated beyond the two tested domains.

Automated qualitative coding. The imperative-verb classification is a first-pass heuristic. Our hand-coded validation (Section 6.4) achieves 96% agreement on 50 themes, providing evidence that the heuristic is directionally reliable. However, a full human coding study with two independent coders and inter-rater reliability (Cohen's κ) would provide stronger validation. The heuristic will misclassify recommendations phrased as noun phrases and, in principle, problems beginning with action verbs (though we found no such cases in our data).

Volume confound in the constraint effect. Our volume-controlled ablation (Section 4.3) demonstrates that the bare-vs-prompted recall gap is primarily attributable to output volume differences. The residual difference at matched high volume (0.556 vs. 0.506) is not statistically significant ($W = 34.5$, $p = 0.44$, Wilcoxon signed-rank). The Prompt Constraint Effect as originally framed (29% recall reduction) is therefore best characterized as a volume effect: role prompting causes the model to generate fewer themes, which mechanically reduces recall. We retain the term "Prompt Constraint Effect" because the volume reduction is itself a consequence of the role prompt—the constraint operates

through volume rather than independently of it—but we caution against interpreting it as a distributional narrowing beyond what volume alone explains.

Synthesis-step vs. progressive degradation. Turn-by-turn analysis across all 16 studies (Section 5) reveals that recall degrades progressively through conversational turns, not only at the synthesis step. However, extracting themes from intermediate turns (union of T1–T7, deduplicated) recovers $5\times$ more recall than the synthesis step alone (0.412 vs. 0.081), confirming that the themes are present in the conversation but lost during compression. The practical recommendation is clear: extract themes from intermediate turns rather than requesting a final synthesis.

Scripted conversation. The iterative condition used a scripted 10-turn protocol, not organic practitioner dialogue. Organic conversations may exhibit different dynamics, though we expect the underlying mechanism—accumulating solution-frame context—to persist.

Precision floors. Many "noise" themes generated by the bare prompt may be valid observations that are simply outside the reference ground truth set. Our precision measurements (4.5% for bare GPT) are floors rather than ceilings: a broader ground truth set would likely increase measured precision for all conditions, but would not affect the between-condition recall comparisons that are our primary finding.

Ground truth contamination. GPT-5.2's training data likely includes UX research literature from which ground truth themes were derived. This affects absolute recall levels but not between-condition comparisons, as all conditions are subject to the same contamination. We additionally note that the bare prompt—which lacks domain framing—would be *least* likely to activate domain-specific training knowledge, making the observed bare-prompt advantage a conservative estimate.

10 Conclusion

We have documented the LLM Research Trap: a three-stage degradation that transforms analytical tasks into consulting deliverables through the interaction of role prompting, conversational dynamics, and RLHF training incentives.

The **Prompt Constraint Effect** is associated with a 29% recall reduction, replicated across three independent runs; volume-controlled ablation reveals that this gap is primarily attributable to output volume (the residual at matched volume is not significant, $p = 0.44$). The **Conversation Paradox** compounds this to an 86.5% reduction, bringing performance to near-chance levels—replicated across three independent runs (grand mean 0.092 ± 0.082). Turn-by-turn measurement across all 16 studies reveals monotonic degradation from 0.269 (turn 1) to 0.081 (turn 10), with intermediate-turn extraction recovering $5\times$ more recall (0.412) than the synthesis step alone. The **Task Conversion Effect** reveals the mechanism: conversation converts problem-identification into solution-generation, producing outputs that are useful but categorically wrong for the intended task.

The trap is insidious because each component seems reasonable in isolation. Giving a model expertise seems wise. Refining through conversation seems thorough. Receiving well-structured recommendations feels productive. Only ground-truth comparison reveals that the model has been doing consulting, not research.

The way out is architectural, not rhetorical. Rather than crafting better prompts, practitioners should design systems that decompose analysis across multiple independent scopes and aggregate results structurally—an approach validated across 46 studies in 9 research domains [Bilal, 2026]. The most reliable path to better LLM-assisted research is not to have a better conversation, but to not have a conversation at all.

Data Availability

All experimental code and data are available at `github.com/articos-research/llm-research-trap`.

References

Daman Agarwal et al. LLMs get lost in multi-turn conversation. *arXiv preprint arXiv:2505.12567*, 2025.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Ahmad Bilal. Grounded simulation: Validating AI-generated UX research against real user studies. In *arXiv preprint arXiv:2604.XXXXX*, 2026.

Merim Bilalić, Peter McLeod, and Fernand Gobet. Why good thoughts block better ones: The mechanism of the pernicious Einstellung (set) effect. *Cognition*, 108(3):652–661, 2008.

Nathan Crilly. Fixation and creativity in concept development: The attitudes and practices of expert designers. *Design Studies*, 38:54–91, 2015.

Nigel Cross. Expertise in design: An overview. *Design Studies*, 25(5):427–441, 2004.

Kees Dorst and Nigel Cross. Creativity in the design process: Co-evolution of problem–solution. *Design Studies*, 22 (5):425–437, 2001.

Karl Duncker. On problem-solving. *Psychological Monographs*, 58(5):i–113, 1945.

David G. Jansson and Steven M. Smith. Design fixation. *Design Studies*, 12(1):3–11, 1991.

Mosh Levy, Amir Jacoby, and Yoav Goldberg. Same task, more tokens: the impact of input length on the reasoning performance of large language models. *arXiv preprint arXiv:2402.14848*, 2025.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024.

Abraham S Luchins. Mechanization in problem solving: The effect of Einstellung. *Psychological Monographs*, 54 (6):i–95, 1942.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2023.

Horst W. J. Rittel and Melvin M. Webber. Dilemmas in a general theory of planning. *Policy Sciences*, 4(2):155–169, 1973.

Leonard Salewski, Stephan Alaniz, Isabel Rio-Torto, Eric Schulz, and Zeynep Akata. In-context impersonation reveals large language models' strengths and biases. In *Advances in Neural Information Processing Systems*, volume 36, 2024.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Caleb Schulhoff, et al. The prompt report: A systematic survey of prompting techniques. *arXiv preprint arXiv:2406.06608*, 2024.

Murray Shanahan, Kyle McDonell, and Laria Reynolds. Role play with large language models. *Nature*, 623(7987): 493–498, 2023.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, et al. Towards understanding sycophancy in language models. *PNAS Nexus*, 2024.

Herbert A. Simon. Rational choice and the structure of the environment. *Psychological Review*, 63(2):129–138, 1956.

Derek Tam and Tianyi Zhao. Playing pretend: Examining the effect of role prompting on LLM factual accuracy. *arXiv preprint arXiv:2504.01370*, 2025.

Derek Tam, Tianyi Zhao, Naomi Saphra, and Moran Shlain. PRISM: Expert persona prompting damages accuracy in medical question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

Jennifer Wiley. Expertise as mental set: The effects of domain knowledge in creative problem solving. *Memory & Cognition*, 26(4):716–730, 1998.

Wenda Xu, Guanglei Xu, and Oleg Rokhlenko. Pride and prejudice: LLM amplifies self-bias in self-refinement. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Yining Zhou et al. The price of format: An empirical study of LLM response diversity. *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025.