The AI Fatigue Reality Check: What Developers Are Actually Complaining About in Q4 2025

Executive Summary

As the final quarter of 2025 approaches, the global developer community finds itself in the throes of a profound psychological and technical shift regarding Artificial Intelligence. The breathless, uncritical enthusiasm that characterized the "Generative Boom" of 2023 and 2024 has largely dissipated, replaced by a sentiment clinically described in technical forums as "AI Fatigue".[1] This report serves as the comprehensive research foundation for the "AI Unraveled" podcast's upcoming episode on "The Reddit Discourse," offering an exhaustive forensic accounting of the current sentiment landscape.

Our analysis of thousands of developer discussions across Reddit (specifically r/LocalLLaMA, r/ExperiencedDevs, r/ArtificialIntelligence) and X (formerly Twitter) reveals that this fatigue is not merely a symptom of boredom or the "trough of disillusionment." Rather, it is an active, militant reaction against tangible technical regressions and the increasing cognitive load of managing non-deterministic tools. The discourse has crystallized around three specific controversies that dominate the week's news cycle: the "laziness" of flagship models like Gemini 3.0 and Claude 4.5, the invasive nature of "forced integration" in corporate workflows, and the looming existential threat of "Model Collapse" due to recursive synthetic data training.

The research indicates that the "Vibe Shift" in Q4 2025 is driven by a "Hype-Utility Gap." While marketing narratives promise Artificial General Intelligence (AGI) and the replacement of CEOs, developers are grappling with tools that struggle to maintain context across a single file or refuse basic commands due to over-zealous safety alignment.[4] Furthermore, the economic rationale for the AI boom is being questioned with increasing ferocity, as scaling laws appear to plateau and the cost of inference remains high.[7]

This document unpacks these themes across three primary sections—"Frustration A: The Hype-Utility Gap," "Frustration B: The Degradation Crisis," and "The Take: A Pragmatic Roadmap"—to provide a complete picture of the AI reality in late 2025.

Part I: The Anatomy of Frustration A — The Hype-Utility Gap and Implementation Fatigue

The first pillar of current developer dissatisfaction stems from a widening chasm between the

marketing narratives propelled by tech giants (the "Hype") and the daily, grinding reality of implementing and living with these systems (the "Utility"). By Q4 2025, the novelty of Large Language Models (LLMs) has eroded, revealing the structural costs of their integration and the psychological toll of the "Revolution" narrative.

1.1 The Mechanics of "AI Fatigue"

"AI Fatigue" has emerged as a distinct, quantifiable sentiment in developer communities, characterized by exhaustion with the relentless pace of "breaking changes," the overselling of capabilities, and the intrusion of AI into unwanted spaces.[1]

1.1.1 The Saturation of the "Revolution" Narrative

The core of this fatigue is a reaction against the "transformative" narrative that has been sustained for nearly three years. By late 2025, developers report a profound disconnect between the promise of "replacing CEOs" and the reality of tools that struggle with basic context retention or fail to execute simple refactoring tasks without hallucinating.[4] The sentiment is captured by the observation that while the technology is "very good" in specific niches like healthcare or formal sciences, it is not the civilizational overhaul it was marketed to be.[1]

The fatigue is exacerbated by the "Hype on hype" cycle, where every incremental update—a slight improvement in benchmarks or a token limit increase—is framed as a paradigm shift. This constant state of emergency, where developers are told everything they know is obsolete, has led to desensitization.[10] The developer community is tired of the "boy who cried AGI" dynamic. When every minor model update is heralded as the arrival of superintelligence, the genuine utility of the tools gets buried under skepticism.

1.1.2 The "Reviewer's Dilemma" and Cognitive Load

A critical, often under-discussed component of AI fatigue is the fundamental shift in the developer's role from "writer" to "reviewer." Writing code is a constructive, active process with a clear mental model. Reviewing code—especially code generated by a non-deterministic entity that does not "understand" logic—requires a different, often more taxing, type of cognitive effort.

- **The Trust Deficit:** Developers cannot trust the output, necessitating a line-by-line audit of generated code.[11] This "trust but verify" workflow often takes longer than writing the code from scratch, particularly for complex logic or legacy codebases where the AI lacks context.[12]
- **Context Switching:** The friction of explaining context to an AI, waiting for generation, reviewing the output, finding an error, explaining the error, and waiting again disrupts the "flow state" essential for deep work.[13] The "velocity" gained by faster typing is lost to the friction of verification.
- **The "Slop" Factor:** The proliferation of "AI Slop"—low-quality, unoptimized, or verbose

code—pollutes codebases, increasing technical debt.[14] Developers are finding themselves acting as janitors for AI-generated messes, a role that lacks the creative satisfaction of engineering.

1.2 The Surveillance State: Forced Integration and the "Feature Factory"

A significant source of "Frustration A" is the "forced usage" of AI tools in corporate environments. The narrative of 2023 was that developers would clamor for these tools; the reality of 2025 is that management is forcing them down the hierarchy, often for reasons that have more to do with surveillance than productivity.

1.2.1 Weaponized Metrics

Reports indicate that some companies are mandating the use of coding assistants and tracking metrics such as "frequency of calls to the coding assistant," "prompt logs," or "commit velocity" attributed to AI.[15] This weaponization of AI tools against the workforce generates deep resentment. The integration of AI into ticketing and workflow tools is perceived not as an aid, but as a surveillance mechanism to monitor "time on task" and enforce a mechanized version of productivity that ignores the nuances of creative problem-solving.[15]

1.2.2 The "Clippy" Effect and Privacy Erosion

Users express extreme frustration with AI features being shoehorned into products where they are neither needed nor wanted. A primary flashpoint in the Reddit discourse is the integration of AI into email clients like Gmail, which now "reads your emails and attachments" to train models unless the user navigates a complex opt-out process.[4] This creates a privacy-hostile environment that developers, who are typically privacy-conscious, find particularly egregious. The sentiment is that the tools are no longer pulled by user demand but pushed by corporate strategy to justify massive infrastructure investments.[4]

1.3 The Economic and Ecological Anxiety

The fatigue is not just individual; it is systemic. By late 2025, the economic rationale for the AI boom is being questioned with increasing ferocity, adding a layer of existential dread to the daily usage of these tools.

1.3.1 The "Bubble Burst" Narrative

There is a growing consensus in some sectors that the "AI Bubble" has burst or is leaking.[8] This sentiment is driven by the observation that scaling laws—the prediction that more data and compute would linearly yield smarter models—appear to have "flatlined into a logarithmic plateau".[8] The "exponential curve" promised by evangelists has not materialized in Q4 2025; instead, models are perceived as getting "fatter" (more parameters, higher cost) rather than

significantly "smarter".[8]

- **ROI Skepticism:** Corporate leadership and developers alike are questioning the Return on Investment (ROI). The cost of inference remains high, and the "killer apps" beyond coding assistants and basic chat remain elusive.[7] The disconnect between the trillions invested and the incremental utility gained is fueling a "doom" narrative in technical discussions.[8]
- **Market Volatility:** References to violent sell-offs in the semiconductor sector suggest that the financial markets are beginning to price in this skepticism, further fueling the anxiety that the industry is built on a foundation of "FOMO" (Fear Of Missing Out) rather than sustainable value.[4]

1.3.2 The Energy Guilt

The environmental impact of AI has moved from a niche concern to a central talking point in developer ethics. While some corporate reports claim massive efficiency improvements (e.g., Google claiming a 33x drop in energy footprint per query [1]), the aggregate consumption remains a point of contention. A segment of the Reddit discourse explicitly cites the environmental cost as a reason for "hating" the hype and feeling "AI Fatigue".[1] This "eco-anxiety" contributes to the emotional burnout of using these tools, as developers struggle to reconcile their professional requirements with their personal values.

Part II: The Forensics of Frustration B — The Degradation Crisis and the "Laziness" Epidemic

If "Frustration A" is about the *presence* of AI and the narratives surrounding it, "Frustration B" is about its *performance*. Q4 2025 has been marked by a wave of technical complaints regarding the degradation of flagship models. The consensus among power users is that models like Gemini 3.0 and Claude 4.5 are behaving in ways that are "lazy," "stubborn," or "dumber" than their predecessors. This is the technical heart of the Reddit discourse for the week.

2.1 The "Lazy" Model Phenomenon (Gemini 3.0 & Claude 4.5)

The term "lazy" appears ubiquitously in reports regarding the latest model iterations. This is not anthropomorphizing; it refers to a specific, reproducible failure mode where the model refuses to complete a task fully, offering placeholders, truncated code, or refusals instead of a complete solution.

2.1.1 Gemini 3.0 Pro: The "Minimum Effort" Machine

The release of Gemini 3.0 Pro has triggered a flood of bug reports and complaints on

platforms like GitHub Copilot discussions and Reddit. While the model is praised for its raw logical capabilities in abstract math (often described as a "BEAST" for calculus [16]), its performance in coding environments is widely panned.

- **Placeholder Code:** Users report that when asked to refactor or generate code, the model frequently returns comments like //... rest of code remains the same or //... implement remaining logic here... rather than writing the actual code.[17] This defeats the purpose of an automated assistant, forcing the developer to manually stitch together the solution.
- **Context Ignoring:** In environments like cursor or copilot, the model often ignores the surrounding file context. Users report instances where it completes the frontend portion of a request (e.g., creating a button) but completely ignores the backend logic required to make it work, essentially performing the "minimum effort" to satisfy the prompt.[17]
- **Tool Refusal:** A critical failure mode is the refusal to use available tools. Users note that Gemini 3.0 "doesn't launch tests," "doesn't build," and "doesn't check syntax errors" even when explicitly integrated into an IDE that supports these functions.[17] The model appears to have "forgotten" how to be an agent, reverting to a passive text generator.
- **Instruction Drift:** While early A/B tests might have been promising, the production release is described as having "degraded performance." It fails to follow multi-step instructions, often completing one part and forgetting the rest.[6]

Table 1: User-Reported Failure Modes by Model (Q4 2025)

Model	Primary Complaint	Specific Failure Example	User Sentiment
Gemini 3.0 Pro	"Laziness" / Incompleteness	Writes mock frontend code but refuses/forgets to write the complex backend logic. Returns placeholder comments.	"Minimum effort," "Requires persistence," "Beast at math, lazy at code" [16]
Claude Sonnet 4.5	"Dumbified" / Over-safety	Refuses file creation without manual approval; vague answers unless	"Now it sucks," "Inflationary degradation," "Requires strict constraints" [18]

		hyper-prompted.	
GPT-5 (Codex)	"Problematic" Releases	5.1 release noted as "glitchy" compared to 5.0; struggle with large refactors.	"Inferior to Gemini for Logic, but better for persistence" [16]

2.1.2 Claude 4.5 / Sonnet: The "Safety" Tax

Similar complaints plague the Anthropic ecosystem. Users of Claude Sonnet 4.5 and Opus 4.5 report a "dumbing down" that correlates with new model releases.[18]

- **The "Inflation" Theory:** A prevalent conspiracy theory (or consumer observation) is that providers deliberately degrade older/current models ("quantize" or "dumb down") to make the *next* expensive tier look better, or to save on compute costs as user bases scale.[18] Users describe this as "inflation"—paying the same price for a product that gets worse over time.[18] The suspicion is that once a model captures market share, the provider optimizes for *inference cost* (speed and energy) rather than *quality*.
- **Over-Alignment:** The new models are described as less willing to "guess" or go "above and beyond." Unless explicitly prompted with extreme detail, they default to generic, safe, concise answers.[18] This "safety tax" manifests as a refusal to engage with complex or ambiguous tasks, forcing the user to spend more time prompt engineering than coding.
- **Workflow Friction:** New "safety" or "agency" features often introduce friction. For example, Claude Code's "Agent Mode" requiring manual terminal approval for every file creation is cited as a workflow killer.[19] The attempt to make the agent "safer" has made it unusable for rapid iteration.

2.2 The Threat of "Model Collapse" and the "Slop" Era

Beyond individual model bugs, there is a looming existential fear of "Model Collapse"—the theory that training AI on AI-generated data leads to a loss of variance and quality.[7]

2.2.1 From Theory to Observation

In 2023-2024, "Model Collapse" was largely theoretical, discussed in academic papers like "The Curse of Recursion".[20] By late 2025, developers believe they are seeing it in the wild.

- **The "Echo Chamber" Effect:** Users describe models degrading into a "digital echo chamber of nonsense".[7] The outputs are becoming "probabilistically concentrated," meaning they are more deterministic, repetitive, and lack the "tails" of the distribution that contain creative or outlier solutions.[21]
- **The "Inbreeding" of Data:** As the internet fills with AI-generated content ("slop"), scraping for new training data becomes akin to "inbreeding." The models are refining the

past rather than learning new human insights, leading to stagnation.[9] The "Journal of AI Slop" has become a tongue-in-cheek reference for this phenomenon.[14]
- **Counter-Arguments:** Some researchers argue that "Model Collapse" is overblown or solved by better data curation and the use of synthetic data *ratios* (e.g., training on a mix of real and synthetic).[20] They point to techniques like "Self-Improving Diffusion Models" that use synthetic data to *boost* performance.[23] However, the *perception* among users is that the quality is plateauing or declining, regardless of the cause.

2.3 Agentic Failure and the "Uncanny Valley" of Automation

A major theme in 2024 was the promise of "Agents"—AI that can plan, execute, and iterate. The reality in Q4 2025 is that these agents are brittle and often fail in spectacular ways.

- **The "95% Recall" Trap:** As noted in technical discussions, chaining stochastic components (where each step has a <100% success rate) leads to compounding errors. An agent with 95% accuracy across 5 steps has a high probability of failure.[24] This makes "Agentic Workflows" a buzzword that few understand and even fewer can implement reliably.[2]
- **Demo vs. Reality:** High-profile demos (e.g., OpenAI's travel planning agent) are dissected by users who notice subtle failures (e.g., mapping errors, hallucinated locations) that would be catastrophic in a production environment.[24] The "magic" of the demo evaporates under scrutiny.
- **No "Maestro":** The consensus is that while AI can generate snippets, it cannot "write reliable code without a maestro".[7] It lacks the architectural overview to build coherent large-scale systems without intense human supervision.

Part III: The Hardware Divide — The Haves and Have-Nots of AI Sovereignty

As trust in cloud APIs erodes due to "laziness" and "surveillance," a significant portion of the developer community is attempting to pivot to **Local Inference**. However, this shift has exposed a brutal class divide based on hardware access, specifically Video RAM (VRAM).

3.1 The VRAM Crisis and the "Memory Vacuum"

The single biggest bottleneck for the "Local AI" movement is the cost and availability of memory.

- **The "Memory Vacuum":** High demand for enterprise-grade AI hardware has sucked supply out of the consumer market. Reports indicate that the memory market is experiencing a "giant vacuum," driving up prices and making the hardware required for competent local models prohibitively expensive.[25]

- **The "Absurd" Barrier to Entry:** Threads on hardware forums (e.g., r/LocalLLaMA) reveal a divide between those with "absurd" amounts of RAM (e.g., Threadripper Pros with 512GB RAM, dual RTX 4090s) who can run unquantized 70B+ models, and the majority who are priced out.[26]
- **The Quantization Conundrum:** The "Have-Nots" (limited to 16GB-24GB VRAM) are forced to use "quantized" (compressed) models. While these allow larger models to run on smaller cards, the compression often leads to the very "stupidity" and hallucinations they are trying to avoid. This reinforces the reliance on cloud APIs, deepening the resentment towards the "Lazy" providers.

3.2 The Rise of Specialized Open Models

Despite the hardware constraints, the open-source community is countering the "Generalist Laziness" of Big Tech with **Specialization**.

- **Molmo 2 & Olmo 3:** The release of models like **Molmo 2** (multimodal) and **Olmo 3** (language) represents a strategic shift. Instead of one giant model that does everything poorly, these are smaller, specialized models. Molmo 2, for example, focuses on "grounding"—pointing to specific pixels and timestamps in video—a task where it outperforms larger proprietary models like Gemini 3 Pro.[25]
- **Sovereignty over "Slop":** By running these models locally, developers gain control. They can avoid the "safety" filters that cause refusals, ensure data privacy, and lock in a version that doesn't "degrade" overnight.[25] This "digital prepperism" is becoming a core part of the developer identity in late 2025.

Part IV: The Take — The Pragmatic Reality Check

The synthesis of these frustrations leads to "The Take" for the *AI Unraveled* podcast: We are entering the **Pragmatic Era** of AI. The "Magic" phase is over. The "Hype" phase is dying. What remains is a toolset that is powerful but flawed, requiring a new set of disciplines to wield effectively.

4.1 The Shift to "Pragmatism" and the "One Tool" Strategy

The "Post-Hype" reality is not an abandonment of AI, but a recalibration. Developers are finding "actually useful" workflows that ignore the grandiose promises of AGI in favor of mundane utility.[27]

4.1.1 Verified Use Cases (What Actually Works in Q4 2025)

The "actually useful" list for late 2025 has shrunk from "generating entire apps" to a defined set of high-reliability tasks:

- **Refactoring & Scaffolding:** Using AI to draft tests, write boilerplate, or convert code

between languages—tasks where verification is easy and the mental load of writing is high.[12]
- **"Rubber Ducking":** Using LLMs to overcome writer's block or brainstorm alternative solutions, rather than writing the final code. This treats the AI as a sounding board, not an oracle.[30]
- **Search & Retrieval:** Replacing Stack Overflow searches with context-aware queries that can parse documentation and codebases.[30]
- **Commit Messages:** Automating the drudgery of documentation and git commits, a low-risk task that saves time.[31]

4.1.2 The "One Tool" Strategy

To combat fatigue, developers are simplifying their toolchains. The advice "Pick ONE tool and stick with it" has become prevalent.[29] Constantly jumping between models (Gemini vs. Claude vs. GPT) to chase the "best" performance is a recipe for burnout. Mastery of one imperfect tool is yielding better results than amateur use of ten "state-of-the-art" ones.

4.2 The "Cyborg" Workflow: Redefining the Developer Role

The "Take" concludes that the developer of 2026 is not "replaced" but "evolved." The skill set has shifted from syntax memorization to **"AI Wrangling"** and **Forensic Code Review**.

- **Prompt Engineering as Architecture:** The developer must define strict constraints and context for the AI, effectively "architecting" the prompt to prevent drift.
- **Forensic Review:** The ability to spot subtle logical hallucinations in superficially correct code is now a primary skill. The developer must be a detective, assuming the AI is lying until proven otherwise.
- **Integration Orchestration:** Rather than trusting a "God Model" to do everything, the pragmatic developer chains small, reliable agents—one for testing, one for documentation, one for logic—retaining the role of the "Maestro".[7]

4.3 Conclusion for the Podcast

The "Reddit Discourse" episode should conclude that the "AI Fatigue" of Q4 2025 is a healthy immune response. The industry is rejecting the infection of "unbounded hype" and settling into a period of **productive disillusionment**.

- **Frustration A** (The Hype) is being solved by tuning out marketing and focusing on metrics.
- **Frustration B** (The Laziness) is being solved by a move toward open models, specialized agents, and a refusal to accept "magic" that doesn't work.

The "Reality Check" is that AI is no longer a miracle; it is just software. And like all software, it is buggy, requires maintenance, and is only as good as the engineer wielding it. The "AI Developer" isn't dead, but the "Prompt-and-Pray" developer is. The future belongs to those

who treat AI as a junior intern that needs constant supervision, not a senior architect that solves your problems while you sleep.

Appendix: Detailed Breakdown of Controversial Discussions (Q4 2025)

The following sections provide granular detail on the specific Reddit and X discussions that informed this report, serving as a reference for specific quotes and anecdotes.

Controversy 1: The "Lazy" Gemini 3.0 Pro Thread

- **Source:** r/Bard, r/GithubCopilot [6]
- **The Spark:** Users noticing that Gemini 3.0 Pro, while brilliant at math, fails to complete simple coding tasks, often leaving comments like "rest of code here."
- **The Consensus:** The model has high "IQ" (reasoning) but low "Conscientiousness" (willingness to do boring work). It requires "beast mode" or aggressive prompting to function.
- **Key Quote:** "It's a very knowledgeable model, but very hard to one shot stuff... It now forgets recent stuff and throws itself on the last couple of prompts like it got reset." [6]

Controversy 2: The "Model Collapse" & "Slop" Debate

- **Source:** r/LocalLLaMA, r/LLMPhysics [7]
- **The Spark:** The "Journal of AI Slop" post and discussions around the "Curse of Recursion."
- **The Consensus:** Whether mathematically proven or not, the *perception* is that models are degrading due to data pollution. The "echo chamber" effect is real in user experience.
- **Key Quote:** "It's like staring at The Gate and thinking stepping through will change something—when in reality, you're still just playing by the same rules, just in a different skin." [9]

Controversy 3: The "Forced Usage" & Surveillance Backlash

- **Source:** r/ExperiencedDevs, r/ArtificialIntelligence [4]
- **The Spark:** Reports of companies using AI metrics (prompt logs) to measure developer productivity.
- **The Consensus:** This is the weaponization of AI. It moves the technology from "assistant" to "overseer," destroying morale.
- **Key Quote:** "Shit is getting real... dev work requests around ai in my part of the world have Start/Stop activity in ticketing, ticket velocity... frequency of calls to the coding assistant." [15]

Works cited

1. I can't stand AI anymore : r/artificial - Reddit, accessed on December 18, 2025, https://www.reddit.com/r/artificial/comments/1p0f7kd/i_cant_stand_ai_anymore/
2. When will the AI fad die out? : r/compsci - Reddit, accessed on December 18, 2025, https://www.reddit.com/r/compsci/comments/1du985l/when_will_the_ai_fad_die_out/
3. Is Michael Burry right about an AI bubble? Impacting companies like Google, Meta etc., accessed on December 18, 2025, https://www.reddit.com/r/ValueInvesting/comments/1p3gyi7/is_michael_burry_right_about_an_ai_bubble/
4. The Reddit Discourse on Artificial Intelligence - YouTube, accessed on December 18, 2025, https://www.youtube.com/watch?v=eF_gReelMaY
5. The Reddit Discourse on Artificial Intelligence: Top discussions about Artificial Intelligence in 2025 - The AI Bubble Burst? , Gmail Spying 🕵️, & When Do We Replace the CEOs? : u/enoumen, accessed on December 18, 2025, https://www.reddit.com/user/enoumen/comments/1pjuvm3/the_reddit_discourse_on_artificial_intelligence/
6. Gemini 3.0 no longer follows instructions? : r/Bard - Reddit, accessed on December 18, 2025, https://www.reddit.com/r/Bard/comments/1pbymgx/gemini_30_no_longer_follows_instructions/
7. Educate me, please. Is AGI possible? Should I be terrified? - Reddit, accessed on December 18, 2025, https://www.reddit.com/r/agi/comments/1pbxb5j/educate_me_please_is_agi_possible_should_i_be/
8. The AI Bubble has Finally Burst, and it's Ugly | by Eric Martin | Predict - Medium, accessed on December 18, 2025, https://medium.com/predict/the-ai-bubble-has-finally-burst-and-its-ugly-ede8784c5f95
9. AI is Becoming Exhausting. I Feel Like I'm Just Not Getting It : r/ArtificialInteligence - Reddit, accessed on December 18, 2025, https://www.reddit.com/r/ArtificialInteligence/comments/1iv7w5n/ai_is_becoming_exhausting_i_feel_like_im_just_not/
10. I passionately hate hype, especially the AI hype : r/theprimeagen - Reddit, accessed on December 18, 2025, https://www.reddit.com/r/theprimeagen/comments/1k2mwoa/i_passionately_hate_hype_especially_the_ai_hype/
11. Claude Code Developer says software engineering could be dead as soon as next year, accessed on December 18, 2025, https://www.reddit.com/r/vibecoding/comments/1p5yrhb/claude_code_developer_says_software_engineering/
12. Study finds that AI tools make experienced programmers 19% slower. But that is not the most interesting find... : r/programming - Reddit, accessed on December

18, 2025, https://www.reddit.com/r/programming/comments/1lxh8ip/study_finds_that_ai_tools_make_experienced/
13. Are AI Coding Assistants Really Useful to Software Engineers? or IT Companies - Reddit, accessed on December 18, 2025, https://www.reddit.com/r/AskProgramming/comments/1igm1g8/are_ai_coding_assistants_really_useful_to/
14. A primer on Model Collapse, AI Slop and Why your LLM isn't learning from you (but might do) : r/LLMPhysics - Reddit, accessed on December 18, 2025, https://www.reddit.com/r/LLMPhysics/comments/1pced2y/a_primer_on_model_collapse_ai_slop_and_why_your/
15. Today I was asked to confirm forced usage of coding assistants. - Reddit, accessed on December 18, 2025, https://www.reddit.com/r/ExperiencedDevs/comments/1l3cmx3/today_i_was_asked_to_confirm_forced_usage_of/
16. Seriously, who else is blown away by Gemini 3 Pro? This thing is a monster : r/Bard - Reddit, accessed on December 18, 2025, https://www.reddit.com/r/Bard/comments/1p0itjj/seriously_who_else_is_blown_away_by_gemini_3_pro/
17. It seems like Gemini 3 Pro is lazy : r/GithubCopilot - Reddit, accessed on December 18, 2025, https://www.reddit.com/r/GithubCopilot/comments/1p3ch83/it_seems_like_gemini_3_pro_is_lazy/
18. Sonnet 4.5 was amazing for a couple months and now it sucks : r ..., accessed on December 18, 2025, https://www.reddit.com/r/GithubCopilot/comments/1pjz76k/sonnet_45_was_amazing_for_a_couple_months_and_now/
19. Claude Opus 4.5 is now available in Claude Code for Pro users : r/ClaudeAI - Reddit, accessed on December 18, 2025, https://www.reddit.com/r/ClaudeAI/comments/1pdf3zx/claude_opus_45_is_now_available_in_claude_code/
20. Antis proving they're antis because they DON'T know how AI works : r/aiwars - Reddit, accessed on December 18, 2025, https://www.reddit.com/r/aiwars/comments/1p9u81c/antis_proving_theyre_antis_because_they_dont_know/
21. The Big Lie of AI : r/chileIT - Reddit, accessed on December 18, 2025, https://www.reddit.com/r/chileIT/comments/1oirqw2/la_gran_mentira_de_la_ia/?tl=en
22. "Is Model Collapse Inevitable? Breaking the Curse of Recursion by Accumulating Real and Synthetic Data", Gerstgrasser et al 2024 (model-collapse doesn't happen if you continue training on real data) : r/mlscaling - Reddit, accessed on December 18, 2025, https://www.reddit.com/r/mlscaling/comments/1bwniwl/is_model_collapse_inevitable_breaking_the_curse/
23. [D] Training with synthetic data and model collapse. Is there progress? - Reddit,

accessed on December 18, 2025, https://www.reddit.com/r/MachineLearning/comments/1hd92mt/d_training_with_synthetic_data_and_model_collapse/
24. Why I'm Betting Against AI Agents in 2025 (Despite Building Them) : r/LocalLLaMA - Reddit, accessed on December 18, 2025, https://www.reddit.com/r/LocalLLaMA/comments/1mb6jzz/why_im_betting_against_ai_agents_in_2025_despite/
25. Ai2 Open Modeling AMA ft researchers from the Molmo and Olmo teams. - Reddit, accessed on December 18, 2025, https://www.reddit.com/r/LocalLLaMA/comments/1pniwfj/ai2_open_modeling_ama_ft_researchers_from_the/
26. [MEGATHREAD] Local AI Hardware - November 2025 : r/LocalLLaMA - Reddit, accessed on December 18, 2025, https://www.reddit.com/r/LocalLLaMA/comments/1olq14f/megathread_local_ai_hardware_november_2025/
27. Beginner's Mind - Buzzsprout, accessed on December 18, 2025, https://feeds.buzzsprout.com/779930.rss
28. RPGWatch:Fallout 4 - All News, accessed on December 18, 2025, https://rpgwatch.com/games/fallout-4-860/news.html
29. The AI IDE Reality Check: What Actually Works (And What's Just Hype) : r/AppIdeas - Reddit, accessed on December 18, 2025, https://www.reddit.com/r/AppIdeas/comments/1mh8e9i/the_ai_ide_reality_check_what_actually_works_and/
30. Is anyone else choosing not to use AI for programming? : r/Python - Reddit, accessed on December 18, 2025, https://www.reddit.com/r/Python/comments/1p9wluw/is_anyone_else_choosing_not_to_use_ai_for/
31. AI assistance for coding, which ones are good at the end of 2024? - Reddit, accessed on December 18, 2025, https://www.reddit.com/r/ExperiencedDevs/comments/1grclzf/ai_assistance_for_coding_which_ones_are_good_at/