v1: 31 May 2026

Commentary

The LLM Productivity Cliff: Threshold Productivity and AI-Native Inequality

Preprinted: 9 December 2025
Peer-approved: 31 May 2026

Qeios, Vol. 8 (2026)
ISSN: 2632-3834

Francesco Bisardi[1]

1. Independent researcher

We define the LLM productivity cliff as a threshold hypothesis: users and organizations with sufficient architectural literacy may achieve step-change productivity gains from using LLMs, while others may experience modest gains, no gains, or slowdowns. Drawing on 2025 evidence and newly emerging 2026 studies, we treat the LLM productivity cliff as a threshold hypothesis rather than a settled empirical law. The current literature supports heterogeneity in LLM productivity outcomes, but it does not yet prove a universal discontinuity across all domains. We operationalize this threshold as architectural literacy, not as a marginal prompt skill but as a qualitative shift toward decomposition, orchestration, and systematic validation. We identify boundary conditions that make cliffs more likely (task complexity, scaffolding, mental models) and develop a three-level framework of inequality at the individual, organizational, and market levels. We conclude with interventions to reduce capability disparities: embedding scaffolding in tool design, institutionalizing architectural literacy training, and encouraging equitable diffusion of architectural literacy and AI-native organizational practices.

Corresponding author: Francesco Bisardi, francescobisardi@gmail.com

1. Introduction

In a randomized study of experienced open-source developers, the latest generation of AI assistants made most participants slower on average; only a developer with substantial experience with AI assistants saw significant gains[1]. Large polls of US programmers reveal a split in which self-identified high-proficiency users report significantly bigger advances than their peers using identical tools[2]. In customer support, however, the pattern reverses: production-level AI assistants significantly benefit junior workers, while senior professionals find only minor or even negative effects[3]. These studies indicate that LLM productivity effects are not uniform. Similar tools might result in gains, stagnation, or slowdowns, depending on the user, task, and workflow environment. At the same time, benchmark progress and adoption indicators reveal that the technological frontier is rapidly evolving. Between 2023 and 2025, the confirmed solution rate of SWE-bench (Software Engineering Benchmark) increased from 4.4% to 71.7%, according to Stanford HAI. In September 2025, Dario Amodei revealed that "70, 80, 90% of the code written in Anthropic is authored by Claude"[4]. Industry-wide adoption data go in the same direction. According to the 2025 DORA Report, 90% of software development professionals use AI in key activities, with 65% indicating considerable reliance and 80% reporting increased productivity[5]. These metrics demonstrate rapid capability improvement and widespread developer adoption, but they do not provide direct evidence of net worker productivity. They contribute to the underlying puzzle: why do fast-improving LLMs result in substantial improvements for some users but modest increases, no gains, or slowdowns for others? The 2026 literature supports the case for productivity heterogeneity and explains the candidate mechanism: gains are determined less by seniority than by whether workflow structure is supplied by the user, entrenched in the system, or absent entirely.

Daniotti et al.[6] find AI-coding gains concentrated among senior developers; Gambacorta et al.[7] find them concentrated among juniors in an enterprise setting. These contradictions can be interpreted through a threshold lens, but they should not be treated as fully resolved. A threshold model explains why identical tools provide different results, but 2026 evidence suggests that the threshold changes depending on (1) how much scaffolding a workflow provides, (2) the user's verification effort, and (3) their prior experience. The performance differences are caused by a qualitative shift in work practice, from treating LLMs as conversational tools to designing workflows around their capabilities. Below the threshold, users engage models as extended auto-complete or search chats. Above the threshold, users adopt an engineering mindset: they decompose tasks, orchestrate tools and agents, structure context engineering, gate risks, and evaluate the outputs as a systematic process. Crossing this cliff generates discontinuous productivity gains: order-of-magnitude changes in both (1) task completion speed and (2) output quality[1][2][8].

1.1. Problem statement

Why do similar LLMs produce such different outcomes across users and firms? We argue that architectural literacy is the primary candidate mechanism: the capacity to break complex goals into model-tractable subtasks, orchestrate multi-step workflows, and validate outputs against external standards. However,

 doi.org/10.32388/5VAOLK.2

architectural literacy should not be treated as the sole explanation. What appears to be a threshold effect may also reflect selection bias, prior expertise, task composition, organizational support, training access, adjustment costs, or measurement noise.

1.2. Scope and positioning

We examine knowledge work where LLMs plausibly complement human labor at scale, focusing on software development, customer support, and labor-market outcomes. The framework fits best when tasks are complex, decomposable, iterative, and externally verifiable. It fits less well where validation is expensive, error tolerance is low, or output quality is hard to observe. Clinical medicine is a useful boundary case. Wang et al.[9] find that human-LLM collaboration shows some quality improvements, but the evidence remains uncertain and context-dependent; time efficiency does not reliably improve, and factual error rates remain high. In high-risk domains, the current productivity gains may be eaten by verification and accountability costs. Our contribution is not a new metric but an account of how and where these cliffs form, and what might be done about them. In this research, AI-native inequality refers to the unequal distribution of productivity gains, wages, organizational capacity, and market power that occurs when select individuals and organizations can redesign their work around AI systems while others can only access the same tools. The concept is not restricted to model access. It is about the uneven adoption of AI-native process capabilities such as decomposition, orchestration, validation, data integration, and organizational restructuring. We examine this inequality on three levels: individual performance and wages, organizational absorption capability and process redesign, and market concentration and returns to scale.

1.3. Contributions

1. **AI architectural literacy.** We introduce AI architectural literacy as a threshold construct distinct from domain expertise or prompt engineering, and propose it as a measurable construct that may help explain discontinuous productivity gains under specific task, workflow, and organizational conditions.
2. **Evidence synthesis.** We compile evidence that LLM productivity effects vary substantially with workflow conditions: open-ended, low-scaffold tasks require users to supply architectural literacy themselves, while high-scaffold systems embed parts of that literacy into the tool and lift less-experienced users.
3. **Three-level framework.** We connect individual thresholds, organizational capability building, and market concentration into a single structural account of AI-native inequality.
4. **Agenda to flatten the cliff.** We derive design and policy implications: scaffolding by design, literacy as infrastructure, and equitable diffusion.

2. Conceptual Framework

2.1. The Cliff

The LLM productivity cliff is proposed as a discontinuity in task performance. Below the threshold, additional effort yields little or negative return on complex work. Above the threshold, there is a qualitative shift in how the user works with models, producing large, stable gains. In a continuous learning curve, performance improves steadily as effort increases, while cliffs suggest something different: modest increases below a capability threshold followed by a step-change over it. The analogy is that progress is not always gradual. Just as some LLM abilities appear only after the model reaches a certain capability level[10], users may see limited gains until they change how they work with AI. The productivity jump comes not from using the tool more, but from using it differently: decomposing tasks, orchestrating workflows, and systematically validating outputs. For empirical purposes, the productivity cliff is a statistically visible change in the level or slope of net task performance after a user crosses a minimum capacity threshold in decomposition, workflow orchestration, and output validation. Net task performance encompasses not just speed and output volume but also quality, error rate, verification cost, and rework. No study from 2026 directly verifies the construct. As a result, we consider it a testable hypothesis that may be identified via repeated task panels, segmented regression, or change-point detection, rather than anecdotal perceptions or self-reported productivity.

Three patterns emerge across early field studies:

- **Non-monotonicity.** Early adoption can be net negative for experienced professionals when work is complex and scaffolding is weak[1].
- **Bimodality.** Outcomes cluster: a long tail of modest gains and a concentrated tail of large gains among high-proficiency users[2].
- **Inversions.** When systems provide scaffolding, beginners can surpass seniors who continue to adhere to outdated practices.[3].

Recent 2026 coding evidence contradicts any single story. Daniotti et al.[6] studied open-source GitHub Python code and found AI coding benefits are concentrated among experienced senior developers. Gambacorta et al.[7] find the opposite in enterprise: junior programmers gain more. The findings appear inconsistent; however, the boundary criteria stated in Section 2.3 help clarify the discrepancy. Daniotti et al. investigate open-source

contributions in which scaffolding is minimal and users must provide their own decomposition and validation practices. Experienced developers in structured codebases already have the decomposition and validation habits that constitute architectural literacy. Gambacorta et al. study an enterprise deployment with integrated tooling (CodeFuse), where organizational workflows, existing codebases, review processes, and tool integration likely supply more scaffolding for junior programmers. The pattern is the same underneath: productivity follows architectural literacy, regardless of how someone got it.

2.2. Architectural literacy as an engineering mindset

Architectural literacy is distinct from both prompting skill and domain expertise. Domain expertise is the capacity to judge whether an output is substantively correct. Prompting skill concerns how the user communicates with the model within a single interaction: specificity, constraint setting, context provision, and output formatting. Architectural literacy concerns how the user designs the full human-model-tool workflow: decomposition, sequencing, validation, and system integration across interactions. The three diverge in practice. A domain expert with strong prompts but no architectural literacy produces better individual outputs but still works one turn at a time, losing efficiency on complex multi-step tasks. A user with architectural literacy but limited domain expertise can design a pipeline that decomposes, routes, and validates but may miss substantive errors only domain knowledge can catch. Prompting skill alone cannot sustain quality across a complex project because it lacks both workflow structure and evaluative judgment. The productivity cliff is specifically an architectural literacy threshold: domain expertise and prompting skill improve outcomes within any level of practice, but neither produces the discontinuous shift from Level 2 to Level 3. Architectural literacy is the ability to use LLMs as part of a structured workflow, and five capabilities define this threshold:

1. **Decomposition.** Breaking ambiguous goals into model-agent-tractable subtasks and isolating judgment calls that must stay human.
2. **Workflow design.** Iterative agentic chains, checkpoints, and critique loops instead of one-shot prompting.
3. **Output evaluation.** Systematic validation tuned to model failure modes rather than ad hoc spot checks.
4. **System integration.** Binding models to data, tools, and agents so the unit of work is a workflow, not a chat turn.
5. **Adaptive mental models.** Updating assumptions about what models can and cannot do and routing work accordingly.

These five capabilities can be operationalized as behavioral indicators. A user demonstrates architectural literacy when they can: define the task boundary, decompose the work into model-tractable subtasks, provide structured context, choose an interaction pattern, verify outputs against external criteria, identify likely failure modes, and revise the workflow according to observed errors. This makes architectural literacy a workflow-level capability, not simply generic AI familiarity or prompt fluency.

Architectural literacy reorients practice from prompt optimization ('what to ask') to workflow and agentic design ('how to structure iterative human-model-agent interaction').

2.3 Boundary conditions: When cliffs are most likely to appear

Cliffs are most visible when three conditions align:

- **High task complexity.** Open-ended design, architecture, research, planning, or multi-module coding are cliff-prone; e.g., summarization and translation are not[11][12].
- **Low scaffolding.** Unstructured prompts or minimally instrumented tools tend to produce high variance in outcomes, whereas structured and constrained systems reduce this variance.
- **Misaligned mental models.** Experts anchored in legacy, UI-based workflows may misuse AI by under- or over-trusting model outputs, while novices using scaffolded tools can perform better by following systematized guidance.

This framework reconciles divergent findings across domains: open-ended coding tasks (high complexity, low scaffolding) exhibit cliff effects, while structured high scaffolding (customer support) shows novice gains[13].

2.4. Levels of practice

To make the threshold/cliff concrete, we use a three-level description of practice. This three-level model operationalizes the cliff[14]: the discontinuity happens in the transition from Level 2 (workflow integration) to Level 3 (architectural redesign)[1][15][2][8][13].

Three-Level Model:

Level / Skills	Capabilities	Productivity Impact	Evidence
Level 1: Surface Usage	Single prompts, obvious error detection, copy-paste workflows	-20% to +15% negative to modest gains	81% of devs in [1]
Level 2: Workflow Integration	Multi-step chains/workflow, context engineering, iteration, boundary recognition	+15% to +35% moderate, more consistent gains	66% “proficient” devs (34% gains)[2]
Level 3: Architectural Design	Task redesign, custom toolchains, programmatic APIs, orchestration, systematic validation	+38% to +200% potential step-change gains under favorable conditions	>50h dev [1], Anthropic (70–90% AI code), [8]

- **Level 1: Surface usage.** One-shot prompts, copy-paste, obvious error catching. Effects range from negative to small positive in complex tasks.
- **Level 2: Workflow integration.** Multi-step prompting, richer context, iterative refinement, basic limits awareness. Gains become consistent but remain bounded.
- **Level 3: Architectural design.** Task redesign, tool/agent orchestration, automated checks, data and API integration, repeatable pipelines creating agentic organizations[15]. Benefits become discontinuous.

Movement from Level 2 to Level 3 is the cliff, where the engineering mindset takes place and the distribution “splits.”

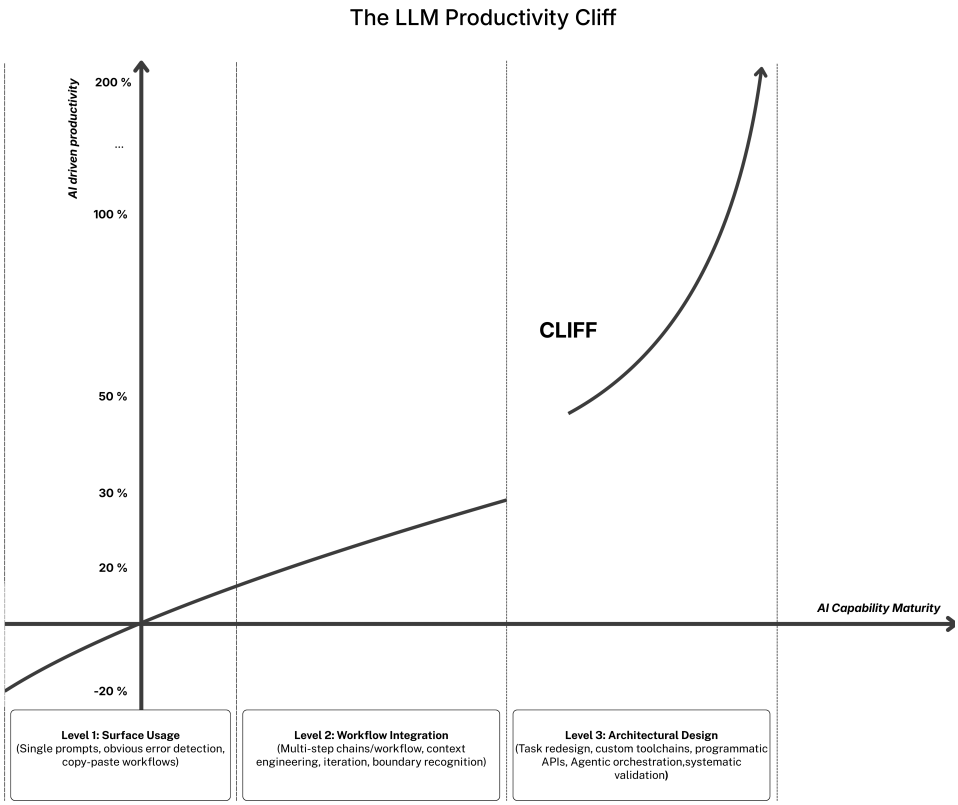

Figure 1. The LLM Productivity Cliff

The productivity ranges are not validated cutoffs but rather illustrative. Existing research suggests varied and nonlinear productivity effects, but precise thresholds have yet to be empirically proven. Future research should aim to determine these cutoffs using repeated task measurements and predefined decomposition, orchestration, and validation criteria.

doi.org/10.32388/5VAOLK.2

2.5. Relation to existing theory

Three theoretical strands help explain why threshold effects may arise:

- **Task-Technology Fit**[16]**.** The same model can be used either as a search-like tool or for structured, orchestrated reasoning, but only the latter suits complex tasks. Fit improves substantially when users redesign tasks to match the tool.
- **Threshold learning & ZPD (Zone of Proximal Development)**[17]**.** Scaffolding moves the boundary of what users can actually do; without it, the self-management and planning demands of LLM use exceed many users' zones of proximal development.
- **Post-routine automation.** When easy tasks are automated, the remaining work becomes more expert, raising the bar for meaningful contribution[18][19]. Under these conditions, productivity gains are more likely to appear as threshold effects than as smooth linear improvements.

2.6. From individual thresholds to AI-native inequality

Once defined this way, AI-native inequality can be analyzed as a cross-level mechanism. Micro-level thresholds may aggregate into organizational and market-level inequality through three channels:

- **Individuals.** Those who cross into Level 3 capture large, persistent gains; others plateau or decline on complex work.
- **Organizations.** Teams that rebuild processes around agents, data, and validation loops pull away from legacy operations; fixed costs and tacit know-how slow catch-up.
- **Markets.** Capability and value concentrate in AI-native startups that can compound workflow advantages; diffusion lags elsewhere.

This inequality mechanism is structural but not permanent: it reflects the current mismatch between the speed of AI capability deployment and the slower diffusion of architectural literacy across individuals, organizations, and markets. Ball et al.[20] show that GenAI competency and adoption readiness are shaped by operational skills, creative skills, attitudes toward AI, decision reliance, and self-efficacy. Wong and Qiu[21] show that guided AI use can improve later independent capability, whereas unrestricted use may improve immediate performance without producing the same learning gains. This evidence suggests that productivity inequality can widen when exposure, training, and scaffolding are uneven, but it can also flatten when institutions distribute workflow literacy and guided-use practices more broadly. At the macro level, productivity inequality is not only about unequal access to tools; it reflects unequal capacity to redesign workflows, organizations, and markets around AI[18][19][22]. These gaps persist not because architectural literacy is inherently scarce, but because the diffusion of workflow practices, training infrastructure, and organizational redesign lags behind tool availability. The risk of inequality is already visible in early evidence, but it is addressable through the interventions outlined in Section 6.

3. Mechanism: A Multi-Level Account of Threshold Effects

We propose a multi-level model in which individual capability thresholds, organizational design choices, and market structure interact to produce distributional bifurcation. We theorize three nested levels that capture this mechanism.

3.1. Individual level: Skill acquisition thresholds

At the individual level, threshold effects may stem from cognitive and procedural factors. The examined evidence supports architectural literacy as a plausible hidden mechanism, but current research hardly tests it directly. Architectural literacy rewards whoever possesses it, regardless of career stage: senior developers gain when they already practice decomposition and validation[6], junior developers gain when scaffolded tools confer these practices on them[7][3], and experienced developers without these practices slow down[1]. The relevant threshold is the presence of architectural work practices, not experience level per se. These results are consistent with a threshold model: performance clusters bimodally, with gains concentrated among users exhibiting architectural practices[1][2].

Users who reach this stage move from treating the model as a conversational partner to treating it as a component in a system (designing context, iteration, and validation loops that the model executes within). Ganuthula[23] frames this shift as Capital Structure Transformation: AI compresses the efficient scale of production, allowing individuals to substitute for small teams when they possess sufficient architectural literacy. This pattern diverges from gradual skill-acquisition models and aligns with threshold learning theories[14], where conceptual advancements yield nonlinear enhancements in performance.

3.2. Organizational level: Capability development

At the meso level, organizations exhibit the same bifurcation. Firms that are AI-native[24] (designed around integrated AI pipelines, contextual data, and agentic workflows) pull ahead of incumbents still layering AI onto

legacy systems. Wang X. et al.[8] provide direct evidence that organizational design, not mere tool access, governs AI spillovers: in lean, AI-native firms, AI adoption generates additional firm-level productivity gains beyond directly affected tasks, with spillovers two to three times larger than those associated with traditional IT firms, even after accounting for firm size. Brodzicki's[25] model demonstrates that the high fixed costs of AI implementation, combined with modest average impacts, lead to market concentration, allowing AI-native enterprises to gain competitive advantages and form oligopolistic structures. Empirical enterprise studies[26] indicate that although roughly 90% of professionals report using AI, only a small minority achieve substantial productivity gains, largely because organizational infrastructure and team structures have not been redesigned for AI systems. Productivity gaps may reflect differences in system design capability rather than tool access[26][27]. Organizations that redesign processes and integrate AI into core infrastructure (connecting model outputs to databases, APIs, and decision/orchestration agentic systems) may experience superlinear returns, though empirical validation of scaling dynamics remains limited[28].

3.3. Market level: Structural concentration

At the macro level, the cliff manifests as AI-native inequality: value concentrates in actors that can cross capability thresholds at scale. Market bifurcation is already emerging. The WEF *Future of Jobs 2025* report indicates that roughly 60% of enterprises now demand some form of AI literacy, a pattern consistent with increasing concentration of opportunities and returns among AI-capable firms and workers (the cliff). Autor and Thompson[18] demonstrate that the automation of ordinary operations increases the required skill for the remaining tasks. As LLMs automate routine cognitive tasks, residual work requires higher-order skills[18]. Mid-skilled workers may be displaced if they cannot shift to supervisory or orchestration jobs;[18][29] this will concentrate gains among those who complement rather than compete with AI. Fan[19] formalizes a similar mechanism: AI-related productivity gains cluster around certain skills, making it harder for workers to switch roles and increasing wage gaps. Calibrated macro models show a similar pattern: AI may narrow some wage differences but still raise wealth inequality because high-skilled AI workers and capital owners capture most of the gains from AI adoption[30]. Johnston and Makridis[22] experimentally demonstrate that salary increases disproportionately benefit younger, highly educated, AI-proficient workers in areas where AI serves as a complement rather than a substitute. The cliff hypothesis predicts that individual capability thresholds aggregate to organizational and market-level inequality through three reinforcing mechanisms: skill concentration, capability lock-in, and returns to scale in AI-native workflows.

4. Evidence: Empirical Patterns of Divergence

The evidence reviewed below varies in strength. Quasi-experimental and field-experimental studies provide the strongest support for heterogeneous productivity effects. Surveys and industry reports are useful for identifying adoption patterns and perceived gains, but they are weaker evidence for causal thresholds. Preliminary empirical evidence from individuals, organizations, and markets aligns with the cliff hypothesis; however, causal interpretation is limited. Research indicates mixed productivity results, with performance aggregating at both high and low extremes instead of following a normal distribution.

4.1. Individual productivity heterogeneity

A survey of ~2,000 U.S. developers[2] shows a 2.4× productivity differential between high- and low-proficiency users: 34% vs. 14% self-reported improvement. Becker et al.[1] provide experimental evidence: even among seasoned open-source programmers, only those with deep prior assistant experience achieved speedups; the rest slowed down. Brynjolfsson et al.[3] observe an inverted pattern in customer support. Junior and lower-skilled customer-service workers who used a generative AI assistant saw significant improvements in issue resolution, but experienced agents saw only minor advances. Brynjolfsson et al.[3] take this as evidence that the system collected and distributed expert practices, helping novices go more quickly along the learning curve. Large-scale field studies with almost 5,000 industrial developers found that AI code helpers enhanced work completion rates by roughly 26% on average, with less-experienced programmers seeing bigger benefits than senior engineers[31]. Together, these findings do not support a straightforward U-shaped experience curve. They suggest a conditional threshold pattern in which less-experienced workers benefit when the environment provides strong scaffolding, and sophisticated users profit when they can supply the missing architecture themselves. Mid-career professionals report smaller productivity improvements than novices or experts[2], which supports the concept that intermediate skill levels incur higher adaptation costs. Wang et al.[32] found that programmable, multi-agent processes resulted in 88% faster job completion and 90% reduced costs compared to prompt-response interactions. The performance increase appears to depend on workflow architecture rather than on isolated prompt behavior alone. The strongest 2026 coding evidence suggests heterogeneity. According to Daniotti et al.[6], AI-generated code led to a 3.6% increase in quarterly online code contributions, primarily among senior developers. Gambacorta et al.[7] find a larger enterprise effect: CodeFuse increased code output by more than 50%, while task-completion measures increased by 22%, but statistically significant gains were concentrated among junior and entry-level staff. The contrast suggests that the same

broad technology can produce different subgroup effects depending on context. The cliff hypothesis is therefore most defensible when framed as a conditional threshold model rather than a typical experience curve.

4.2. Organizational capability divergence

Firm-level dynamics mirror the individual cliff. AI-native startups, structured around agentic integration and feedback loops, are scaling faster than legacy enterprises still layering AI on top of older infrastructure. At the startup level, Shi[24] reports that a subset of AI-native firms are achieving, in some cases, $1M revenue/employee, crediting this to intensive AI automation, senior technical talent, high-value contracts with outcome-based pricing, rapid iteration, and highly AI-scalable, automated go-to-market processes run by small AI generalist teams rather than narrow specialists. While some of the underlying data are not peer-reviewed, they illustrate how aggressive workflow redesign and automation can shift the productivity frontier for small teams. Conversely, Nanda et al.[27] reported that 95% of generative AI pilots at corporations are failing, with the core issue not being model performance but a "learning gap": organizations lack experience integrating AI into workflows, and generic conversational tools such as ChatGPT tend to stall in enterprise settings because they are not integrated with firm-specific workflows and data and therefore cannot adapt to local processes. Field experiments in large enterprises show similar heterogeneity in realized gains: workers with Copilot access spend less time on routine solo tasks, but usage intensity and benefits vary sharply across firms[33]. Brodzicki[25] predicts oligopolistic drift due to high setup costs and limited knowledge diffusion. Using an inter-firm mobility network of AI workers among more than 16,000 U.S. companies, Wang X. et al.[8] showed that productivity spillovers from AI talent are conditional on organizational context: hiring from lean startups yields transferable AI generalists and outsized productivity gains, whereas hires from traditional, multi-layered incumbents transmit little advantage. Doddapaneni et al.[26] empirically confirm that enterprises with legacy architectures capture little of the theoretical productivity gain, even with high AI adoption rates, pushing firms toward strategic partnerships or acquisitions to obtain generative-AI capabilities[27]. In short, access parity masks systemic inequality: the productivity frontier shifts from who has the tools to who can rebuild processes around agentic and AI-native workflows.

4.3. Labor market restructuring

At the labor market level, multiple 2025 studies detect polarization consistent with the cliff hypothesis[34][35][22][29]. Using high-frequency ADP payroll data, Brynjolfsson, Chandar, and Chen[34] find that employment in highly AI-exposed jobs fell sharply for entry-level workers (a 6–13% drop among 22–25-year-olds) after ChatGPT's release, while mid- and senior-level workers remained stable or gained. Johnston and Makridis[22] similarly find that wage gains in high-AI-exposure sectors are concentrated among educated, AI-fluent employees. Dominski and Lee[29] show that as model capabilities expand, tasks within affected occupations are reclassified from "human" to "AI-capable," raising unemployment and reducing hours for those unable to complement models. In contrast, Humlum and Vestergaard[36] detect minimal average wage change across the Danish labor market, implying that national aggregates hide strong within-sector variance: the gains are captured by a narrow subset of workers and firms. The labor evidence parallels the micro results: only those who can operate at the architectural level—supervising, integrating, and verifying AI outputs—retain or increase value up to a 23% wage increase[37]. The rest face gradual or sudden displacement.

4.4. Cross-level synthesis

The empirical picture is consistent with the threshold theory:

- **Micro level:** Productivity outcomes vary sharply across individuals. The gains scale only after architectural literacy is achieved[1].
- **Meso:** Firms integrating or redesigning AI into core workflows report higher productivity than incumbents, consistent with capability-based concentration[25][26].
- **Macro:** Labor market studies find wage gains concentrated among AI-complementary roles[22][37], consistent with skill-biased technical change[18]. At the same time, cross-country experiments suggest that generative AI can narrow productivity gaps between high- and low-wage regions by 'leveling up' workers in lower-income countries[38].

Early evidence across several levels of analysis supports the threshold hypothesis, indicating that productivity gains are concentrated among individuals, firms, and workers who redesign tasks and workflows to capitalize on AI capabilities, while others face stagnation or decline.

5. Methodological Considerations

Until such data exist, the LLM productivity cliff should be treated as a structural hypothesis: it is supported by convergent early evidence but requires validation through sustained and transparent measurement. A direct test of the productivity cliff would require repeated task performance data across users with varying levels of AI workflow literacy. One design would assign participants comparable complex tasks over multiple iterations,

adjust the workflow scaffolding they receive, assess architectural literacy using the previously described rubric, and have blinded human evaluators analyze the outputs. The cliff hypothesis would be supported only if performance shows a statistically detectable change in level or slope after users cross predefined decomposition, workflow-design, and validation thresholds. Studies such as Becker et al.[1] and Brynjolfsson et al.[3] rely on short-term field or workplace trials, typically lasting weeks. They reveal heterogeneity but cannot yet establish persistence or causality. Survey data[2] depend on self-reported proficiency and perceived gains, which may conflate skill with confidence. The current empirical base is therefore temporally narrow and demographically skewed. Early adopters tend to be younger, highly educated, and technically inclined[2][39][3]. Their performance patterns may overstate both potential gains and the magnitude of inequality. Longitudinal observation is needed to determine whether current dispersion stabilizes or widens as organizational scaffolding matures and training diffuses. Measurement practices also vary. "Productivity" spans different denominators (speed, quality, or output volume) and is rarely benchmarked against controlled baselines. Most datasets are observational, not experimental, limiting inference about thresholds versus continuous effects. Wiese[40] demonstrates one approach to longitudinal quality measurement using bias-calibrated LLM-as-judge evaluation.

Future research should:

1. Track performance and compensation longitudinally across experience cohorts.
2. Instrument variation in scaffolding and workflow redesign to isolate the causal effects of architectural literacy.
3. Develop cross-domain metrics of AI fluency analogous to digital-literacy indices, enabling comparison beyond software work.

6. Flattening the Cliff: Design and Agenda

The observed discontinuities are not technologically inevitable. They arise from the speed of the industry and mismatched design, training, and diffusion regimes. Three complementary interventions (design scaffolding, literacy infrastructure, and equitable diffusion) can mitigate these effects.

6.1. Scaffolding by design

Interfaces and development environments can embed progressive scaffolds that translate architectural practices into guided workflows. Evidence from office productivity suites suggests that when such tools are integrated into everyday workflows, they systematically shift how workers allocate time rather than only improving isolated tasks[33]. Tools that visualize context state, support iteration checkpoints, or automatically test outputs reduce metacognitive load and help users cross the threshold without full engineering expertise. Empirical parallels already exist in high-scaffold domains such as customer support, where structured tools collapsed the performance gap between novices and experts[3]. Similar scaffolding could be generalized to creative and analytical work[32]. Scaffolding should not be understood as generic templates alone. Wong and Qiu[21] show that unrestricted ChatGPT use can improve immediate creative performance without producing equivalent later independent learning gains, while guided sequencing can improve later independent creativity. This implies that effective scaffolding should preserve human planning, critique, and verification rather than simply automate output generation.

6.2. Literacy as individual capability infrastructure

Crossing the cliff requires individual architectural literacy, not just prompt fluency. A practical architectural-literacy curriculum should include task decomposition, context construction, model limitation awareness, output evaluation, ethical judgment, and iterative workflow design. This is consistent with 2026 GenAI literacy research that treats effective GenAI use as a multidimensional capability involving interaction, evaluation, and ethical engagement rather than only prompt formulation[41][42]. According to the World Economic Forum's Future of Jobs Report[43], 60% of companies will demand AI literacy for competitive advantage[43]. Crossing the cliff appears to require more than short tutorials; it likely depends on curricular integration of computational and systems thinking. Educational programs and professional certification should frame LLMs not as conversational aids but as programmable systems. 2025 experiments in enterprise training show that limited exposure (<50 hours) yields minimal change[1]; depth and repetition are critical. National or sectoral "AI fluency frameworks" could standardize competencies in decomposition, orchestration, and validation, paralleling digital-literacy initiatives of earlier decades.

6.3. Equitable diffusion as organizational and market capability

Productivity gaps persist even when individuals acquire baseline literacy because the larger bottleneck lies in organizational redesign capability. Firms vary dramatically in their capacity to restructure workflows, integrate AI-native processes, and accumulate complementary organizational capital. Diffusion therefore requires investment in institutional supports: firm-level retraining programs, operational redesign, and strategic acquisitions that transfer AI-native practices into legacy environments. Wang X. et al.[8] suggest that labor

mobility from lean AI firms may operate as a diffusion mechanism for AI-native practices. These mechanisms address the structural threshold that determines whether organizations can absorb and scale architectural literacy. Without intentional diffusion, productivity gains will remain concentrated among AI-native firms, amplifying market power and wage dispersion[25][8][18]. Rockall et al.[30] explicitly characterize this as an efficiency–equity trade-off: firms endogenously over-adopt AI in high-wage tasks, which raises aggregate productivity but widens wealth gaps absent policy intervention. Policy interventions should aim to extend organizational capability, not only individual skill.

7. Conclusion

Despite the rapid diffusion of generative AI, productivity outcomes remain heterogeneous across users, tasks, organizations, and sectors. The 2025 evidence reviewed in this paper suggests substantial variance across software development, customer support, enterprise adoption, and labor-market outcomes[1][3][2][33][22]. Emerging 2026 evidence strengthens this interpretation by showing that AI productivity gains depend on the workflow environment, not seniority alone. Daniotti et al.[6] find stronger AI-coding gains among experienced developers in open-source GitHub data, while Gambacorta et al.[7] find large enterprise coding gains concentrated among junior and entry-level programmers in a structured CodeFuse deployment. This article distinguishes between model capability, tool adoption, net task productivity, and market results. Improvements in model capability do not always translate into increased productivity, and widespread adoption does not guarantee effective organizational transformation. The proposed causal chain is as follows: task structure and scaffolding define a user workflow capability; user workflow capability influences net task performance; repeated performance gains build up into organizational capability; and uneven organizational capability can result in labor-market and market-level inequality.

The strongest established pattern is persistent heterogeneity in productivity outcomes. The productivity cliff is our proposed mechanism for explaining why that heterogeneity is discontinuous rather than gradual: it predicts that gains concentrate above an architectural literacy threshold, whether that threshold is crossed by the user or embedded in the system through scaffolding. The threshold is not seniority itself; it is whether the user applies the decomposition, orchestration, and validation practices that define architectural literacy, either individually or through system scaffolding.

This framing is consistent with the customer-support evidence, where structured AI assistance helped less-experienced workers by embedding expert routines into the workflow[3], and with studies showing that AI productivity and learning effects depend heavily on task structure, collaboration design, and guided use[44][21]. It also requires caution: clinical and high-stakes domains show that potential quality gains can be offset by verification costs, factual errors, and accountability burdens[9]. Current data implies that unequal AI productivity gains are already occurring, driven by the disparity between universal tool access and uneven architectural literacy. The broader inequality consequences are conditional: if architectural literacy and workflow scaffolding remain unequally distributed, gains may concentrate among workers and organizations that can redesign work around AI capabilities[18][19][37][22]. At the same time, this outcome is not inevitable. Cross-country evidence suggests that generative AI can narrow some productivity gaps by raising performance in lower-income contexts[38], while macroeconomic work suggests that AI adoption may create both productivity gains and equity trade-offs depending on who captures the returns[30]. Zhang[45] similarly shows that AI can reshape within-occupation power dynamics even when outright job loss is limited.

As tools mature and scaffolding, literacy, and diffusion mechanisms improve, the cliff may flatten. Mitigating capability disparities requires democratizing not only model access but also the architectural literacy needed to decompose, orchestrate, and validate AI-augmented work. The central empirical task ahead is to determine when these thresholds persist, when scaffolding flattens them, and when apparent cliffs are better explained by selection effects, task mix, organizational context, or measurement noise.

Statements and Declarations

Funding

No specific funding was received for this work.

Potential Competing Interests

No potential competing interests to declare.

Data Availability

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Author Contributions

F.B. was the sole author and is responsible for all aspects of the manuscript.

References

1. ^ a b c d e f g h i j k l m *Becker J, Rush N, Barnes E, Rein D (2025). "Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity." arXiv. doi:10.48550/arXiv.2507.09089.*
2. ^ a b c d e f g h i j k *Sleegers W, Elsey J, Moss D (2025). "Adoption and Uses of LLMs Among U.S. Tech Workers." Rethink Priorities. https://rethinkpriorities.org/research-area/adoption-llms-tech-workers/.*
3. ^ a b c d e f g h i j *Brynjolfsson E, Li D, Raymond L (2025). "Generative AI at Work." Q J Econ. **140**(2):889942. doi:10.1093/qje/qjae044.*
4. ^ *Axios (2025). "Anthropic's Dario Amodei & Jack Clark & Axios' Jim VandeHei [Video]." YouTube. https://www.youtube.com/watch?v=nvXj4HTiYqA.*
5. ^ *Salva RJ (2025). "How Are Developers Using AI? Inside Our 2025 DORA Report." Google. https://blog.google/innovation-and-ai/technology/developers-tools/dora-report-2025/.*
6. ^ a b c d e *Daniotti S, Wachs J, Feng X, Neffke F (2026). "Who Is Using AI to Code? Global Diffusion and Impact of Generative AI." Science. **391**(6787):831835. doi:10.1126/science.adz9311.*
7. ^ a b c d e *Gambacorta L, Qiu H, Shan S, Rees DM (2026). "Generative AI and Labour Productivity: A Quasi Experiment on Coding." J Financ Stab. **84**:101543. doi:10.1016/j.jfs.2026.101543.*
8. ^ a b c d e f g *Wang X, Feng C, Sun T (2025). "AI Spillover Is Different: Flat and Lean Firms as Engines of AI Diffusion and Productivity Gain." arXiv. doi:10.48550/arXiv.2511.02099.*
9. ^ a b *Wang G, Zhang K, Jiang J, Wang C, Bi H, Liang H, Qi Z, Huang Y, Li Y, Yang X (2026). "HumanLarge Language Model Collaboration in Clinical Medicine: A Systematic Review and Meta-Analysis." npj Digit Med. **9**:195. doi:10.1038/s41746-026-02382-2.*
10. ^ *Havlk V (2025). "Why Are LLMs' Abilities Emergent?" arXiv. doi:10.48550/arXiv.2508.04401.*
11. ^ *Gulati K (2025). "Benchmarking the Future of Work: Mapping AI Progress to Occupational Tasks." SSRN Electron J. doi:10.2139/ssrn.5452354.*
12. ^ *Schorcht S, Mller FA, Buchholtz N (2026). "No One-Size-Fits-All: A Study of Prompt Techniques and Large Language Models to Enhance AI's Mathematics Educational Quality." ZDM Math Educ. Advance online publication. doi:10.1007/s11858-026-01784-6.*
13. ^ a b *Brynjolfsson E, Li D, Raymond L (2025). "Generative AI at Work." Q J Econ. **140**(2):889942. doi:10.1093/qje/qjae044.*
14. ^ a b *Meyer JHF, Land R (2003). "Threshold Concepts and Troublesome Knowledge: Linkages to Ways of Thinking and Practising Within the Disciplines." ISL10 Improving Student Learning: Theory and Practice Ten Years On. Oxford Brookes University. 412424. https://www.research.ed.ac.uk/en/publications/threshold-concepts-and-troublesome-knowledge-linkages-to-ways-of-.*
15. ^ a b *Chi Z, Dong L, Dong Q, Hao Y, Wu X, Huang S, Wei F (2025). "The Era of Agentic Organization: Learning to Organize With Language Models." arXiv. doi:10.48550/arXiv.2510.26658.*
16. ^ *Goodhue DL, Thompson RL (1995). "Task-Technology Fit and Individual Performance." MIS Q. **19**(2):213236. doi:10.2307/249689.*
17. ^ *Verenikina I (2003). "Understanding Scaffolding and the ZPD in Educational Research." Proceedings of the International Education Research Conference (AARE-NZARE), 30 November3 December 2003, Auckland, New Zealand. Research Online, University of Wollongong. https://ro.uow.edu.au/edupapers/381.*
18. ^ a b c d e f g h *Autor D, Thompson N (2025). "Expertise." NBER Working Paper No. 33941. National Bureau of Economic Research. doi:10.3386/w33941.*
19. ^ a b c d *Fan T (2025). "The Labor Market Incidence of New Technologies." arXiv. doi:10.48550/arXiv.2504.04047.*
20. ^ *Ball C, Huang K-T, Nie X, Kong E, Dilinika JMS (2026). "The Generative Artificial Intelligence (GenAI) Divide: Exploring the Individual Factors Influencing GenAI Competency, Adoption Readiness, and Psychological Barriers." Cyberpsychol Behav Soc Netw. **29**(3). doi:10.1177/21522715261423789.*
21. ^ a b c *Wong SSH, Qiu SX (2026). "Think First, ChatGPT Later: Guiding HumanAI Collaboration for Learning Gains in Independent Human Creativity." Educ Psychol Rev. **38**:45. doi:10.1007/s10648-026-10118-7.*
22. ^ a b c d e f g *Johnston AC, Makridis CA (2026). "AI, Output, and Employment." CESifo Working Paper No. 12579. CESifo. https://www.ifo.de/DocDL/cesifo1_wp12579.pdf.*
23. ^ *Ganuthula VRR (2025). "The Solo Revolution: A Theory of AI-Enabled Individual Entrepreneurship." arXiv. doi:10.48550/arXiv.2502.00009.*
24. ^ a b *Shi H (n.d.). "Official Lean AI Native Companies Leaderboard: The Future of 1-Person Billion-Dollar Startups." Lean AI Leaderboard. https://leanaileaderboard.com/.*
25. ^ a b c d *Brodzicki T (2024). "Heterogeneous Firms and AI Adoption. Dynamic Insights Into Market Structure and Global Trade." MPRA Paper No. 127767. Munich Personal RePEc Archive. https://mpra.ub.uni-muenchen.de/127767/.*
26. ^ a b c d *Doddapaneni P, Radzevych B, Breeden S, Bansal B, Rao T (2025). "From Pilots to Payoff: Generative AI in Software Development." Bain & Company. https://www.bain.com/insights/from-pilots-to-payoff-generative-ai-in-software-development-technology-report-2025/.*
27. ^ a b c *Challapally A, Pease C, Raskar R, Chari P (2025). "The GenAI Divide: State of AI in Business 2025." MIT NANDA. https://mlq.ai/media/quarterly_decks/v0.1_State_of_AI_in_Business_2025_Report.pdf.*

28. ^Gassmann O, Wincent J (2025). "The Non-Human Enterprise: How AI Agents Reshape Organizations." California Management Review. https://cmr.berkeley.edu/2025/10/the-non-human-enterprise-how-ai-agents-reshape-organizations/.
29. a, b, cDominski J, Lee YS (2025). "Advancing AI Capabilities and Evolving Labor Outcomes." arXiv. doi:10.48550/arXiv.2507.08244.
30. a, b, cRockall EJ, Tavares MM, Pizzinelli C (2025). "AI Adoption and Inequality." IMF Working Papers. **2025**(068). doi:10.5089/9798229006828.001.
31. ^Cui KZ, Demirer M, Jaffe S, Musolff L, Peng S, Salz T (2025). "The Effects of Generative AI on High-Skilled Work: Evidence From Three Field Experiments With Software Developers." SSRN Electron J. doi:10.2139/ssrn.4945566.
32. a, bWang ZZ, Shao Y, Shaikh O, Fried D, Neubig G, Yang D (2025). "How Do AI Agents Do Human Work? Comparing AI and Human Workflows Across Diverse Occupations." arXiv. doi:10.48550/arXiv.2510.22780.
33. a, b, cDillon EW, Jaffe S, Immorlica N, Stanton CT (2025). "Shifting Work Patterns With Generative AI." arXiv. doi:10.48550/arXiv.2504.11436.
34. a, bBrynjolfsson E, Chandar B, Chen R (2025). "Canaries in the Coal Mine? Six Facts About the Recent Employment Effects of Artificial Intelligence." Stanford Digital Economy Lab Working Paper. https://digitaleconomy.stanford.edu/publication/canaries-in-the-coal-mine-six-facts-about-the-recent-employment-effects-of-artificial-intelligence/.
35. ^Chen D, Kane C, Kozlowski A, Kunievsky N, Evans JA (2025). "The (Short-Term) Effects of Large Language Models on Unemployment and Earnings." arXiv. doi:10.48550/arXiv.2509.15510.
36. ^Humlum A, Vestergaard E (2025). "Large Language Models, Small Labor Market Effects." NBER Working Paper No. 33777. National Bureau of Economic Research. doi:10.3386/w33777.
37. a, b, cBone M, Gonzlez Ehlinger E, Stephany F (2025). "Skills or Degree? The Rise of Skill-Based Hiring for AI and Green Jobs." Technol Forecast Soc Change. **214**:124042. doi:10.1016/j.techfore.2025.124042.
38. a, bWade M, Baldwin R, Bjerkan-Wade B (2025). "Research: Gen AI Changes the Value Proposition of Foreign Remote Workers." Harvard Business Review. https://hbr.org/2025/01/research-gen-ai-changes-the-value-proposition-of-foreign-remote-workers.
39. ^Hartley J, Jolevski F, Melo V, Moore B (2024). "The Labor Market Effects of Generative Artificial Intelligence." SSRN Electron J. doi:10.2139/ssrn.5136877.
40. ^Wiese T (2026). "Human-Anchored Longitudinal Comparison of Generative AI With a Bias-Calibrated LLM-as-Judge." PLoS One. **21**(2):e0339920. doi:10.1371/journal.pone.0339920.
41. ^Ding L, O'Berry R, Tallent H, Sanju Gharti Chhetri GC, Gapud A (2026). "Learning With Large Language Models: Beyond Prompt Engineering." Educ Inf Technol. **31**:28772901. doi:10.1007/s10639-026-13924-2.
42. ^Durak G, ankaya S, nc S (2026). "A Theory-Driven Scale for Assessing Text-Based Generative AI Literacy From a Self-Efficacy Perspective (T-GASE)." Educ Inf Technol. doi:10.1007/s10639-026-14023-y.
43. a, bWorld Economic Forum (2025). "The Future of Jobs Report 2025." World Economic Forum. https://www.weforum.org/publications/the-future-of-jobs-report-2025/.
44. ^Li Y, Yang F, Wu M, Li J (2026). "Human-AI Collaboration Enhances the Performance of Large Language Models in Risk of Bias Assessment." BMC Med Res Methodol. **26**:37. doi:10.1186/s12874-025-02763-3.
45. ^Zhang Y (2025). "From Automation Technology to Generative AI: Skill Heterogeneity in Technology's Impact on Laborers." J Chin Sociol. **12**:22. doi:10.1186/s40711-025-00249-9.

Declarations

Funding: No specific funding was received for this work.
Potential competing interests: No potential competing interests to declare.