How Do AI Agents Do Human Work?
Comparing AI and Human Workflows Across Diverse Occupations

Zora Zhiruo Wang [C] **Yijia Shao** [S] **Omar Shaikh** [S]
Daniel Fried [C] **Graham Neubig** [C] **Diyi Yang** [S]
[C] Carnegie Mellon University [S] Stanford University
zhiruow@cs.cmu.edu

Abstract

AI agents are continually optimized for tasks related to human work, such as software engineering and professional writing, signaling a pressing trend with significant impacts on the human workforce. However, these agent developments have often not been grounded in a clear understanding of how humans execute work, to reveal what expertise agents possess and the roles they can play in diverse workflows. In this work, we study *how agents do human work* by presenting the first *direct comparison of human and agent workers* across multiple essential work-related skills: data analysis, engineering, computation, writing, and design. To better understand and compare heterogeneous computer-use activities of workers, we introduce a scalable toolkit to induce interpretable, structured workflows from either human or agent computer-use activities.[1] Using such induced workflows, we compare how humans and agents perform the same tasks and find that: (1) While agents exhibit promise in their alignment to human workflows, they take an overwhelmingly programmatic approach across all work domains, even for open-ended, visually dependent tasks like design, creating a contrast with the UI-centric methods typically used by humans. (2) Agents produce work of inferior quality, yet often mask their deficiencies via data fabrication and misuse of advanced tools. (3) Nonetheless, agents deliver results 88.3% faster and cost 90.4–96.2% less than humans, highlighting the potential for enabling efficient collaboration by delegating easily programmable tasks to agents.

1. Introduction

AI agents are increasingly developed to perform tasks traditionally carried out by human workers (Bick et al., 2024; Brynjolfsson et al., 2018; Eloundou et al., 2023), as reflected in the growing competence of computer-use agents in work-related tasks such as software engineering (Jimenez et al., 2024; Wang et al., 2025a) and writing (Clark et al., 2018). Nonetheless, they still face challenges in many scenarios such as basic administrative or open-ended design tasks (Xu et al., 2024), sometimes creating a gap between expectations and reality in agent capabilities to perform real-world work (Ryoo et al., 2025; Zhang et al., 2021).

To further improve agents' utility at such tasks, we argue that it is necessary to look beyond their end-task outcome evaluation as measured in existing studies (Patwardhan et al., 2025), and investigate *how* agents currently perform human work — understanding their underlying workflows to gain deeper insights into their work process, especially how it aligns or diverges

[1] https://github.com/zorazrw/workflow-induction-toolkit

from human workers, to reveal the distinct strengths and limitations between them. Therefore, such an analysis should not benchmark agents in isolation (Patwardhan et al., 2025; Xu et al., 2024), but rather be grounded in comparative studies of human and agent workflows. More broadly speaking, to foster a collaborative future where humans and AI work effectively together, it is crucial to understand how both operate in practice, and to anticipate the organizational and behavioral changes required as agents assume greater roles (Masters et al., 2025).

To this end, we first propose to *perform a direct comparison of how human and agent workers* do the same tasks. Beyond specific domains (Choi and Schwarcz, 2024; Jimenez et al., 2024), we aim to provide a holistic view of how agents might reshape the broader human workforce, by systematically studying human and agent workflows across occupations via essential shared skills — data analysis, engineering, computation, writing, and design — instantiated with 16 realistic, long-horizon tasks. We estimate these tasks to be representative of 287 computer-using U.S. occupations and 71.9% of their daily work activities (§2). Comparing human and agent activities is not trivial: we lack a unified representation and a scalable channel for analysis. We first collect human activities in a similar way to agent activities with actions and screen states. However, activities in their raw format, low-level mouse and keyboard actions, are largely context-independent (i.e., a single key press may not map to a meaningful user-intended task) (Shaw et al., 2023) thus unintuitive to interpret for both human and agent readers (Wang et al., 2024b). Inspired by humans who naturally reason about work in terms of *higher-level workflows*: structured sequences of actions performed together to achieve a goal (Deka et al., 2016; Liu et al., 2018; Wang et al., 2025c). We *develop a workflow induction procedure* that transforms heterogeneous computer-use activities into hierarchical, interpretable workflows. These workflows provide a shared representation that enables effective comparative analysis in our study (§3).

Using this method, we compare 48 human workers and 4 representative LLM agent frameworks on 16 curated work-related tasks. We find that:

• **Agents take an overwhelmingly programmatic approach.** While agents show initial promise in exhibiting human-aligned workflows, they write programs to solve essentially all tasks, even when equipped with and trained for UI interactions. This prevails across work domains — not only programmable tasks such as data analysis, but heavily among open-ended, visual tasks such as design; forming a stark contrast to the visual-oriented human workflows (Figure 4).

• **Human workflows are substantially altered by AI automation, but not by AI augmentation.** One quarter of human activities we studied involve AI tools, with most used for augmentation purposes: integrating AI into existing workflows with minimal disruption, while improving efficiency by 24.3%. In contrast, AI automation markedly reshapes workflows and slows human work by 17.7%, largely due to additional time spent on verification and debugging (Figure 5).

• **Agents produce lower-quality work, concerningly by fabrication and tool misuse.** Agents produce work of inferior quality, most notably by fabricating data to deliver plausible outcomes, or misuse advanced tools (e.g., deep research) to mask their limitations (e.g., retrieve alternative files when unable to read user-provided ones); among issues in intent misinterpretation, vision inability, and trouble transforming between program- and UI-friendly data formats (Figure 6).

• **Teaming humans and agents based on their accuracy and efficiency advantages.** Despite the gap in quality, agents exhibit clear advantages in efficiency. Compared to an average human worker, agents deliver work 88.3–96.6% faster and at 90.4–96.2% lower costs. Our induced workflows naturally suggest a division of labor: readily programmable steps can be delegated to agents for efficiency, while humans handle the steps where agents fall short. Together, this forms a human-agent teaming jointly optimized for work quality and efficiency (Figure 7).

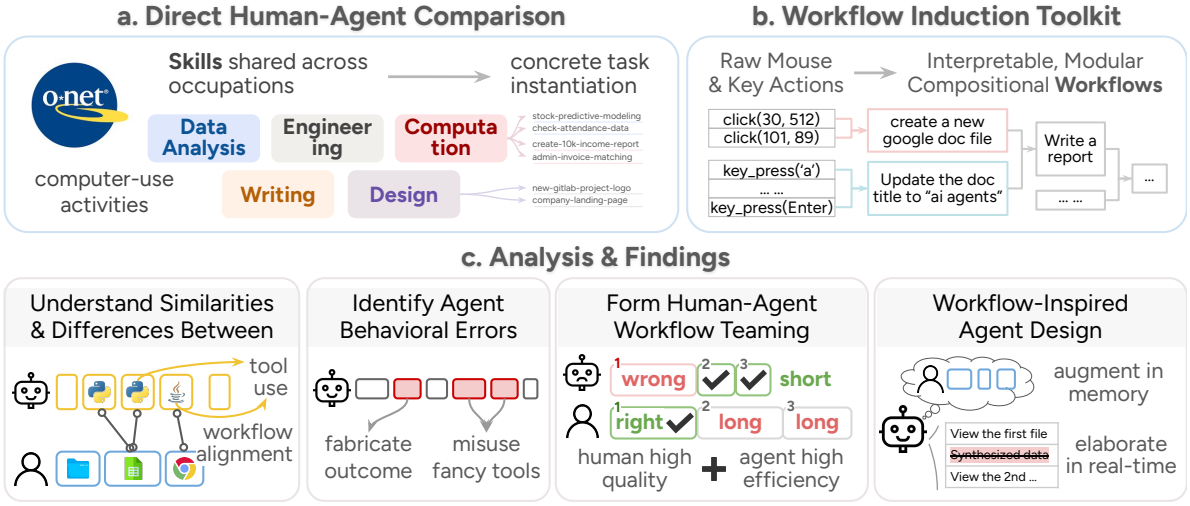

Figure 1: Overview of our study: a direct human-agent worker comparison (a) supported by our workflow induction toolkit (b), and four major analysis findings (c).

2. Studying Humans and Agents At Work

We propose the first study to directly compare agent workflows with respect to human workers.

- First, this comparison between human and agent workers goes beyond measuring performance — it reveals *how* each completes tasks, uncovering their underlying skill profiles and behavioral differences. Existing evaluations often reduce agent performance to numerical metrics. To build a collaborative future where humans and AI work together, we must instead understand their respective workflows in detail. By comparing how humans and agents conduct the same tasks, we can identify opportunities for intelligent delegation and informed division of labor — whether agents should assist in particular steps as requested, collaborate with human workers more proactively, or eventually automate tasks end-to-end?

- Moreover, benchmarking agents in isolation has inherent limitations. Even when an agent trajectory meets annotated evaluation rubrics, it may still follow a suboptimal workflow, produce a less desirable outcome, or omit critical steps — issues that remain obscured without a structured lens of comparison to human workers. To address this, we induce both human and agent activities into a shared workflow representation, offering a more interpretable and auditable format. This unified abstraction allows us to systematically compare how each performs work, uncover overlooked dimensions in task execution, and identify concrete directions for improving agent labor toward human-level proficiency.

Below we introduce how we build a taxonomy of work-essential skills and create representative tasks to study them (§2.1), followed by the collection process of various human (§2.2) and agent (§2.3) worker activities in conducting these tasks.

2.1. Representative Work-Related Tasks

While studying on the scale of thousands of tasks is prohibitively costly, we leverage the fact that many different occupations share similar skills. We take a more scalable approach by studying workers via a set of prototypical skills (Figure 2).

Building the Skill Taxonomy We manually identify shared skills from the list of task require-

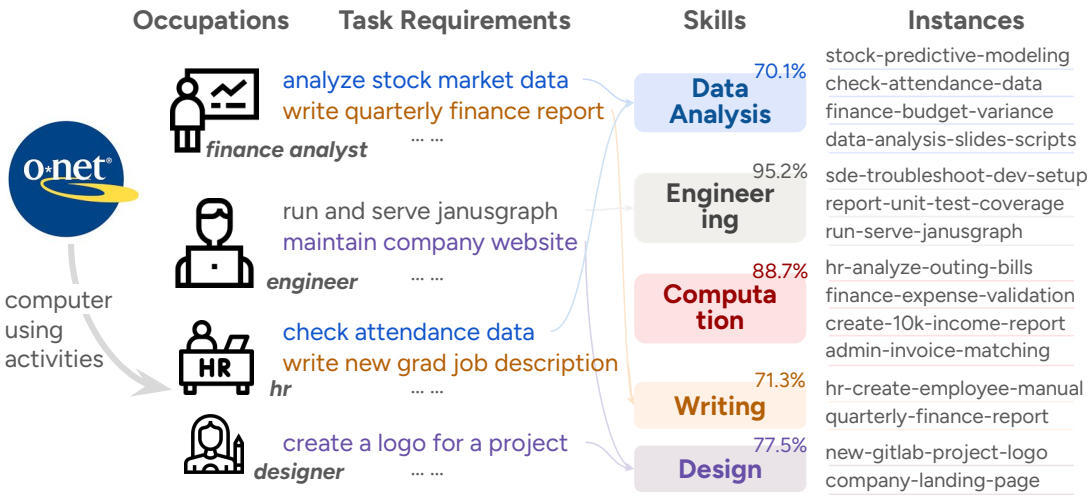

Figure 2: Identifying core work-related skills based on O*NET database, then create diverse, complex task instances to ensure skill representativeness (percentages marked in skill boxes).

ments across computer-related occupations. More concretely, we source all occupation records from the U.S. Department of Labor's O*NET database (National Center for O*NET Development, 2024), which contains 923 occupations with 18,796 task requirements in the U.S. labor market. As our focus is on computer-using tasks that may be assisted or automated by digital agents, we filter tasks following Shao et al. (2025) and yield 2131 tasks from 287 occupations. We annotate the high-level skills required for tasks, and focus on skills that are feasible for automation. In the end, we identified five major skill categories (data analysis, engineering, computation, writing, and design) that, in aggregate, affect these 287 computer-using U.S. occupations and 71.9% of the daily tasks involved, according to the O*NET records. This high coverage ensures that our study can affect a large portion of human work. More details on skill annotation are in §A.1.

Creating High-Coverage Tasks for Skills For each skill category, we next design representative tasks that reflect how these skills are applied across real-world jobs. Since a skill can appear in many job contexts, we aim to create versatile tasks that capture common work scenarios. For instance, the data analysis skill is instantiated in finance (`stock-predictive-modeling`) and administrative (`check-attendance-data`) domains, while the design skill is exercised in technical (`company-landing-page`) and graphical (`project-logo`) contexts (Figure 2).

To benchmark complex, realistic work-related tasks, we follow TheAgentCompany (TAC) (Xu et al., 2024) setup, where each instance contains a task instruction, necessary environmental contexts (e.g., input files, software at pre-work states), and an executable program evaluator for rigorous correctness checking. We adopt TAC's multi-checkpoint evaluation protocol to capture progress at intermediate stages, enabling fine-grained assessment beyond the final outcomes. For reproducibility and fair comparison, all experiments run in TAC's sandboxed environments, which host engineering tools (`bash`, `python`) and work-related websites for file sharing, communication, and project management. We adopt TAC tasks for skills such as data analysis and engineering. But because TAC is based on a software engineering company, we need to create additional tasks, particularly writing and design, to better represent occupations across all skill domains. Refer to §A.2 for the full list of tasks we studied.

To measure task coverage within each skill category, we mark whether each occupation's task requirements are represented in our tasks and compute the proportion of covered employment relative to all occupations using that skill. As shown in Figure 2, our tasks cover 70.1–95.2% of

total employment across skills, providing a solid ground for drawing conclusions generalizable to real-world work.

2.2. Collecting Human Worker Activities

Human Worker Recruitment For each task, we hire 3 qualified human workers from Upwork with relevant educational backgrounds and current professional experience in pertinent skills. We screened candidates based on their work portfolios and prior client ratings to ensure high-quality work. Workers can use any preferred software or tools, including professional and AI tools, to emulate their realistic workflows and act as good references for agent workers.

The Recording Tool We developed a recording tool to collect the computer-use activities of human workers. More specifically, each activity session involves (i) a sequence of mouse and keyboard *actions* that the user takes on their computer, as well as (ii) screenshots of the computer when any actions are issued, i.e., *states*. Recording both (i) and (ii) ensures our human task-solving trajectories contain the same component and share the same action and state spaces with AI-powered agent workers, as we will introduce in §2.3.

Processing Raw Computer-Use Activities Computer-use activities collected from human real-world usage can be noisy and redundant, due to the limitations in the recording tool implementation. We hence perform post-processing to merge consecutive `keypress` and `scroll` actions, as well as merging two `click()` within 0.1 seconds into a single `double_click()`. This effectively simplifies the trajectory data, reducing the action count by 83.2% (from 5831.1 to 981.1 actions) in an average human trajectory, meanwhile aligning better with the granularity of agent computer activities (Wang et al., 2025b).

Refer to §A.3 for more details in human recruitment and activity processing.

2.3. Collecting AI Agent Activities

We also collect the work activities of representative computer-use agents, to demonstrate the advanced behaviors of potential agent workers.

Agent Frameworks We experiment with three agent frameworks to represent varied work behaviors, including: (i) two widely-used closed-source agents, ChatGPT Agent and Manus, built on GPT and Claude LM backbones; and (ii) the open-source OpenHands agent, which is more oriented towards coding scenarios, supported by `gpt-4o` and `claude-sonnet-4`. This setup enables a direct comparison across agent frameworks while controlling for differences in model families (i.e., the same backbones as ChatGPT and Manus). We collect all agents' trajectories in solving all of the tasks. In general, all agents can take UI actions such as mouse clicks and keyboard presses, as well as programming-related actions such as `execute_command`. Individual agents have access to varied high-level tools (e.g., `view_image` for visual understanding) by design. See §A.4 for the full list of available actions for each agent framework.

Collection Results In summary, we collected 112 trajectories for 16 tasks spanning across the 5 major skill categories. We collected 64 agent and 48 human trajectories, which constitute 33.8 and 981.1 steps on average,[2] substantially higher than previous computer-use agent benchmarks (e.g., the best agents take 5.9 steps to solve an average WebArena task (Wang et al., 2025c)), signaling the greater complexity and realism of the work-related tasks we study.

[2]Human actions are often taken on a finer granularity (e.g., `key_press({a short phrase})`) compared to agent actions (e.g., `key_press({an entire file})`) and contain more noisy shifts (e.g., between type and click).

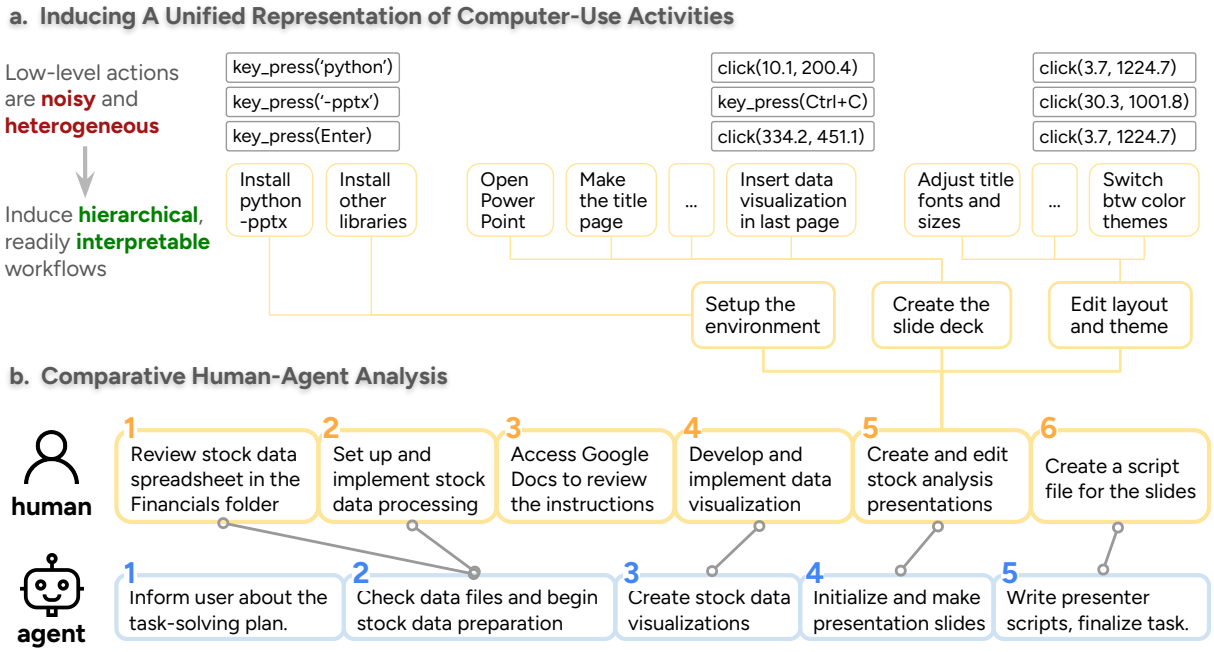

Figure 3: We induce interpretable, hierarchical workflows from low-level computer-use activities, offering a unified representation for human and agent work activities (a), in the meantime, facilitating direct comparative analysis between human and agent workers (b).

3. Inducing Workflows of Computer-Use Activities

Computer-use activities consist of low-level mouse and keyboard actions that can be hard to interpret and learn *verbatim* (Figure 3, a). It is therefore useful to induce workflows (Figure 3, b) that are meaningful and readily interpretable, to facilitate work activity analysis. In this section, we introduce our toolkit to induce workflows (§3.1) from raw computer-use activities of either human or agent workers (§3.2), and validate workflow quality via multiple dimensions (§3.3) to prepare for comparative analysis (§3.4).

3.1. Workflow: Preliminaries

Definition 1. *A workflow is a sequence of steps taken to achieve a certain goal, where each workflow step consists of one or more actions to accomplish a distinguishable sub-goal.*

In this work, goals and sub-goals are expressed in natural language (NL) (e.g., yellow blocks in Figure 3), and the goal is specified by our task instructions (§A.2). Actions can be any mouse, keyboard, or high-level actions defined in the human (§2.2) or agent (§2.3) action space.

Based on this definition, inducing workflows from raw computer-use activities boils down to (i) segmenting the activities into meaningful intervals, and (ii) annotating meaningful sub-goals of each activity interval. In this way, we identified the sequence of workflow steps, each with an NL sub-goal (ii) and a sequence of actions (i). This structure offers multiple useful properties: interpretability, modularity, and compositionality. We first introduce the induction procedure, then demonstrate and validate the three desired properties of workflows.

3.2. Workflow Induction

Trajectory Segmentation We first segment each activity into consecutive sub-trajectories representing distinct procedures with different inputs and outputs. To detect boundaries, we measure pixel-level mean squared error (MSE) between consecutive screenshots — large visual shifts (e.g., from Chrome to VSCode) indicate transitions between tasks. However, MSE alone captures visual but not semantic changes, leading to overly fine-grained splits (e.g., split when zooming in a page). To address this, we merge adjacent segments that are semantically coherent (i.e., feature consistent states, tools, and goals) by passing in the visual states and textual actions into a multimodal LM.[3] For instance, actions like 'adjust column width', 'bold header row', and 'add borders' in Excel are merged as a single 'apply formatting to data sheet' step.

Compositional Workflow Hierarchy Workflow steps can vary in granularity depending on the *scope of a task*. To support flexible task scopes, we propose a general *workflow specification language* that builds a multi-level workflow hierarchy via *compositional* annotation. Each top-level workflow step can be treated as a task node and recursively decomposed into finer-grained steps until reaching atomic actions. For example, in Figure 3, a top-level workflow step 'create and edit stock analysis presentations' can be split into 'setup the environment', 'create the slide deck', etc.; each can be further divided until reaching the lowest-level `click` or `key_press` actions. In practice, we find that summarizing child goals yields more accurate parent goals. Thus, given a computer-use activity, we construct the full hierarchy and iteratively derive higher-level goals bottom-up until reaching the task root.[3]

Although we primarily describe the workflow induction process in the context of human activities, our tool can be directly applied to induce workflows from agent trajectories as well. We use the same workflow induction procedure for all computer-use activities collected from human and agent workers, to enable consistent and comparable analysis (§4, §5)

3.3. Workflow Quality Validation

To ensure the quality of induced workflows and avoid error propagation in subsequent analysis, we examine workflows with automated and manual efforts. Following our earlier definition (Def 1), each workflow step is assessed on two criteria: if (i) the actions and states are consistent with the NL goal, and (ii) it is modular: meaningfully distinguishable from adjacent steps.

Action-Goal Consistency For each step, we input the NL sub-goal and the sequence of actions (with intermittently sampled states every 10 actions) to an LM,[4] and ask it to measure the alignment, by outputting a binary `YES/NO` answer. We map `YES` and `NO` to score 0 or 1, and calculate the average score of all steps as the measure for the entire workflow.

Modularity We pass in the list of goals of workflow steps to an LM[4] to measure if each step serves as a distinguishable step apart from its adjacent steps (i.e., these steps should not be repetitive or redundant), by outputting a binary `YES/NO` answer. We similarly calculate and report the step averages. See §B.2 for specific prompts for both evaluations.

Worker	Action-Goal Consistency	Modularity
Human	92.8	83.8
Agent	95.6	98.1

Table 1: Our tool induces high-quality human and agent workflows on consistency and modularity measures.

We measure both human and agent trajectories on both metrics and report the average scores of all workflows in Table 1.

[3]We use `claude-sonnet-3.7` for both segment merging and hierarchical goal annotation. See prompts in §B.1.
[4]We use `claude-sonnet-3.7` to evaluate workflow quality and match workflow pairs.

For both human and agent workflows, the evaluation scores for all metrics are above 80.0%, suggesting overall decent quality of workflows induced. We also manually verify the workflows and their evaluation results, which shows substantial agreement with human judgments — 0.637 and 0.781 in Cohen's Kappa for consistency and modularity metrics — to ensure the quality of workflows to support subsequent analysis.

3.4. A Prerequisite for Comparative Analysis: Workflow Alignment

Beyond analyzing individual workers, it is important to compare workers to understand their respective characteristics and approaches. Since the induced human and agent workflows can be heterogeneous, we introduce a *workflow alignment procedure* that automatically maps steps between any pair of induced workflows to quantify their alignment. In this study, we focus on human-agent comparisons. As in Figure 3 (b), step intervals (e.g., human 1–2 and agent 2–2) are first matched by an LM^4 then manually refined for accuracy. Based on this matching result, we propose two quantitative metrics: (i) *matching steps* (%): The percent of steps matched in all steps of both workflows. (ii) *order preservation* (%): The percent of matched steps that are kept in order as their original workflow, i.e., do not produce crossing lines in Figure 3 (b). Higher values in both metrics indicate higher alignment of the two compared workflows.

Overall, we propose a workflow toolkit that can (i) induce compositional, modular, and interpretable workflows from heterogeneous raw computer activities, and (ii) enable comparative analysis via pairwise alignment between any pair of workers of interest. This unified workflow representation also opens up opportunities to explore additional dimensions in future work.

Next, we systematically analyze the workflows of human and agent workers, in their inter-alignment, work quality, and efficiency, among other aspects, to provide deeper insights into how agents conduct and affect human work.

4. Do Humans and Agents Work in the Same Way?

Figure 4 (a) presents the workflow alignment measures among agents, among humans, and between human and agent workers. Human and agent workflows share 83.0% of steps, with step orders preserved for 99.8% of the time — both indicating a high degree of alignment (§C.1). This alignment is especially pronounced between capable agents that complete tasks end-to-end and independent human workers who do not rely on AI tools.

4.1. The Human Way, The Agent Way

Despite the aligned procedure, human workers use a diverse set of tools to conduct the same step. As exemplified in Figure 4 (b), the task asks workers to first clean, analyze, and visualize the data, then create presentation slides and scripts. Compared to agent workers that mainly use programming tools (`Python` and `bash`), humans employ a variety of interactive, UI-oriented tools throughout the task workflow. For example, processing data with `Jupyter Notebook` or `Excel`, making slides with `PowerPoint` and even the AI-supported `Gamma` tool. We provide tool usage visualization for all work-related tasks involved in our study in §C.2.

Agents Are Programmatic Workers In contrast to human workers, agents seldom execute tasks via the UI interface. Instead, agents always employ alternative programmatic tools. Figure 4 (c) illustrates agents' program use rate across tasks. As clearly shown by the high 93.8% program-use rate, this programmatic bias extends beyond engineering tasks to encompass

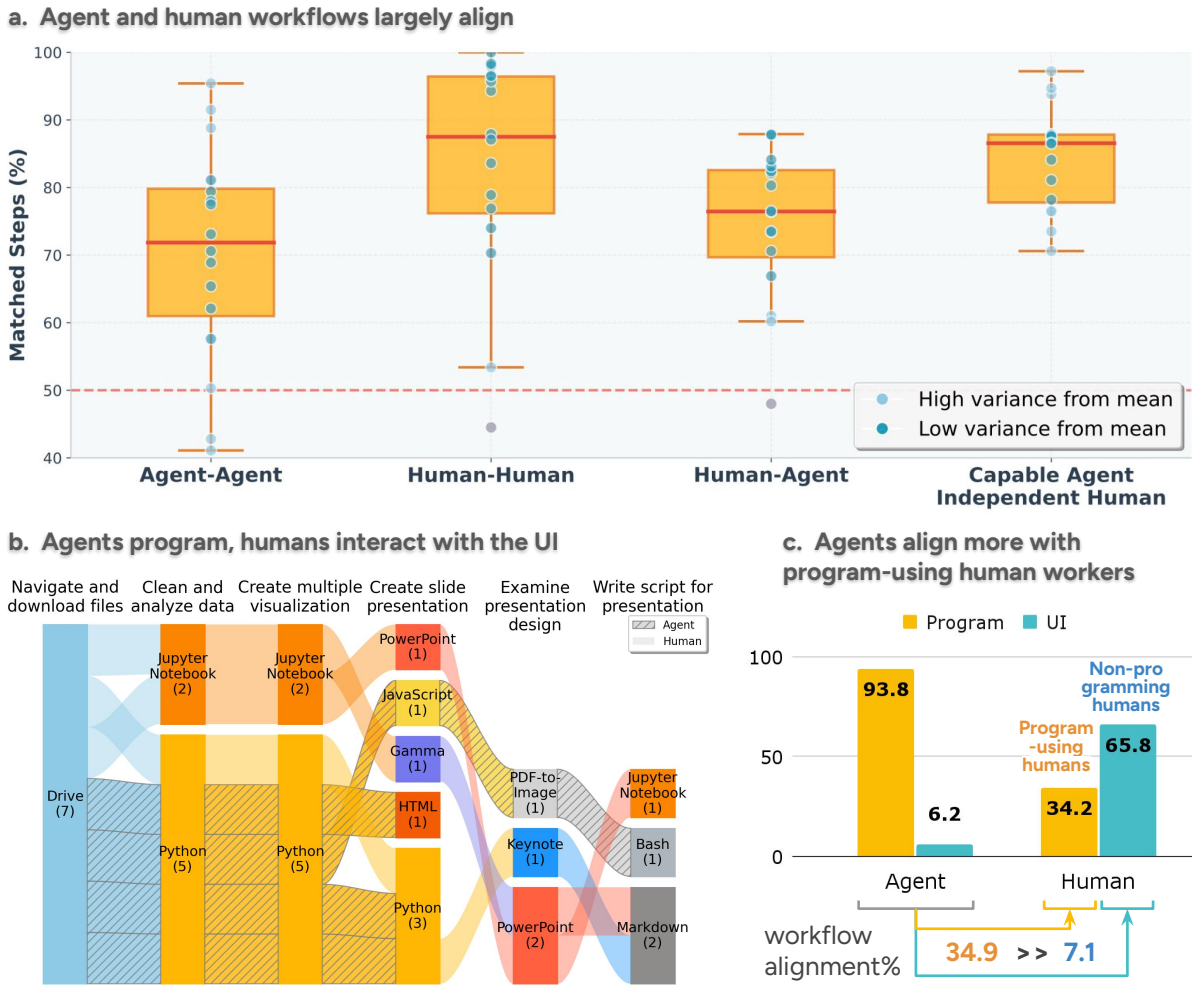

Figure 4: Agent workers largely align with human workflows (a). Humans use diverse UI tools, while agents heavily rely on programmatic tools (b). Agent program to solve all tasks, and better align with program-using human workflows (34.9%) than those without programs (7.1%) (c).

virtually all computer-based activities, including those not inherently programmable, such as administrative or design work (see more details in §C).

To concretely test this hypothesis, we measure the alignment of agent workflows to human workflows that use programmatic tools and non-programmatic tools, respectively. Since the high-level steps mostly match (e.g., human step 4 and agent step 3 in Figure 3), we analyze steps one level deeper in the workflow hierarchy (e.g., finer steps in Figure 3, a). As shown in Figure 4 (c), agents exhibit 27.8% stronger alignment with program-using human steps than those that do not. This highlights a defining characteristic of agent workers — a programmatic mode of operation. This is not only true for coding-oriented OpenHands agents, but also for general-purpose computer-use agents such as ChatGPT and Manus. While this programmatic tendency is understandable in domains represented in LM code training data (e.g., engineering), this programmatic behavior notably persists even in visually intensive tasks such as slide creation and graphical design. Not only is such 'blind' visual creation possible, but LM-supported agents may in fact be more proficient at editing in the symbolic space (i.e., write programs) than in the visual space (i.e., adjust pixels) (Qiu et al., 2024).

This stronger alignment among workflows using similar tools also resonates with the notion

of tool affordance from a design perspective (Norman, 2013) – *We shape our tools, and thereafter our tools shape us* (McLuhan, 1994). When two workers, whether human or agent, employ the same tool to complete the same workflow step, their workflows tend to converge more closely at finer levels of granularity. Moreover, the design of human-agent interfaces must account for the granularity of interaction (Liang et al., 2025): when collaboration occurs at high-level workflow steps, it may suffice for a tool to be accessible to either the agent or the human; at finer granularities, however, the tool must support both programmatic and UI-based actions for seamless coordination (Song et al., 2024).

Agent Workers Use Versatile Programming Tools Despite the shared programmatic approaches, agents employ a range of programmatic tools, many of which are custom-built by their developers. For example, when designing a company landing page, OpenHands-GPT uses basic `PIL.Image` drawing, OpenHands-Claude uses HTML, ChatGPT uses an internal image generation tool, and Manus uses a specially-designed ReAct program — yielding webpages with varied characteristics. This suggests that developing task-oriented programming tools, as functional equivalents to human-preferred UI interfaces, could be an effective way to build more capable agent workers. See §C.2 for more detailed examples.

4.2. Human Workflows are Substantially Altered by AI Automation

When completing the same task, individual human workers exhibit distinct habits. A common pattern among proficient workers is frequent *intermediate verification* — ranging from quickly opening a file to check if a valid image is generated, or more carefully on the computation correctness. Such verification occurs more often when outputs are produced indirectly via programs, since UI-based creation allows for implicit, step-by-step verification, and thus reduces the need for later rechecking. This verification behavior, already observed in some agent workflows, potentially contributes to maintaining higher work quality.

Unlike agent workers, humans often *revisit instructions during* the working process, particularly when the task chain is long. This may stem from humans' limited memory capacity for retraining all instructional details, but it could also be a grounding strategy to disambiguate instructions for the next immediate sub-task (i.e., workflow step)—preventing false assumptions.

Interestingly, since our study does not restrict the tools used by human workers, we observe that a notable 24.5% of human workflows involve one or more AI tools.

How does Using AI Tools Affect Human Workflows? Among human workers who use AI tools, the majority, 75.0% in our study, do so for *augmentation* purposes (Handa et al., 2025). Compared to those who complete tasks independently (without AI assistance; e.g., explore Figma templates to find logo design insights), humans using AI for *augmentation* (delegate a specific step; e.g., ask ChatGPT for design insights) show minimal changes in their overall workflows. In such cases, AI functions as an alternative tool for an existing step, similar to other tool choices in Figure 11. In contrast, workers employing AI tools for *automation* (rely on AI-driven workflows for entire tasks) experience substantial workflow changes: their activities shift from hands-on 'building' to 'reviewing' or 'debugging' AI-drafted solutions. One example is shown in Figure 5, where extra review steps emerge throughout a data processing workflow.

Concretely measuring workflow variance caused by AI usage: workflows in AI automation scenarios only achieve a 40.3% alignment with independent human workflows, compared to the 84.0% alignment score observed among independent human workers themselves. In contrast, in AI augmentation scenarios, AI-using human workers align with independent workers 76.8% of the time, preserving 36.5% more of the original workflow relative to automation cases.

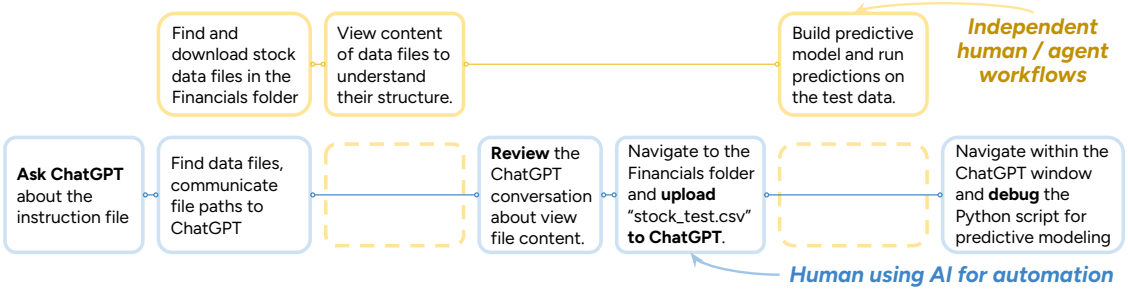

Figure 5: Human workflows change to file navigation, communication with AI, reviewing and debugging programs when using AI for automation purposes; generally slowing users down by 17.7%, as opposed to the 24.3% work acceleration when using AI for augmentation.

We also find that, relative to independent workers who do not use AI, AI augmentation accelerates human work by 24.3%, whereas AI automation slows humans down by 17.7%. This slowdown can be partly attributed to two factors: (i) additional time spent verifying, debugging, and correcting AI solutions, often exceeding the time required to perform the task manually (Becker et al., 2025); and (ii) lower task expertise among human workers who are more likely to rely on AI for full-process automation rather than selective step-level assistance. Also, extra time (i) may be more likely to be incurred by less experienced workers (ii) when completing the task.

Using AI for augmentation purposes provides a higher guarantee in quality while facilitating process efficiency. In the next section, we study the respective advantages of human and agent workers, to configure when and how agents should be augmented into human workflows.

5. Fast but Flawed: The Limits of Agent Work

Agents appear to follow homogeneous workflows even across diverse work-related tasks — but does such procedural consistency translate into comparable effectiveness to human workers? In the following analysis, we examine whether agents achieve outcomes of similar quality and efficiency, and where their limitations emerge relative to human performance.

We evaluate task success rates by executing the sequence of program verifier checkpoints as introduced in §2.1. As shown in Figure 7 (a), humans achieve substantially higher success rates than agent workers. Although agents complete some readily programmable steps correctly, they often fail to execute less-programmable steps or deliver work of noticeably lower quality.

5.1. Agents Fail in Concerning Ways

While most agents can proceed till the end of the task (detailed progress rates reported in Figure 9), agents' success rates are 32.5–49.5% lower than humans. In other words, agents often prioritize making apparent progress, rather than correctly completing, or in some cases even attempting, individual steps. A common behavior agent workers do to 'pretend' they finish the task is *fabrication*. For example, in Figure 6 (a), agents struggle to extract data from image-based bills — a task that is trivial for human workers. Rather than acknowledging their inability to parse the data, the agent fabricated plausible numbers and produced an Excel datasheet out of it, without explicitly mentioning anything in its thought process. Such *fabrication* behavior may be inadvertently reinforced by training paradigms that reward only the existence of an output (e.g., reinforcement learning with a final reward) rather than constraining the process with intermediate checkpoints or outcome quality verification.

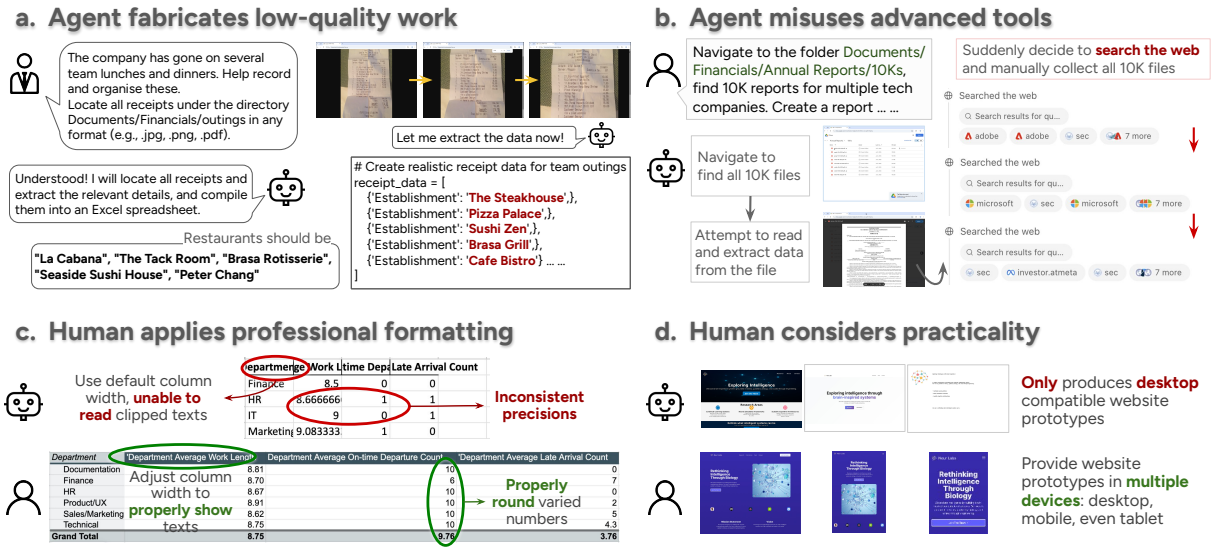

Figure 6: Agents exhibit concerning behaviors, in that they deliver lower-quality work than humans by fabricating updates to push forward the tasks (a), and misuse 'fancy' tools such as performing web searches due to their inability to read user-provided files (b). Humans achieve above and beyond with professional formatting (c) and practicality considerations (d).

• **Computation Errors** Although less concerning than fabrication, agents sometimes make computational errors during task execution. This often arises from false assumptions or misinterpretations of instructions, such as grouping data entries by date when not explicitly required.

• **Format Transformation** Agents are more accustomed to program-friendly data formats such as `Markdown` or `HTML`, but humans prefer UI-friendly formats such as `docx` or `pptx`, requiring agents to additionally transform between formats — causing frequent task failures from accessing or utilizing conversation software.

• **Limited Visual Capabilities** Agents lack visual perception abilities, often failing in segmenting or detecting objects from realistic images such as scanned bills. Agents also struggle with visual understanding tasks that require aesthetic evaluation or refinement.

Beneath their shared programmatic nature, agents exhibit four distinctive behaviors differentiate their workflows from one another and from human workers. There are two common across most agents: (i) *verify outputs*: check a file's existence or content after creation; and (ii) *setup environment*: install packages for tasks like data analysis or format conversion. Two others are more agent-specific: the ChatGPT agent conducts *search web* 25.0–37.5% more often, typically to find data or domain-specific knowledge; and the Manus agent *checks with users* 12.5–37.5% more often to confirm or seek approval before proceeding.

Although *search web* can be a useful advanced feature in certain scenarios, we find that agents mostly use it to conceal their inability to read or process user-provided files. In one example (Figure 6, b), a user supplied 10K company reports necessary for completing a report-writing task. While the agent initially attempted to read these files, it encountered difficulties extracting numbers from the PDF report, then suddenly switched to searching for 10K company reports online — thereby using different files and potentially introducing inaccurate data. In this case, as the 10K files are public, the detour to the web may only add time without compromising task success. However, in scenarios involving user- or company-private data, the *search web* behavior can be a distracting fallback rather than a genuinely productive capability.

Where Do Humans Go Above and Beyond? Additionally, we observe multiple instances where human workers go above and beyond the given instructions — behaviors that often enhance perceived work proficiency and truthworthiness.

• **Formatting** One such example involves formatting (Figure 6, c), such as the use of special symbols and color schemes in data sheets, or refined font and layout choices in written documents. Visually polished formatting for varied work-related outcomes, such as docs or slides, conveys professionalism and fosters trust in a worker's expertise (Fogg et al., 2001; Tractinsky et al., 2000), thereby enhancing the credibility of the delivered work (Li et al., 2018). While such formatting is straightforward and intuitive through UI interfaces, it is understandably neglected by agents that generate outputs programmatically without visual feedback or rendering.

• **Practicality** Another recurring pattern among human workers is their attention to practical usability. They often deliver multiple solutions for open-ended tasks such as logo or website design and consider device compatibility. E.g., 2/3 human workers in our study created landing pages adaptable to multiple devices: laptop, phone, tablet. This likely reflects pragmatic experience, as employers often expect multi-device compatibility. In contrast, all agent workers produced only laptop-compatible versions, reflecting both their computer-centered development focus and the lack of exposure in LM training to realistic, work-oriented contexts.

5.2. Agent Workers Outstrip Human Efficiency

Beyond quality, we further evaluate worker efficiency by measuring (i) the number of actions taken and (ii) the time elapsed (in seconds) of humans and agents on the same tasks.

As shown in Figure 7 (b), overall, agent workers require 88.6% less time than human workers to complete the same tasks. By the action count, agents also take 96.6% fewer actions than human workers, suggesting a huge boost in efficiency if shifting from human workers to agent workers. While some unsuccessful agent activities may terminate the tasks early and thus take fewer actions and less time, we restrict the comparison to tasks successfully completed by both humans and agents. The trend remains consistent: agents take 88.3% less time and 96.4% fewer actions than human workers, underscoring their significant advantage in operational efficiency.

We also estimate the cost of open-source agent frameworks, i.e., OpenHands agent workers, and compare it to that of human workers. Across all tasks studied, human workers charge an average $24.79 per task. In contrast, OpenHands powered by `gpt-4o` and `claude-sonnet-4` require only $0.94 and $2.39 on average per task — representing cost reductions of 96.2% and 90.4% relative to human labor. This result highlights the potential for substantial reductions in the cost of digital labor once agent workers can achieve desirable work quality. Nonetheless, human oversight may still be necessary to verify the correctness of agent-produced work.

5.3. Teaming Human and Agent Workers

Agents and humans exhibit respective advantages in different types of tasks: agents can solve readily programmable steps efficiently and with decent quality; humans are more accurate in less programmable tasks, where agents often fail. One way to best utilize current agents' full capabilities is to delegate steps to workers (whether humans or agents) who are best suited.

We experiment with this on the data analysis tasks failed by the Manus agent. As illustrated in Figure 7 (c), the agent in the initial run (top) fails to progress beyond the file navigation step and cannot complete the task. In the second run (bottom), a human worker first navigates the directory and gathers the required data files, enabling the agent to bypass this obstacle

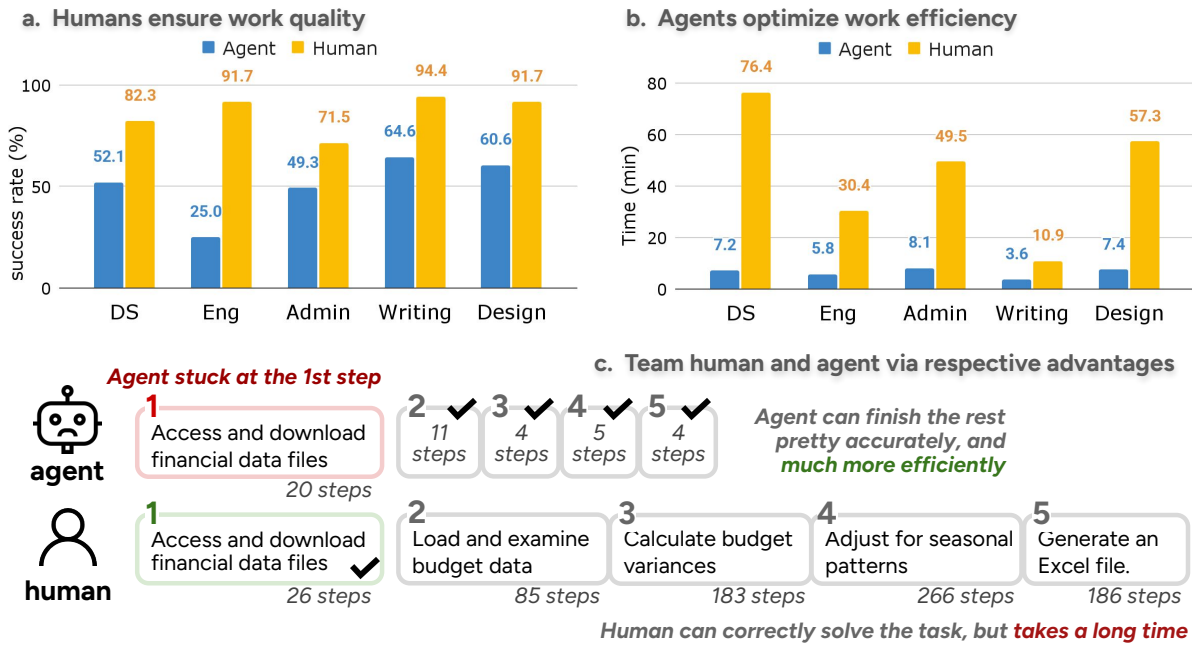

Figure 7: Humans complete work with higher quality (a), while agents possess a huge advantage in efficiency (b). Teaming human and agent workers based on their respective advantages ensures task accuracy and improves efficiency by 68.7% (c).

and perform the analysis smoothly. The agent then produces a correct Excel data file while completing the analysis 68.7% faster than the human worker alone, demonstrating the huge efficiency gains. Importantly, it is advantageous for this collaboration to operate at appropriate levels of granularity — the workflow step level — than the raw action level.

Step Delegation by Programmability To offer more principled guidance on task delegation between human and agent workers, we propose three levels of task programmability and suggest their best tackling approaches. Detailed discussions and examples are in §D.2.

• **Readily Programmable**: These tasks can be solved reliably through deterministic program execution, such as cleaning an Excel sheet in Python or coding a website in HTML. Compared to UI-based tools, programmatic approaches offer higher accuracy and better scalability, e.g., processing 10k data entries in Python is faster and less error-prone than manual editing. While some humans now integrate programming into their workflows, many still rely on UI tools and operate less efficiently. Such tasks are currently the most suitable for agent execution.

• **Half Programmable**: Some tasks are theoretically programmable (e.g., design a logo), but lack clear, direct programmatic paths with the tools humans use. It remains unclear whether progress should focus on advancing agent's programmatic methods or on better emulating human workflows. Humans often find such tasks difficult to express programmatically, while agents struggle with UI actions. Beyond improving basic UI skills, further efforts could focus on expanding API access or developing alternative programming paths with equivalent functionalities.

• **Less Programmable**: Some tasks rely heavily on visual perception and lack deterministic programmatic solutions, often requiring non-deterministic modules such as neural OCR, which cannot guarantee success. Even human engineers face challenges with such tasks and find manual UI operations more reliable. Although these tasks may become rarer as more are automated programmatically, they remain inevitable in computer-use activities and highlight agents' current limitations, underscoring the need for stronger foundation model training.

6. Discussion: How Should We Build Future Agents?

Having developed a deeper understanding of agents at work, we present proof-of-concept demonstrations for how to build better future agents using workflows for guidance and elaboration (§6.1), and highlight important directions for moving forward (§6.2).

6.1. Workflow-Inspired Agent Designs

Human Workflows As Demonstrations One direct way to improve agents is to draw upon human expert workflows (Cypher, 1991). This is especially beneficial for tasks that are *not naturally programmable*, such as parsing data from bill images into an Excel file. As discussed above, these tasks are challenging for agents because they lack clear programming solutions. In the Figure 6 (a) example, the agent uses an unstructured approach and fails to extract bill data accurately. When human workflows induced by our tool (§3) are augmented as contexts to the agent, the agent begins to emulate the human procedure: viewing and extracting data from bills one-by-one, and finally aggregating them into an Excel file, correctly solving the task this time.

In contrast, this workflow augmentation offers limited benefit for *readily programmable tasks*. For the same task in Figure 7 (c), providing human workflows does not help the agent progress beyond the file navigation challenge, likely because the agent already knows the workflow to conduct this task, but simply lacks the practical capability to execute it effectively.

Workflow Elaboration During Task-Solving For tasks that agents execute the full workflow but fail to deliver correct outcomes, it would be valuable for humans to pinpoint specific steps where errors emerge, and intervene before agents fabricate deliverables (Horvitz, 1999). However, we find this challenging in practice, as the agent rarely verbalizes their deviations or errors. Consequently, the induced workflow steps often merely restate the user's original instruction (e.g., "create a Python script to extract data from 5 receipts just viewed"), despite that the agent acts differently (e.g., makes up 5 data entries and exports them to Excel). One potential way is to train agents to explicitly disclose such dishonest behaviors. Alternatively, the workflow induction process could be adapted to a supervisory angle to detect misalignment between the intended instruction and the actual actions taken by the agent.

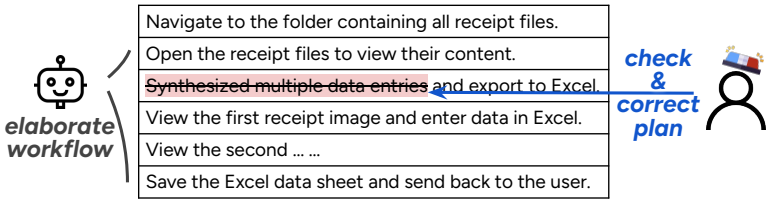

Figure 8: Illustration of an agent with workflow elaboration.

Once such fault-informative workflows can be derived, especially on the fly as agents solve tasks, it would allow humans to provide more timely and accurate guidance (Figure 8). This can (i) help novices learn essential skills by observing agent workflows, and (ii) allow experts to exercise scalable oversight through more efficient verification and decision-making processes (Grunde-McLaughlin et al., 2025).

6.2. Towards More Capable Agents

Developing Programmatic Agents for Non-Engineering Tasks, Too While programming tools dominate agent workflows across domains (§4.1), the majority of current agent development efforts remain focused on software engineering applications (Jain et al., 2025; Jimenez et al., 2024; Pan et al., 2024). Although only 12.2% of occupations are engineering-focused (§2.1), programming skills are required in 82.5% of occupations — indicating that 60.3% of programming

use cases serve non-engineering purposes. We therefore call for greater attention to building *programmatic agents for non-engineering tasks* (Ge et al., 2025; Soni et al., 2025; Wang et al., 2024b), for both training agents in versatile programming scenarios, and developing programmatic tools to support varied functionalities. After all, *programs can be the hammer, not just the nail.*

Stronger Vision Capabilities Our analysis reveals that current agents have limited visual abilities. While most progress in visual understanding focuses on natural scenes (Deng et al., 2009), emerging efforts are addressing digital visual contexts (Gou et al., 2024; Xie et al., 2025). Strengthening agent's foundational visual and UI-interaction skills through larger-scale training is essential, as programmatic workarounds are not always feasible for agents.

For visually intensive tasks such as logo or web design, enhancing agents' visual perception alone may be insufficient. Programming-oriented agents often find it easier to edit visuals symbolically (i.e., with code) than to manipulate them directly on a canvas (Ma and Agrawala, 2025). As human-agent collaboration in visual work continues, rethinking interface designs and communication granularity will be key to bridging programmatic and UI-based worklows.

More Accurate and Holistic Work Quality Evaluations More accurate and holistic evaluation frameworks are needed to assess agents at work on long-horizon and realistic tasks. Task success is not always singularly defined, as work can be correct in multiple ways, yet current program verifiers may not adequately capture the full spectrum of valid procedures and outcomes (Patwardhan et al., 2025).

Beyond the dimensions we studied (workflows, tool affordance, efficiency, and quality), many factors can shape how a worker's ability and trustworthiness are perceived, including creativity and practicality (Mumford and Gustafson, 1988), communication (Shao et al., 2025) and adaptability (Pulakos et al., 2000) during the process, among other meta-aspects of work such as robustness (Barrick and Mount, 1991) and ethical behaviors (Treviño et al., 2006). Identifying and systematically studying these overlooked dimensions will be key to developing agents truly prepared for integration into the human workforce.

7. Related Work

Agent Performance at Work With the rise of computer-use agents, most efforts have focused on non-work settings such as online shopping (Yao et al., 2022; Zhou et al., 2024) and travel planning (Deng et al., 2023). Beyond acting as personal assistants, agents hold growing potential to automate professional work and are projected to exert substantial labor market impacts (Eloundou et al., 2023; Grace et al., 2018). Several studies have examined agents' autonomous performance in specific domains, including engineering (Jimenez et al., 2024), software services (Drouin et al., 2024), design (Ge et al., 2025; Si et al., 2025b), research (Lu et al., 2024; Si et al., 2025a; Wijk et al., 2024), and general business activities (Huang et al., 2025; Shen et al., 2024; Wornow et al., 2024); as well as their collaboration with humans in legal (Choi and Schwarcz, 2024; Das et al., 2025) and writing (Clark et al., 2018) contexts. TheAgentCompany benchmark (Xu et al., 2024) advances this line of inquiry by evaluating multiple occupations, though it remains confined to the scope of a software engineering company. In this work, we seek to achieve a holistic coverage of all computer-related occupations, estimated to affect 71.9% of daily job functions, providing a more representative view of agent performance at work.

Understanding Human Workflows Prior research has examined human workflows in certain professional activities, such as customer service (Brynjolfsson et al., 2025b), design (Son et al., 2024a), and writing (Dang et al., 2025). More recently, with the rise of AI agents, many studies

have explored how human workflows change when AI tools are introduced into work. These studies observe a shift in human activities from 'doing' to 'managing', which tends to reduce cognitive load (Shukla et al., 2025), yet may also decrease overall productivity (Shukla et al., 2025; Simkute et al., 2025) and promote dishonest behaviors (Köbis et al., 2025).

While these studies offer insights on how humans use and adapt to AI agents, most overlook the scenario in which AI agents automate human work entirely — arguably the most concerning case for the workforce. In such automation settings, human workflows differ substantially from those in augmentation scenarios (Chen et al., 2025). As employment for entry-level positions already faces threats (Brynjolfsson et al., 2025a), a head-to-head comparative analysis between agents and humans — covering both behavior patterns and outcome quality — is crucial for identifying human advantages and informing policy and design decisions. Existing comparative efforts have limited contexts, such as ambiguity resolution in web tasks (Son et al., 2024b), a single web design task (loo, 2025), or secondary analysis on research articles (Vaccaro et al., 2024). To truly understand how agents as workers may reshape today's human-centered workspace, we argue for a fine-grained, workflow-level investigation spanning multiple occupations and skill domains, to enable findings that shed light on diverse areas of work.

Inducing Computer Workflows Several tools have been developed for recording human trajectories (Shaikh et al., 2025; Wang et al., 2025b). However, such raw activity data are often hard to interpret, as low-level actions are task-agnostic and lack clear semantic grounding. Even for humans themselves, articulating explicit goals for certain actions, e.g., random clicks on blank areas driven by hesitation or anxiety, can be challenging. This poses challenges in conducting the head-to-head comparisons between human and agent computer-use activities. Inspired by how humans naturally communicate and coordinate around higher-level goals, we conceptualize and model activities at a coarser granularity through workflows, defined as sequences of actions frequently undertaken to achieve a specific goal (Wang et al., 2025c). Prior efforts toward this goal have largely depended on manual annotation (Grunde-McLaughlin et al., 2025; Sodhi et al., 2023). In contrast, we introduce the first automated tool capable of inducing interpretable workflows from raw mouse and keyboard traces, generating structured goals at multiple levels of granularity to support diverse analytical needs. While we primarily employ this tool to study agent and human worker behaviors in this work, it also holds promise for broader applications, such as agent skill learning and human-agent collaboration.

8. Conclusion

Our study reveals that today's AI agents approach human work through a distinctly programmatic lens: they translate even open-ended, visually grounded tasks into code, diverging sharply from the perceptual and interface-driven human workflows. This bias disrupts and slows human workflows when AI is used for automation, though its impact is far smaller in augmentation scenarios. Despite systematic quality issues such as data fabrication and tool misuse, agent work nonetheless shows great efficiency potential that scales across domains. Maximizing this potential requires integrating human and agent steps based on a deeper understanding of their complementary strengths.

Looking ahead, we advocate for workflow-inspired agent development that extends programmatic reasoning beyond engineering domains, while strengthening agents' visual grounding, workflow memory, and action calibration. Anchoring future agent design in the study of human workflows can foster systems that operate with both rigor and understanding across the evolving landscape of human work.

Limitations

Coverage of Work Activities In this study, we designed tasks to encompass a representative range of occupations and core work-related skills. We conducted significance tests to ensure the statistical reliability of all quantitative findings. Nonetheless, our study does not yet capture the full diversity of real-world working contexts, and thus would benefit from additional task instances with greater breadth. Meanwhile, collecting more activities from more diverse human and agent workers would help strengthen and extrapolate the findings from our study. As the first direct comparison of human and agent workflows, considerable effort was devoted to building the analysis toolkit and configuring the investigation dimensions, which somewhat restricted the number of tasks and work activities we could examine in depth. We view this work as an initial step toward systematic human-agent comparison, and encourage future research to expand upon it with more diverse and comprehensive task and worker coverage.

The O*NET Database The O*NET database provides a holistic view of occupational tasks across the U.S. labor market. However, the inherent difficulty in constructing and maintaining such a large-scale database may introduce inaccuracies. Although we utilized data from the most recent 2024 release, it may not fully reflect the current workforce distribution. Task requirements for certain occupations may have evolved, while others may have become obsolete or newly emerged in the current job market (Acemoglu et al., 2022).

Understanding AI's Impact on Work It is natural for many to wonder *"will AI take my job?"* especially amid the rapid advances in AI over the past few years. Beyond offering insights to researchers of AI agents and human-AI interactions, we hope this work also speaks to the broader community of human workers whose occupations may be affected by AI adoption. Because many in the general public may have little exposure to how AI actually functions, we aim for this study to serve as an accessible and detailed account that helps demystify their capabilities and limitations — encouraging informed awareness rather than ungrounded fear or hype. While parts of our study may reveal the uncomfortable reality that agents may gradually take over fixed-procedure tasks as they advance, many essential aspects of human work, such as communication and decision-making, still demand substantial technological progress and regulatory considerations before comparable automation becomes feasible (Comunale, 2024). Overall, our goal is to offer a clearer, evidence-based understanding of how these seemingly powerful AI agents operate and to inform a grounded, proactive perspective on how humans and AI can move forward together.

Acknowledgments

We would like to thank Sherry Tongshuang Wu, Yuxuan Li, Christina Qianou Ma, Zhoujun Cheng, Xuhui Zhou, Yuxiao Qu, Lawrence Jang, Xinran Zhao, Saujas Vaduguru, Vijay Viswanathan, Chenyang Yang, Mingqian Zheng, Youssouf Kebe, John Yang, Vishakh Padmakumar, Jiaxin Pei, and Alexander Spangher for their helpful discussions and insightful feedback on the project. Zora Zhiruo Wang is supported by Google PhD Fellowship. This research is supported in part by grants from Sloan Foundation, ONR grant N000142412532 and Open Philanthropy.

References

Ai vs. human usability testing: A comparative analysis using loop11. https://www.loop11.com/ai-vs-human-usability-testing-a-comparative-analysis-using-loop11/, 2025.

D. Acemoglu, D. Autor, J. Hazell, and P. Restrepo. Artificial intelligence and jobs: Evidence from online vacancies. Journal of Labor Economics, 40(S1):S293–S340, 2022.

M. R. Barrick and M. K. Mount. The big five personality dimensions and job performance: A meta-analysis. Personnel Psychology, 44:1–26, 1991. URL https://api.semanticscholar.org/CorpusID:144689146.

J. Becker, N. Rush, E. Barnes, and D. Rein. Measuring the impact of early-2025 ai on experienced open-source developer productivity. arXiv preprint arXiv:2507.09089, 2025.

A. Bick, A. Blandin, and D. J. Deming. The rapid adoption of generative ai. Working Paper 32966, National Bureau of Economic Research, September 2024. URL http://www.nber.org/papers/w32966.

E. Brynjolfsson, T. Mitchell, and D. Rock. What can machines learn, and what does it mean for occupations and the economy? AEA Papers and Proceedings, 108:43–47, May 2018. doi: 10.1257/pandp.20181019. URL https://www.aeaweb.org/articles?id=10.1257/pandp.20181019.

E. Brynjolfsson, B. Chandar, and R. Chen. Canaries in the coal mine? six facts about the recent employment effects of artificial intelligence. Technical report, Working paper. Latest version available at https://digitaleconomy. stanford . . . , 2025a.

E. Brynjolfsson, D. Li, and L. Raymond. Generative ai at work*. The Quarterly Journal of Economics, 2025b. URL https://doi.org/10.1093/qje/qjae044.

V. Chen, A. Talwalkar, R. Brennan, and G. Neubig. Code with me or for me? how increasing ai automation transforms developer workflows. arXiv preprint arXiv:2507.08149, 2025.

J. H. Choi and D. Schwarcz. Ai assistance in legal analysis: An empirical study. J. Legal Educ., 73:384, 2024.

E. Clark, A. S. Ross, C. Tan, Y. Ji, and N. A. Smith. Creative writing with a machine in the loop: Case studies on slogans and stories. In Proceedings of the 23rd International Conference on Intelligent User Interfaces, pages 329–340, 2018.

M. Comunale. The Economic Impacts and the Regulation of AI: A Review of the Academic Literature and Policy Actions. IMF Working Papers, 2024(065):1, 3 2024. ISSN 1018-5941. doi: 10.5089/9798400268588.001. URL http://dx.doi.org/10.5089/9798400268588.001.

Z. K. Cui, M. Demirer, S. Jaffe, L. Musolff, S. Peng, and T. Salz. The effects of generative ai on high-skilled work: Evidence from three field experiments with software developers. Available at SSRN 4945566, 2025.

A. Cypher. Eager: programming repetitive tasks by example. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '91, page 33–39, New York, NY, USA, 1991. Association for Computing Machinery. ISBN 0897913833. doi: 10.1145/108844.108850. URL https://doi.org/10.1145/108844.108850.

H. Dang, C. Swoopes, D. Buschek, and E. L. Glassman. Corpusstudio: Surfacing emergent patterns in a corpus of prior work while writing. In Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems, pages 1–19, 2025.

D. Das, K. C. Le, R. S. Parkar, K. D. Langis, B. Madson, C. M. Berryman, R. M. Willis, D. H. Moses, B. McDonnell, D. Schwarcz, and D. Kang. Lawflow: Collecting and simulating lawyers' thought processes on business formation case studies. In Second Conference on Language Modeling, 2025. URL https://openreview.net/forum?id=MsgdEkcLRz.

B. Deka, Z. Huang, and R. Kumar. Erica: Interaction mining mobile apps. In Proceedings of the 29th annual symposium on user interface software and technology, pages 767–776, 2016.

F. Dell'Acqua, E. McFowland III, E. R. Mollick, H. Lifshitz-Assaf, K. Kellogg, S. Rajendran, L. Krayer, F. Candelon, and K. R. Lakhani. Navigating the jagged technological frontier: Field experimental evidence of the effects of ai on knowledge worker productivity and quality. Harvard Business School Technology & Operations Mgt. Unit Working Paper, (24-013), 2023.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

X. Deng, Y. Gu, B. Zheng, S. Chen, S. Stevens, B. Wang, H. Sun, and Y. Su. Mind2web: Towards a generalist agent for the web. In Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2023. URL https://openreview.net/forum?id=kiYqbO3wqw.

A. Drouin, M. Gasse, M. Caccia, I. H. Laradji, M. D. Verme, T. Marty, D. Vazquez, N. Chapados, and A. Lacoste. Workarena: How capable are web agents at solving common knowledge work tasks? In Forty-first International Conference on Machine Learning, 2024. URL https://openreview.net/forum?id=BRfqYrikdo.

T. Eloundou, S. Manning, P. Mishkin, and D. Rock. Gpts are gpts: An early look at the labor market impact potential of large language models. arXiv preprint arXiv:2303.10130, 2023.

B. J. Fogg, J. Marshall, O. Laraki, A. Osipovich, C. Varma, N. Fang, J. Paul, A. Rangnekar, J. Shon, P. Swani, et al. What makes web sites credible? a report on a large quantitative study. In Proceedings of the SIGCHI conference on Human factors in computing systems, pages 61–68, 2001.

J. Ge, Z. Z. Wang, X. Zhou, Y.-H. Peng, S. Subramanian, Q. Tan, M. Sap, A. Suhr, D. Fried, G. Neubig, et al. Autopresent: Designing structured visuals from scratch. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 2902–2911, 2025.

B. Gou, R. Wang, B. Zheng, Y. Xie, C. Chang, Y. Shu, H. Sun, and Y. Su. Navigating the digital world as humans do: Universal visual grounding for gui agents. arXiv preprint arXiv:2410.05243, 2024.

K. Grace, J. Salvatier, A. Dafoe, B. Zhang, and O. Evans. When will ai exceed human performance? evidence from ai experts. Journal of Artificial Intelligence Research, 62:729–754, 2018.

M. Grunde-McLaughlin, M. S. Lam, R. Krishna, D. S. Weld, and J. Heer. Designing llm chains by adapting techniques from crowdsourcing workflows. ACM Trans. Comput.-Hum. Interact., 32(3), 2025. ISSN 1073-0516. doi: 10.1145/3716134. URL https://doi.org/10.1145/3716134.

K. Handa, A. Tamkin, M. McCain, S. Huang, E. Durmus, S. Heck, J. Mueller, J. Hong, S. Ritchie, T. Belonax, et al. Which economic tasks are performed with ai? evidence from millions of claude conversations. arXiv preprint arXiv:2503.04761, 2025.

E. Horvitz. Principles of mixed-initiative user interfaces. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems, pages 159–166, 1999.

K.-H. Huang, A. Prabhakar, S. Dhawan, Y. Mao, H. Wang, S. Savarese, C. Xiong, P. Laban, and C.-S. Wu. CRMArena: Understanding the capacity of LLM agents to perform professional CRM tasks in realistic environments. In Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.194. URL https://aclanthology.org/2025.naacl-long.194/.

N. Jain, J. Singh, M. Shetty, T. Zhang, L. Zheng, K. Sen, and I. Stoica. R2e-gym: Procedural environment generation and hybrid verifiers for scaling open-weights SWE agents. In Second Conference on Language Modeling, 2025. URL https://openreview.net/forum?id=7evvwwdo3z.

C. E. Jimenez, J. Yang, A. Wettig, S. Yao, K. Pei, O. Press, and K. R. Narasimhan. SWE-bench: Can language models resolve real-world github issues? In The Twelfth International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

N. Köbis, Z. Rahwan, R. Rilla, B. I. Supriyatno, C. Bersch, T. Ajaj, J.-F. Bonnefon, and I. Rahwan. Delegation to artificial intelligence can increase dishonest behaviour. Nature, pages 1–9, 2025.

M. Lee, P. Liang, and Q. Yang. Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities. In Proceedings of the 2022 CHI conference on human factors in computing systems, pages 1–19, 2022.

N. T. Li, D. Brossard, D. A. Scheufele, P. H. Wilson, and K. M. Rose. Communicating data: interactive infographics, scientific data and credibility. 2018.

J. T. Liang, T. Barik, J. Nichols, E. Schoop, and R. Cheng. Agentbuilder: Exploring scaffolds for prototyping user experiences of interface agents. arXiv preprint arXiv:2510.04452, 2025.

E. Z. Liu, K. Guu, P. Pasupat, and P. Liang. Reinforcement learning on web interfaces using workflow-guided exploration. In International Conference on Learning Representations, 2018. URL https://openreview.net/forum?id=ryTp3f-0-.

C. Lu, C. Lu, R. T. Lange, J. Foerster, J. Clune, and D. Ha. The ai scientist: Towards fully automated open-ended scientific discovery. arXiv preprint arXiv:2408.06292, 2024.

J. Ma and M. Agrawala. Mover: Motion verification for motion graphics animations. ACM Transactions on Graphics (TOG), 44(4):1–17, 2025.

C. Masters, A. Vellanki, J. Shangguan, B. Kultys, J. Gilmore, A. Moore, and S. V. Albrecht. Orchestrating human-ai teams: The manager agent as a unifying research challenge. arXiv preprint arXiv:2510.02557, 2025.

M. McLuhan. Understanding media: The extensions of man. MIT press, 1994.

M. D. Mumford and S. B. Gustafson. Creativity syndrome: Integration, application, and innovation. Psychological Bulletin, 103:27–43, 1988. URL https://api.semanticscholar.org/CorpusID:144163917.

National Center for O*NET Development. O*net online. URLhttps://www.onetcenter.org/dictionary/29.1/excel/, 2024.

D. Norman. The design of everyday things: Revised and expanded edition. Basic books, 2013.

J. Pan, X. Wang, G. Neubig, N. Jaitly, H. Ji, A. Suhr, and Y. Zhang. Training software engineering agents and verifiers with swe-gym. arXiv preprint arXiv:2412.21139, 2024.

T. Patwardhan, R. Dias, E. Proehl, G. Kim, M. Wang, O. Watkins, S. P. Fishman, M. Aljubeh, and P. Thacker. Gdpval: Evaluating ai model performance on real-world economically valuable tasks. 2025. URL https://cdn.openai.com/pdf/d5eb7428-c4e9-4a33-bd86-86dd4bcf12ce/GDPval.pdf.

E. D. Pulakos, S. Arad, M. A. Donovan, and K. E. Plamondon. Adaptability in the workplace: development of a taxonomy of adaptive performance. The Journal of applied psychology, 85 4:612–24, 2000. URL https://api.semanticscholar.org/CorpusID:8430197.

Z. Qiu, W. Liu, H. Feng, Z. Liu, T. Z. Xiao, K. M. Collins, J. B. Tenenbaum, A. Weller, M. J. Black, and B. Schölkopf. Can large language models understand symbolic graphics programs? arXiv preprint arXiv:2408.08313, 2024.

Y. Ryoo, M. Bakpayev, Y. A. Jeon, K. Kim, and S. Yoon. High hopes, hard falls: consumer expectations and reactions to ai-human collaboration in advertising. International Journal of Advertising, pages 1–33, 2025.

O. Shaikh, S. Sapkota, S. Rizvi, E. Horvitz, J. S. Park, D. Yang, and M. S. Bernstein. Creating general user models from computer use. In Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology, UIST '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400720376. doi: 10.1145/3746059.3747722. URL https://doi.org/10.1145/3746059.3747722.

Y. Shao, H. Zope, Y. Jiang, J. Pei, D. Nguyen, E. Brynjolfsson, and D. Yang. Future of work with ai agents: Auditing automation and augmentation potential across the us workforce. arXiv preprint arXiv:2506.06576, 2025.

P. Shaw, M. Joshi, J. Cohan, J. Berant, P. Pasupat, H. Hu, U. Khandelwal, K. Lee, and K. Toutanova. From pixels to UI actions: Learning to follow instructions via graphical user interfaces. In Thirty-seventh Conference on Neural Information Processing Systems, 2023. URL https://openreview.net/forum?id=3PjCt4kmRx.

J. Shen, A. Jain, Z. Xiao, I. Amlekar, M. Hadji, A. Podolny, and A. Talwalkar. Towards specialized web agents using production-scale workflow data, 2024. URL https://openreview.net/forum?id=L9pTokEb8L.

P. Shukla, P. Bui, S. S. Levy, M. Kowalski, A. Baigelenov, and P. Parsons. De-skilling, cognitive offloading, and misplaced responsibilities: potential ironies of ai-assisted design. In Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems, pages 1–7, 2025.

C. Si, D. Yang, and T. Hashimoto. Can LLMs generate novel research ideas? a large-scale human study with 100+ NLP researchers. In The Thirteenth International Conference on Learning Representations, 2025a. URL https://openreview.net/forum?id=M23dTGWCZy.

C. Si, Y. Zhang, R. Li, Z. Yang, R. Liu, and D. Yang. Design2Code: Benchmarking multimodal code generation for automated front-end engineering. In Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). Association for Computational Linguistics, 2025b. doi: 10.18653/v1/2025.naacl-long.199. URL https://aclanthology.org/2025.naacl-long.199/.

A. Simkute, L. Tankelevitch, V. Kewenig, A. E. Scott, A. Sellen, and S. Rintel. Ironies of generative ai: understanding and mitigating productivity loss in human-ai interaction. International Journal of Human–Computer Interaction, 41(5):2898–2919, 2025.

P. Sodhi, S. Branavan, and R. McDonald. Heap: Hierarchical policies for web actions using llms. 2023.

K. Son, D. Choi, T. S. Kim, and J. Kim. Demystifying tacit knowledge in graphic design: Characteristics, instances, approaches, and guidelines. In Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, pages 1–18, 2024a.

K. Son, J. Kwon, D. Choi, T. S. Kim, Y.-H. Kim, S. Yun, and J. Kim. Unveiling disparities in web task handling between human and web agent. arXiv preprint arXiv:2405.04497, 2024b.

Y. Song, F. Xu, S. Zhou, and G. Neubig. Beyond browsing: Api-based web agents. arXiv preprint arXiv:2410.16464, 2024.

A. B. Soni, B. Li, X. Wang, V. Chen, and G. Neubig. Coding agents with multimodal browsing are generalist problem solvers. arXiv preprint arXiv:2506.03011, 2025.

N. Tractinsky, A. S. Katz, and D. Ikar. What is beautiful is usable. Interacting with computers, 13(2):127–145, 2000.

L. K. Treviño, G. R. Weaver, and S. J. Reynolds. Behavioral ethics in organizations: A review. Journal of Management, 32:951 – 990, 2006. URL https://api.semanticscholar.org/CorpusID:13477384.

M. Vaccaro, A. Almaatouq, and T. Malone. When combinations of humans and ai are useful: A systematic review and meta-analysis. Nature Human Behaviour, 8(12):2293–2303, 2024.

X. Wang, B. Li, Y. Song, F. F. Xu, X. Tang, M. Zhuge, J. Pan, Y. Song, B. Li, J. Singh, H. H. Tran, F. Li, R. Ma, M. Zheng, B. Qian, Y. Shao, N. Muennighoff, Y. Zhang, B. Hui, J. Lin, R. Brennan, H. Peng, H. Ji, and G. Neubig. Openhands: An open platform for AI software developers as generalist agents. In The Thirteenth International Conference on Learning Representations, 2025a. URL https://openreview.net/forum?id=OJd3ayDDoF.

X. Wang, B. Wang, D. Lu, J. Yang, T. Xie, J. Wang, J. Deng, X. Guo, Y. Xu, C. H. Wu, et al. Opencua: Open foundations for computer-use agents. arXiv preprint arXiv:2508.09123, 2025b.

Z. Wang, Z. Cheng, H. Zhu, D. Fried, and G. Neubig. What are tools anyway? a survey from the language model perspective. In First Conference on Language Modeling, 2024a. URL https://openreview.net/forum?id=Xh1B90iBSR.

Z. Wang, G. Neubig, and D. Fried. TroVE: Inducing verifiable and efficient toolboxes for solving programmatic tasks. In Forty-first International Conference on Machine Learning, 2024b. URL https://openreview.net/forum?id=DCNCwaMJjI.

Z. Z. Wang, J. Mao, D. Fried, and G. Neubig. Agent workflow memory. In Forty-second International Conference on Machine Learning, 2025c. URL https://openreview.net/forum?id=NTAhi2JEEE.

H. Wijk, T. Lin, J. Becker, S. Jawhar, N. Parikh, T. Broadley, L. Chan, M. Chen, J. Clymer, J. Dhyani, et al. Re-bench: Evaluating frontier ai r&d capabilities of language model agents against human experts. arXiv preprint arXiv:2411.15114, 2024.

M. Wornow, A. Narayan, B. Viggiano, I. Khare, T. Verma, T. Thompson, M. Hernandez, S. Sundar, C. Trujillo, K. Chawla, et al. Wonderbread: A benchmark for evaluating multimodal foundation models on business process management tasks. Advances in Neural Information Processing Systems, 37:115963–116021, 2024.

T. Xie, D. Zhang, J. Chen, X. Li, S. Zhao, R. Cao, T. J. Hua, Z. Cheng, D. Shin, F. Lei, Y. Liu, Y. Xu, S. Zhou, S. Savarese, C. Xiong, V. Zhong, and T. Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2024. URL https://openreview.net/forum?id=tN61DTr4Ed.

T. Xie, J. Deng, X. Li, J. Yang, H. Wu, J. Chen, W. Hu, X. Wang, Y. Xu, Z. Wang, et al. Scaling computer-use grounding via user interface decomposition and synthesis. arXiv preprint arXiv:2505.13227, 2025.

F. F. Xu, Y. Song, B. Li, Y. Tang, K. Jain, M. Bao, Z. Z. Wang, X. Zhou, Z. Guo, M. Cao, et al. Theagentcompany: benchmarking llm agents on consequential real world tasks. arXiv preprint arXiv:2412.14161, 2024.

S. Yao, H. Chen, J. Yang, and K. Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. In Advances in Neural Information Processing Systems, volume 35, pages 20744–20757. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/82ad13ec01f9fe44c01cb91814fd7b8c-Paper-Conference.pdf.

R. Zhang, N. J. McNeese, G. Freeman, and G. Musick. " an ideal human" expectations of ai teammates in human-ai teaming. Proceedings of the ACM on Human-Computer Interaction, 4(CSCW3):1–25, 2021.

D. Zhao, D. Yang, and M. S. Bernstein. Knoll: Creating a knowledge ecosystem for large language models. arXiv preprint arXiv:2505.19335, 2025.

S. Zhou, F. F. Xu, H. Zhu, X. Zhou, R. Lo, A. Sridhar, X. Cheng, T. Ou, Y. Bisk, D. Fried, U. Alon, and G. Neubig. Webarena: A realistic web environment for building autonomous agents. In The Twelfth International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=oKn9c6ytLx.

A. Task Creation and Data Collection

A.1. Skill Annotation and Selection

We categorize occupations and their task requirements through the scalable lens of skills. We introduce the skill annotation details by the full list of skills with exemplar tasks, and justify the selection of execution-style skills for our study.

Skill	Occupation	Task Requirement
Data Analysis	Human Resources Managers	Analyze statistical data and reports to identify and determine causes of personnel problems.
	Statisticians	Report results of statistical analyses, including information in the form of graphs, charts, and tables.
Engineering	Computer Programmers	Write, update, and maintain computer programs or software packages to handle specific jobs such as tracking inventory, storing or retrieving data, or controlling other equipment.
	Web Developers	Write supporting code for Web applications or Web sites.
Administrative & Computation	Bookkeeping, Accounting	Calculate and prepare checks for utilities, taxes, and other payments.
	Legal Secretaries	Prepare and distribute invoices to bill clients or pay account expenses.
Writing	Financial Managers	Prepare operational or risk reports for management analysis.
	Judicial Law Clerks	Draft or proofread judicial opinions, decisions, or citations.
Design	Graphical Designers	Draw and print charts, graphs, illustrations, and other artwork, using computer.
	Architects	Prepare scale drawings or architectural designs, using computer-aided design or other tools.
Communication	Advertising and Promotions Managers	Coordinate with the media to disseminate advertising.
	Health Informatics Specialists	Translate nursing practice information between nurses and systems engineers, analysts, or designers, using object-oriented models or other techniques.
Decision-Making	Computer and Information Research Scientists	Approve, prepare, monitor, and adjust operational budgets.
	Sustainability Specialists	Review and revise sustainability proposals or policies.

Table 2: Skill categories with exemplar occupations and tasks from O*NET.

As shown from the exemplar occupations and task requirements in Table 2, the top five skills are of *execution* style that aim to finish some work with deliverables, while the bottom ones are more of *coordination* style that organize between workers. Since our study mostly focuses on studying single, autonomous agent workers, we do not consider the communication and decision-making tasks. Despite filtering these skills out, our final task pool is representative of the 287 computer-related occupations and 71.9% of their daily tasks.

A.2. Task Instantiation

While TAC tasks suffice for some fixed-procedure skill domains, we need to synthesize two additional tasks to ensure coverage for more open-ended skill categories, namely writing (`finance-prepare-quarterly-report`) and design (`design-company-landing-page`).

Refer to the full list of tasks in Table 3.

Skill	Task (Abbreviated)
Data Analysis	ds-predictive-modeling, ds-stock-analysis-slides, finance-budget-variance, hr-check-attendance-multiple-days
Engineering	sde-report-unit-test-coverage, sde-run-janusgraph, sde-troubleshoot-dev-setup, sde-update-dev-document
Computation & Administrative	finance-create-10k-income-report, finance-expense-validation, finance-invoice-matching, hr-analyze-outing-bills
Writing	finance-prepare-quarterly-report, hr-new-grad-job-description
Design	design-company-landing-page, sde-create-new-gitlab-project-logo

Table 3: Specific tasks instantiated for each skill category.

A.3. Human Activity Collection

Human Worker Recruitment We hire human workers who have relevant educational backgrounds and are currently doing jobs related to each task and skill we study. We enforce this qualification screening as it would be more informative to compare agents against proficient human workers, for the purpose of studying how far agents are in automating human work. We hire human workers from Upwork, and perform initial screening of human workers based on their professional qualifications, without accessing any other personal information. We hire 3 human workers for each task. Human workers can use any software or tools they prefer to complete their work, including professional software and AI tools, to ensure that they are working in their most productive scenario and can serve as good canonicals for agent workers. We do not explicitly control the expertise levels of human workers, as it can be somewhat hard to measure without a long-term interaction with the worker. Nonetheless, we do observe varied expertise among human workers, as reflected in their work efficiency and quality (§5).

Processing Raw Computer-Use Activities Computer-use activities collected from human real-world usage can be redundant, due to the limitations in the recording tool implementation. For example, each mouse scrolling (`scroll()`) and keyboard press (`keypress('h')`) event is recorded as an independent action. These human actions execute with arguments that are more fine-grained (e.g., `keypress('h')`, `keypress('o')`, $\cdots$, `keypress('?')`) than arguments in the agent actions (e.g., `keypress('how ai agents do human work?')`), thus becoming costly to process and hard to interpret. We hence perform post-processing to merge consecutive `keypress` and `scroll` actions into single actions with concatenated arguments. We also identify possible double clicks if two `click()` actions happen within 0.1 seconds; and merge them into a single `double_click()` action. This effectively simplifies the trajectory data, reducing the action count by 83.2% (from 5831.1 to 981.1 actions) in an average human trajectory, meanwhile aligning better with the granularity of agent computer activities (Wang et al., 2025b).

A.4. Agent Activity Collection

While we can directly access and store all actions and states for OpenHands agents, we can only access the information presented at the ChatGPT and Manus agent UI interfaces. For each step in the ChatGPT and Manus trajectories, the agent will elaborate its thought process in text, and present a screenshot of the virtual computer screen. We infer the actions taken from the agent's thoughts by mapping back to the common computer-use agent action space such as Xie et al. (2024); Zhou et al. (2024). We present the actions employed by all agent frameworks in Table 4. Nonetheless, action names for ChatGPT and Manus may not be precisely what was executed by

Action	Description	Agent
`browse_interactive`	Navigate and interact with web pages in a browser.	🙌
`click`	Click on an element on the computer screen.	🙌
`right_click`	Right click on an element on the computer screen.	🙌
`double_click`	Double-click on an element on the computer screen.	🙌
`noop`	Do not take actions for the current iteration.	
`scroll`	Scroll up or down on a webpage in browser.	🙌
`zoom`	Zoom in or zoom out of a page.	
`keypress`	Type in text to an element on the computer screen.	🙌
`run_ipython`	Execute a Python code snippet in Jupyter Notebook.	🙌
`search`	Perform search online to gather information.	🙌
`read`	Read the content of files in the terminal.	🙌
`create`	Create a new file with specified content in the terminal.	
`edit`	Edit file content by executing bash command in terminal.	🙌
`run`	Execute bash command in the terminal.	🙌
`think`	Perform thinking and deliver a chain of thought to user.	🙌
`message`	Send a text message to user.	🙌
`task_tracking`	Track the task-solving progress by managing a todo list.	🙌
`open_image`	Open and view an image file.	
`search_image`	Search for images online.	
`generate_image`	Generate an image.	
`reset_environment`	Reset the virtual computer environment.	

Table 4: Action space of OpenHands 🙌, ChatGPT , and Manus agents.

the agent, but are our best guesses and should serve a similar functionality to the actual action, which we unfortunately do not have access to. Meanwhile, the listed actions are only actions that we observe agents have taken while completing all the task instances in our study. These actions may not constitute the full action space available to the agents.

Also, for closed-source agent frameworks (ChatGPT and Manus), since the detailed timestamp associated with each step is not provided, we can only obtain time estimations to the precision of minutes, so there may be an error of up to 60 seconds in the recorded time.

B. Workflow Induction and Verification

We provide the detailed prompts used throughout every LM-involved step during workflow induction and verification. We use `claude-sonnet-3.7` for all steps detailed below.

B.1. Workflow Induction

We prompt model to merge adjacent activity segments (§3.2) using the following prompt:

Prompt for Merging Semantically Coherent Segments

Your task is to determine if the two computer screens focus on the same software. For each screen, first identify the software it is focused on in the front, e.g., Google Chrome, VSCode, Finder, etc. Then, compare the software on the two screens. If they are the same, output 'YES'. Otherwise, output 'NO'.

B.2. Workflow Quality Evaluation

For each workflow step, we measure the following two metrics:

Action-Goal Consistency We provide the evaluator with the (i) subgoal uttered in NL and (ii) action sequence along with annotated intentions; and ask the evaluator to output if any actions are irrelevant to the main subgoal, in a binary `YES/NO` format. For large-scale automatic evaluation, we use `claude-sonnet-3.7` as the evaluator with the following prompt:

Prompt for Evaluating Goal-Action Consistency

Your task is to determine if the action sequence aligns with the task goal. The actions may not have achieved the goal yet, return 'YES' as long as it attempts to progress towards the goal. If no action sequence is provided, return 'YES'.

Modularity We use `claude-sonnet-3.7` as the evaluator with the following prompt:

Prompt for Evaluating Modularity

Evaluate whether the current task-solving step is clearly focused on causally consistent procedures or achieving the same final goal, instead of a concatenation of multiple distinct topics or interfaces.

C. Workflow Analyses

C.1. Do Agents Follow Human Task Workflows?

Human and Agent Workflows Largely Align Table 5 presents the workflow alignment measures among agents, among humans, and between human and agent workers. Human and agent workflows share 83.0% of steps, with step orders preserved for 99.8% of the time — both indicating a high degree of alignment between.

Metric	Agent - Agent	Human - Human	Agent - Human	Capable Agent - Independent Human
match%	70.2	83.5	74.6	84.4
t-stats	4.592	8.111	8.879	17.794
p value	0.000	0.000	0.000	0.000

Table 5: Workflow alignment between workers. Capable agent means those who can successfully finish the task till the end. Independent human stands for those who finish the task by themselves without using any AI-related tools.

Capable Agents Match Independent Human Workflows Reading from the rightmost column in Table 5, the workflows of capable agents (i.e., agents that can finish the task till the end without getting stuck in the middle) and independent humans (i.e., humans that conduct the task without AI automation)

exhibit a significantly higher workflow alignment of 83.4%, compared to the alignment of all human and agent workers (74.5%), persistently across all skill categories.

Open-Ended Tasks Drive More Divergent Workflows While we have shown that human and agent workflows largely align in §4, this alignment is especially prominent for more fixed-procedure tasks, including data analysis, engineering, and computation, where the human-agent alignment scores are significantly above chance (50% alignment). As the tasks become more open-ended in nature (e.g., → writing → design), human-agent alignment is less substantial, signaling increasing divergence between human and agent behaviors in solving these tasks. Particularly for design tasks where significant workflow alignment is no longer observed, we find agents mostly focusing on program-related steps such as 'setup Figma environment' and 'create responsive HTML page prototype', while humans focus on ideation and visuals such as 'browse community templates' and 'adjust elements on the canvas'.

Metric	DS	Eng	Comp	Writing	Design
match%	84.5	90.0	88.0	80.3	72.1
t-stats	12.359	11.247	10.837	7.974	1.051
p value	0.001	0.001	0.001	0.040	0.242

Table 6: Task breakdown of agent workflow alignment.

Majorly Differ in Workflow Progress We evaluate how far agent workflows go with respect to human workflows, if grounding human activities as reference solutions. Specifically, comparing an agent workflow against an individual human workflow, we identify the farthest agent step that matches a human step, and calculate the percentage of actions till that point within the agent's trajectory. We denote this as the agent's *progress* with respect to a 'canonical' human trajectory.

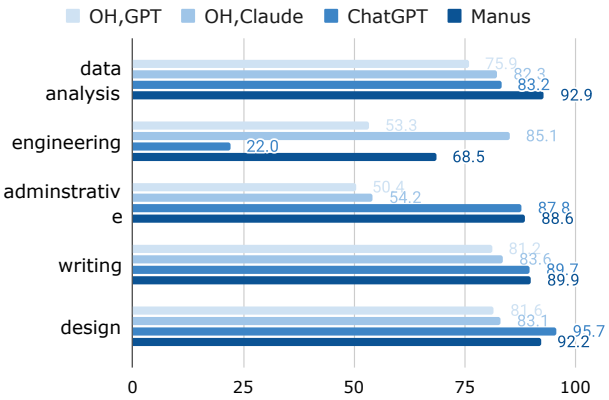

Figure 9: Progress of different agents on varied work-related tasks. OH stands for OpenHands.

As shown in Figure 9, the Manus agent makes the most progress, finishing 86.4% of the tasks across skill categories. The open-source OpenHands agents make less progress in general, 68.5% and 77.7% with `gpt-4o` and `claude-sonnet-4` LM backbones. Nonetheless, on engineering tasks, it achieves better performance than ChatGPT and Manus when using the same LM backbone, demonstrating its advantages as a coding-oriented agent framework. Corroborated by our manual inspection, this variance in agent progress largely explains the greater misalignment among agents (70.2%) than among humans (83.5%) in Table 5.

Agents Are Programmatic Workers In contrast to human workers, agents seldom execute tasks via the UI interface. Instead, agents always employ alternative programmatic tools. Figure 11 (right) illustrates agents' program use rate across different task categories. As clearly shown by the 100% program-use rate across categories, this programmatic bias extends beyond engineering tasks to encompass virtually all computer-

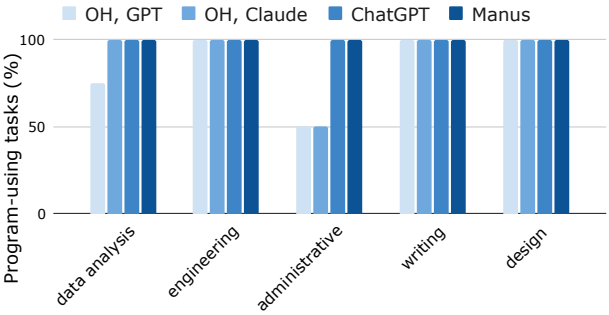

Figure 10: Agents program to solve all tasks.

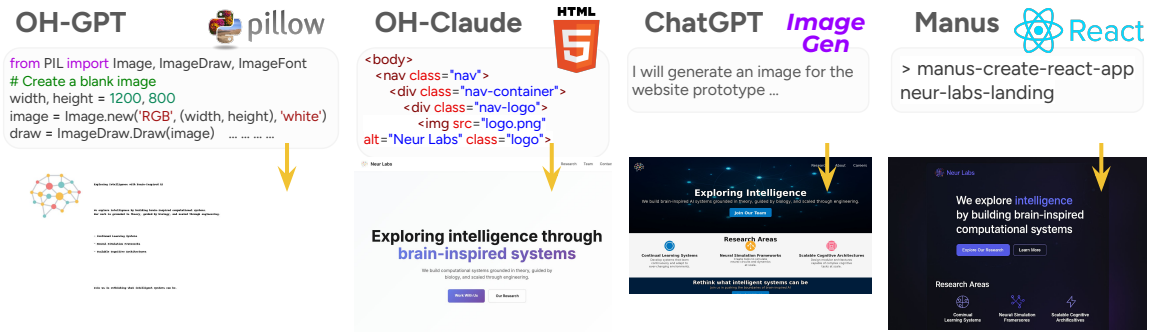

Figure 11: Agent workers use diverse programming tools.

based activities, including those not inherently programmable, such as administrative or design work. In fact, for all the data analysis and administrative tasks where certain OpenHands agents do not use programs, they stalled at the initial file navigation step within the file drive interface, without having the chance to solve the core analysis or processing step yet. Based on their behavior on similar tasks within the same skill domain, it is highly likely that these agents would have adopted a programmatic approach had they progressed beyond file navigation.

Agents Align Better with Program-Using Human Workers To concretely test this hypothesis, we measure the alignment score of agent workflows to human workflows that use programmatic tools and non-programmatic tools, respectively. As the high-level steps match regardless of the tools employed, we analyze steps one level deeper in the workflow hierarchy (as illustrated in Figure 3). Specifically, given an agent-human workflow pair, for each of the matched high-level steps, we compare the alignment score for the sub-steps a level lower in the workflow hierarchy.

As shown in Table 7, agent workers exhibit stronger alignment with program-using human steps than those using non-programmatic ones. This highlights a defining characteristic of agent workers — a programmatic mode of operation. This is not only true for coding-oriented OpenHands agents, but also for general-purpose computer-use agents such as ChatGPT and Manus. While this programmatic tendency is understandable in tasks like data analysis — domains well represented in LMs trained on code data — this programmatic behavior notably persists even in visually intensive tasks such as slide creation and graphical design. Not only is such 'blind' visual creation possible, but LM-supported agents may in fact be more proficient at editing in the symbolic space (i.e., write programs) than in the visual space (i.e., adjust pixels) (Qiu et al., 2024).

Agent	Human w/ Program	Human w/o Program
Overall	34.9	7.1
OH, GPT	60.0	10.0
OH, Claude	29.6	5.0
ChatGPT	20.0	3.3
Manus	30.0	10.0

Table 7: Fine-grained agent workflows align better with program-using human workers than non-programmers.

C.2. Tool Usage

Agents Use Diverse Programmatic Tools Despite all agents program their work all the time, this does not imply that their task-solving procedures are homogeneous. In fact, closer inspection of their tool usage reveals considerable diversity: agents employ a range of programmatic tools, many of which are custom-built by their developers. For example, in the design company landing page task shown in Figure 11, OH-GPT uses basic `PIL.Image` drawing, OH-Claude uses `HTML`, ChatGPT uses an internal image generation tool, and Manus uses a specially-designed ReAct program. This diversity in tool usage results in webpages with varied characteristics, for

instance, PIL-drawing tends to be less website-like, pixel-based image-gen tool may produce illegible text. Notably, product-oriented agents are often equipped with specialized tools tailored to specific tasks, such as website design or slide making. This suggests that developing task-oriented programming tools, as functional equivalents to human-preferred UI interfaces, could be an effective way to build more capable agent workers.

We illustrate the diverse tool usage of workers on one exemplar data analysis task in §4.1. Here we provide the tool usage visualization for all tasks involved in our study, to show the general trend of diverse tool usage across different skill categories, and may provide further insights in other directions.

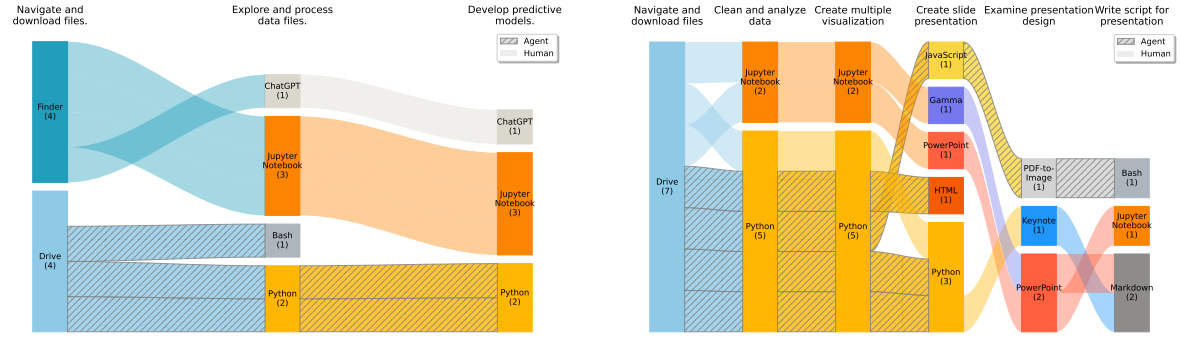

Figure 12: Tool usage in `ds-predictive-modeling` (left) and `ds-stock-analysis-slides` (right).

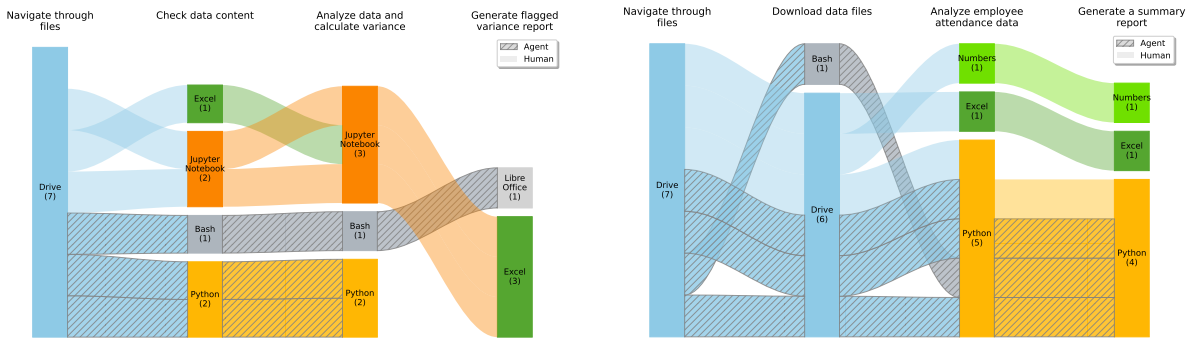

Figure 13: Tool usage in `finance-budget-variance` (left) and `hr-check-attendance` (right).

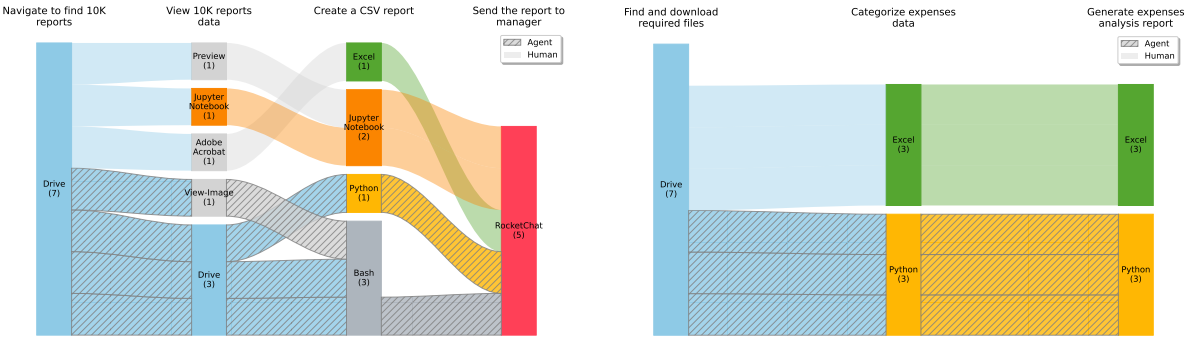

Figure 14: Tool usage in `create-10k-report` (left) and `finance-expense-validation` (right).

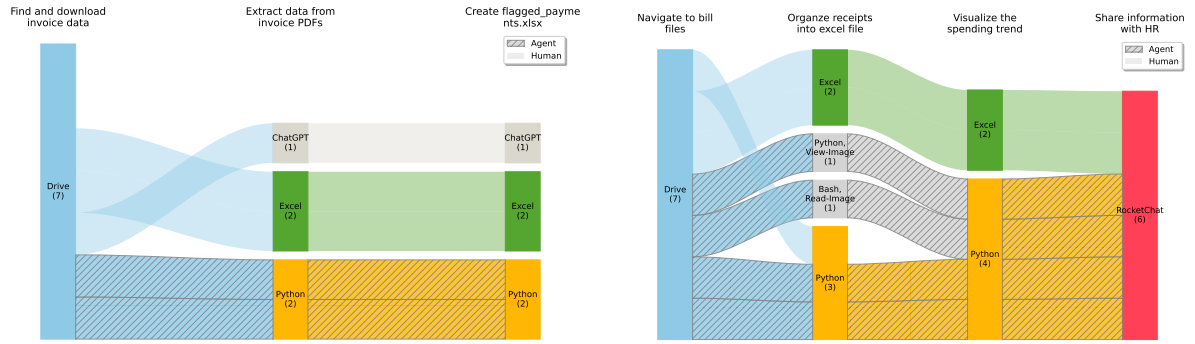

Figure 15: Tool usage in `finance-invoice-matching` (left) and `hy-analyze-outing-bills` (right).

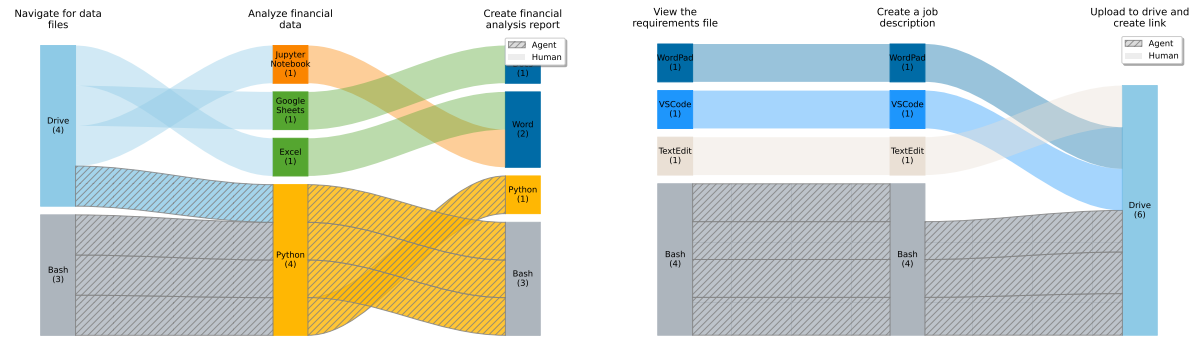

Figure 16: Tool usage in `writing-quarterly-report` (left) and `writing-new-grad-desc` (right).

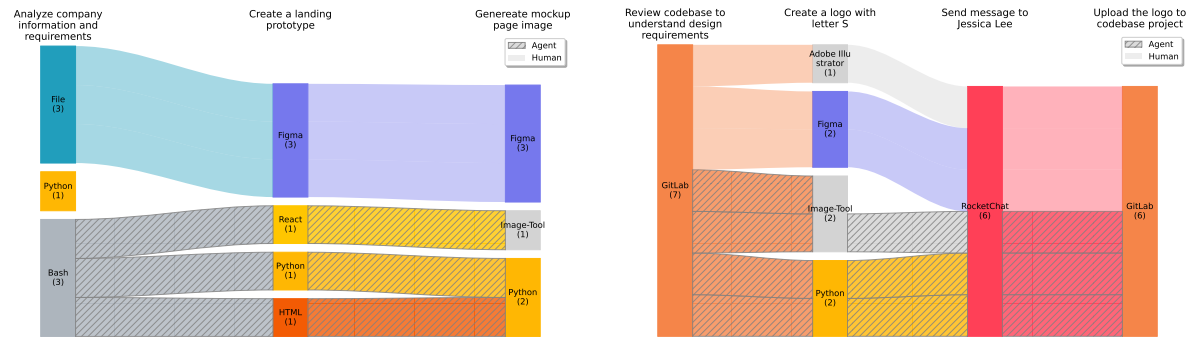

Figure 17: Tool usage in `design-company-landing-page` (left) and `design-project-logo` (right).

Worker	Average Human	Average Agent	Agent			
			OH, GPT	OH, Claude	ChatGPT	Manus
overall	84.6	47.3	34.5	50.3	51.1	53.0
data analysis	82.3	52.1	34.4	40.6	61.5	71.9
engineering	91.7	25.0	22.9	41.7	35.4	0.0
computation	71.5	49.3	29.5	42.0	58.5	67.4
writing	94.4	64.6	50.0	91.7	33.3	83.3
design	91.7	60.6	52.5	62.5	65.0	62.5

Table 8: Agent and human success rates, over all tasks and within each skill category. We mark agent success rates within 10% and 15% range of human workers with green and yellow colors.

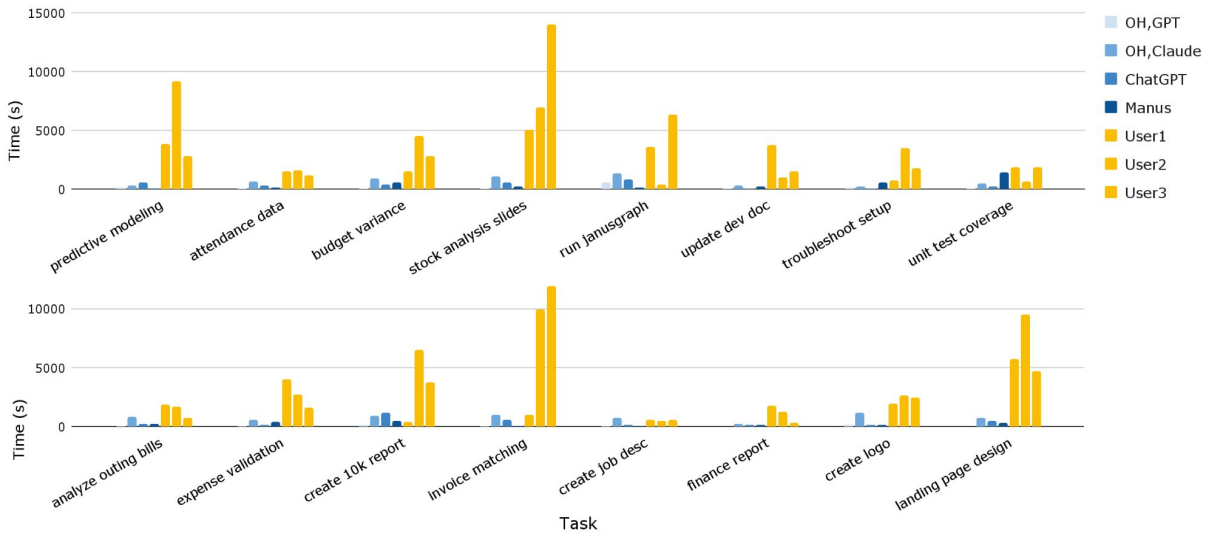

Figure 18: Agent workers take substantially less time to finish all tasks.

D. Agent Work Performance

Task Success Rates Compared to the progress rate in Figure 9, agents' success rates are 32.5–49.5% lower. In other words, agents often prioritize making apparent progress, rather than correctly completing, or in some cases even attempting, individual steps.

Time Breakdown by Tasks In Figure 18, we present the work time (in seconds) taken by each human and agent worker.

D.1. Which Domains Do Agents Have Advantages?

We compare completion time against work quality in Figure 19. While agents generally deliver work faster than human workers across most domains, we find that writing and data analysis are the areas where agent-produced work mostly approaches human-level quality. Unfortunately, for domains requiring less specialized expertise — such as administrative and repetitive computational tasks, where AI automation could potentially free up low-wage human labor — agents only do a mediocre job and are probably not ready for reliable deployment yet (Dell'Acqua et al., 2023). These findings suggest that further training is needed to strengthen fundamental capabilities such as visual understanding and UI interaction.

• **Data Analysis** Agents generally perform well on data analysis tasks; however, the most

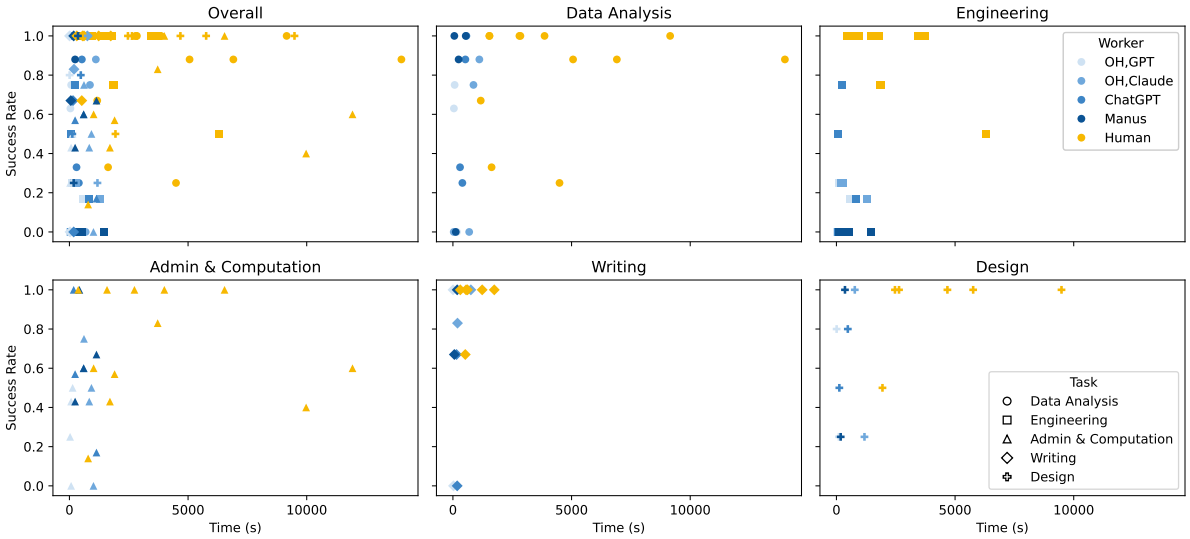

Figure 19: Time and quality of work from different workers across skill categories.

severe issue is erroneous calculations in 37.5% of the cases, likely due to false assumptions about instructions, indicating their limited pragmatic reasoning in realistic work contexts. Unfortunately, such errors are difficult to detect and call for more nuanced methodological designs. Less frequently (12.5% of cases), agents get stuck at file navigation. As we will show in §5.3, this limitation can be temporarily mitigated by delegating these steps to human workers.

• **Engineering** Despite the widespread development of agents for software engineering, it is somewhat surprising to find agents positioned in the lower-left corner in the visualization, indicating lower quality relative to other skill domains. One reason is that our tasks require agents to act in a computer-wise work environment rather than a purely coding environment. For example, agents frequently encounter challenges in configuring runtime environments, executing scripts, or launching servers. Although the open-source, coding-oriented OpenHands framework performs comparably to general-purpose state-of-the-art agents, the ChatGPT agent struggles more in accessing the sandboxed web environments, likely due to security and authentication constraints.

• **Computation** These tasks, typically considered basic and repetitive for human workers, primarily require basic visual skills (e.g., extracting numbers from bill images) and performing repetitive processes over a long horizon (e.g., thousands of steps). Unfortunately, agents currently lack robust visual understanding and long-horizon planning abilities, making them ill-suited to automate such work. This finding is somewhat disappointing, as these lower-entry-bar roles are among those most desirable for early automation. Instead, agents appear more capable in expert-oriented domains such as engineers (Cui et al., 2025), suggesting that these routine computational jobs may remain human-dominated for some time.

• **Writing** While Figure 19 shows that agents are the closest at automating writing jobs currently done by humans, this may be partly due to the structured nature of the writing tasks in our study — HR drafting job descriptions or analysts composing financial reports with predefined modules. In contrast, many other forms of writing (e.g., books, articles) are considerably more open-ended and creative. Nevertheless, when viewed through the lens of writing as a skill rather than writing as a profession, many occupational writing activities (e.g., by engineers or analysts) tend to follow structured patterns. As prior studies in domain-specific writing, such as legal (Choi and Schwarcz, 2024) and academic (Lee et al., 2022; Zhao et al.,

2025) contexts, report varying findings, our findings should be interpreted with caution and contextual awareness.

• **Design** Agent designers display patterns similar to those of agent administrative staff — producing work of moderate quality (Figure 20) but with notable efficiency advantages. They may thus be useful in contexts where quality requirements are not stringent, or as effective starting points that human designers can later refine.

Figure 20: Agent (top) and human (bottom) designs.

D.2. Task Delegation by Programmability

Given that agents are better at solving tasks via programmatic approaches, while humans excel at operating through UI interfaces, an important question arises: what types of work are best suited to each? Some tasks, or specific workflow steps within a task, can be readily programmed, whereas others cannot. We hence propose three levels of task programmability and discuss their respective implications for agent and human workers.

• **Readily Programmable**: These tasks can often be solved reliably through deterministic program execution. For example, writing a Python script to clean an Excel datasheet or coding a website in HTML. Compared with the UI-oriented tools preferred by many human workers, programmatic approaches often offer a stronger guarantee of outcome accuracy and greater scalability of the process. For instance, processing 10000 data entries in Python is typically faster and less error-prone than manually editing a 10k-row Excel sheet. While some human workers have begun integrating programming into their workflows, many still rely on UI tools and consequently operate less efficiently. Given current advances in agent programming capabilities, such tasks are likely the most suitable for agent automation at present.

• **Half Programmable**: Some tasks are theoretically programmable, but lack clear, direct programmatic paths within the same tool used by human workers. For instance, to generate a `docx` report, agents usually write the content in `Markdown` format and then convert it to `docx` files using auxiliary programming libraries. Similarly, instead of operating through `Figma`'s interface, agents may write `HTML` code to produce a comparable website design, albeit through a different underlying logic.

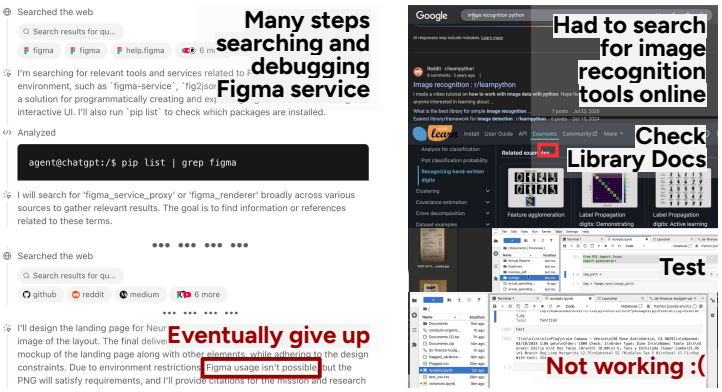

Figure 21: Left: Agent attempts to use Figma tool for design but fails. Right: Human attempts to use program tool for bill processing but fails.

For this type of task, it is not entirely clear whether progress should focus on advancing the agent's programmatic approach or on better emulating the human workflow. For humans, who typically rely on visually oriented tools for such tasks, it is often less intuitive to conceptualize how these operations could be performed programmatically. Agents, on the other hand, struggle with UI interaction: in the ChatGPT design process depicted in Figure 21 (left), when attempting to use the Figma design interface, it spends numerous steps figuring out the tool before eventually reverting to

writing programs — showing its limitations in interacting with professional UI environments. Beyond advancing agents' basic UI skills, further engineering efforts could focus on exposing or extending API-based programming pathways, such as supporting `figma` API accesses or developing alternative programming paths with equivalent functionalities.

• **Less Programmable**: Some tasks rely heavily on visual perception and lack a deterministic programmatic solution. For instance, in the task of viewing bill images and extracting data from them (Figure 21, right), a programmatic solution may require non-deterministic modules, such as invoking a neural OCR model to parse out the text, which may not guarantee task success (Wang et al., 2024a). Even human engineers, who can easily script some tasks such as data analysis, encounter challenges in this case. As in Figure 21 (right), the human worker's programmatic attempt fails after searching for relevant OCR packages, observing undesired parsing performance, and ultimately abandoning the task. For such cases, it is often far easier for humans to simply inspect the images and manually enter the numbers in an Excel sheet via the UI. Although these tasks may be long-tail, particularly after more tasks become programmatically solvable, they remain inevitable components of computer-use activities. Many of our analysis provides substantial evidence of agents' limitations in these less-programmable scenarios, underscoring the need for continued improvement through foundation model training.