Task Completion Agents are Not Ideal Collaborators

Anonymous Author(s)
Affiliation
Address
email

Abstract

Large Language Model (LLM) agents are increasingly capable of handling complex tasks autonomously, but current development and evaluation practices remain centered around one-shot task completion. This dominant paradigm fails to account for the inherently iterative and collaborative nature of many real-world problems, where human goals are often underspecified and evolve over time. This position paper argues for a shift in focus: from building and assessing task completion agents to developing *collaborative agents* — those evaluated not just by the quality of their final outputs, but by how well they engage with and enhance human effort throughout the problem-solving process. To support this shift, we introduce **collaborative effort scaling**, a framework that captures how an agent's utility grows with increasing user involvement. Through case studies and simulated evaluations, we show that state-of-the-art agents often underperform in multi-turn, real-world scenarios, revealing a missing ingredient in agent design: the ability to sustain engagement and scaffold user understanding. Collaborative effort scaling offers a new lens for diagnosing agent behavior and guiding development toward deeper, more adaptive interaction.

1 Introduction

Large Language Model (LLM) agents capable of handling complex tasks are becoming increasingly attractive [1–4]. Given a task description, we want agents that can *automatically* engage in long-form reasoning [5–7], interact with environments [8, 9], and use tools effectively [10–12] — with minimal human guidance. As a result, agent development has largely focused on producing high-quality, final outputs in one shot — what we call *task completion agents*. These agents are evaluated primarily through outcome-based metrics: did the result satisfy the user's prompt? This framing has also been proven operationally convenient and has driven much of the progress in LLM capabilities [13].

However, this dominant paradigm obscures a fundamental limitation: real-world tasks are rarely completed in one shot. Many are inherently iterative and collaborative — *requiring the agent not just to solve a problem, but to work with a human in navigating it* [14–16]. For example, in complex knowledge work like data analysis, users may not know exactly what insights they want to dig deeply into, until they have seen partial results and uncovered previously unknown constraints. In such cases where human goals are inevitably underspecified, When goals are underspecified, agents that assume static targets risk producing technically "complete" but practically useless outputs.

In fact, as we show through diverse case studies across domains like education, data analysis, and travel planning (Section 2), such agents frequently underperform in multi-turn settings: They prematurely generate overly polished answers that are hard to digest [17], fail to incorporate user feedback [18–20], and offer little transparency into their reasoning [21–24].

What's missing is a view of agent utility that reflects the process of collaboration, not just its endpoint. We argue that **desirable collaborator agents should be evaluated on their ability to encourage**

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

and leverage human effort — to inspire users to continuously refine their task specification (e.g., provide users with initial exploratory data analysis instead of full reports), and to draw on and amplify user input in ways that improve joint outcomes over time. This view shifts evaluation away from static outcomes and towards dynamic interaction trajectories between two usually overlooked, human-centered dimensions of collaborative agents (Fig. 1): *user effort* — how much cognitive and investigative work users invest in actively building an understanding of the task or the agent's reasoning process, rather than merely responding to the agent's clarification prompts; and *agent utility* — how much the agent contributes to the human, not only through improved task outcomes, but also by offering additional knowledge and scaffolding.

To better capture such iterative back-and-forth required for complex tasks, we take inspiration from the scaling laws in machine learning [7, 25], and introduce the concept of **collaborative effort scaling**: A framework that captures how well an agent's utility impacts, and scales with increasing user involvement. Our framework emphasize on two desired properties of collaborative agents: (1) *Continuous usability*: Agents should generate greater value with more user effort, and (2) *Maximum feasibility*: Agents should encourage and sustain engagement across longer interaction trajectories, especially in tasks where deeper understanding or high-stakes decisions are involved.

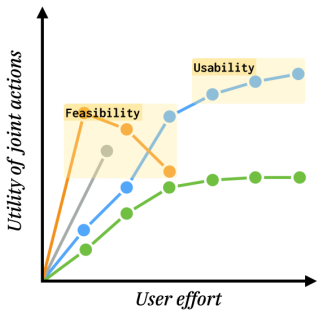

Figure 1: Collaboration Scaling.

As a first attempt, we apply this framework to study existing human agent collaboration setups in a simulated environment by Shao et al. [26]. In Section 4, we show that current agents are merely mediocre collaborators in complex, real-world knowledge tasks like travel planning [27] in that the additional user effort frequently leads to minimal or no improvement compared to a fully autonomous baseline. In-depth analysis of the collaboration reveals key limitations in agents' collaborative capabilities. A key issue is their reliance on a seemingly recursive problem-solving approach: they focus on completing immediate, individual tasks or user asks, but fails to come up and operate with an overarching optimal global plan.

In summary, this position paper advocates for developing collaborative agents and evaluating them with collaborative effort scaling. The current approach of optimizing for task completion does not yield important collaborative capabilities needed in the iterative process for accomplishing long-form tasks; and evaluating via collaborative effort scaling can offer helpful diagnostic insights and supports agent development in more challenge and complex real-world tasks.

2 Task completion agents in collaboration: Cases and Reflections

Most recent agents share a common, task completion objective: given a task description, the agent is expected to take actions to produce an output that satisfies the user's need. The agent can either be a standalone LM [28] or one equipped with tools and can autonomously perceive and take actions in an environment [1, 3]. Thanks to their ease of use, these task-to-output pipelines have become the dominant mode of interaction. To name a few: Manus [29] and OpenAI Operator [30] automate user tasks through web browsing; Cursor [31] and OpenHands [32] generate and edit code on demand.

We investigate how far this task completion paradigm extends to complex, real-world scenarios. Specifically, we examine knowledge-based tasks [33, 18] — those that demand significant human involvement for informed decision-making, learning, or creative work. Examples include data analysis, literature review and synthesis, or trip planning. A defining feature of these tasks is their *iterative nature* [14, 15]: reaching a satisfactory outcome often requires multiple rounds of refinement. To explore how agents perform in such settings, we collaborate with experts across five domains, analyzing concrete use cases.

2.1 Case Studies

Data analysis. Consider a case where a data scientist works with an agent in Google Colab [34] to analyze a coffee survey dataset [35]; their goal is to understand the data and come up with informed decisions for their business. After receiving the user's instructions and multiple steps of automated planning and action, the agent presents the user with a full-fledged report, However, this

Figure 2: We study five case studies of take completion agents in real-world iterative processes and distill key takeaways around collaboration success and challenges.

report includes hundreds of lines of code, visualizations, and a summary of the analysis, which is challenging to digest; It also glosses over early-stage exploratory analysis. As a result, it contains incorrect assumptions that go unnoticed. The data scientist struggles to pose meaningful follow-up questions and ultimately overlooks critical insights—such as patterns in regional coffee preferences or anomalies in pricing—due to limited transparency into how the conclusions were derived. In this case, while the agent technically fulfilled the user's request, the outcome is suboptimal. The root issue lies in the user's initial inability to fully articulate their analytical goals—often a consequence of not yet having a clear understanding of the data. Ideal agent should respect that developing a deep understanding of the data is naturally an iterative process. Rather than delivering a one-off report, the agent should focus on guiding the user through well-scoped, incremental analyses. By doing so, it can support the user in forming sharper questions and arriving at deeper, more actionable insights.

Reflection on: Data Analysis

What the Agent Currently Does	What the Agent Should Ideally Do
• Generates full reports with complex code and visuals.	• Support iterative exploration of data.
• Presents conclusions without process transparency.	• Expose assumptions and reasoning steps gradually.
• Assumes static goals from users.	• Facilitate goal refinement as insights evolve.

Travel planning. Consider the typical use case of travel planning – An American tourist uses a web agent such as OpenAI Operator [30] to plan a 7-day trip to Rome. The agent quickly provides a detailed itinerary, but fails to explain why certain attractions are included while others are omitted, or why specific durations are allocated. This triggers a series of follow-up questions from the tourist, which the agent struggles to answer. Worse, as the conversation unfolds, the agent begins to misread the tourist's intent and incorporates misleading or low-quality content from unreliable sources. Eventually, the tourist gives up and resorts to manual research, missing out on what could have been a more personalized and efficient experience. In this scenario, the novice tourist lacks the domain knowledge to interpret the itinerary on their own. This gap in understanding triggers unnecessary questions—questions that could have been easily avoided had the agent explained its reasoning. And because the user is already uncertain, any error or ambiguity from the agent becomes a breaking point, leading them to abandon the interaction entirely.

Reflection on: Travel Planning

What the Agent Currently Does	What the Agent Should Ideally Do
• Produces static itinerary from initial input.	• Support iterative sensemaking of travel options.
• Overloads user with opaque or generic suggestions.	• Explain rationale behind recommendations.
• Misinterprets user intent during follow-up.	• Respond constructively to evolving feedback.
• Breaks user trust with low-quality content.	• Maintain reliability across the interaction.

Financial advising. Consider a 35-year-old client who recently purchased their first home and welcomed their first child seeking personalized financial guidance from an LLM agent [36, 37]. After they provide basic information about their income and goals, the agent delivers comprehensive recommendations including investment allocations and insurance coverage. However, after discussing with colleagues, the client realizes their original self-assessments of goals and risk tolerance were

flawed and not well-calibrated for their social context and location. When the client tries to correct these assumptions and clarify their conservative investment preferences, the agent continues to suggest mismatched, aggressive strategies, leading to a loss of trust and the need for manual corrections. In this case, the agent's plan is again suboptimal because it prematurely locked in the user's initial preferences—despite the user's limited familiarity with the financial decision space—and failed to adapt as those preferences evolved. Ideally, the agent should support the user's sensemaking of the domain and, at a minimum, accommodate updated assumptions to reduce the mismatch between advice and context.

Reflection on: Financial Advising

What the Agent Currently Does	What the Agent Should Ideally Do
• Relies on a single-shot user self-assessment.	• Support users on reflective decision-making.
• Treats initial preferences as fixed throughout the session.	• Allow for dynamic re-evaluation of financial goals.
• Fails to adapt suggestions to new user insights.	• Revisit assumptions as user awareness evolves.

Education. Consider a high school student struggling with mathematical concepts they've encountered in class, unsure how to proceed with a homework assignment, and turning to a large language model (LLM) for assistance. The agent provides step-by-step answers, helping the student complete the task efficiently. However, it does not engage with what the student does or does not understand, nor does it adapt its explanations. As a result, the student completes the homework without building true comprehension, leading to poor performance in subsequent assessments [38]. In such a learning-oriented setting, the goal is not merely to fulfill the student's immediate request; it is to explain concepts in a way that equips the student to complete the assignment and internalize generalizable principles that support transfer learning. Achieving this requires more than correct answers—it requires adaptation and interaction. Furthermore, such effective learning also requires personalization — the LLM needs to engage with the student in an iterative process to understand what the student does not understand and what sorts of explanations click for the student.

Reflection on: Education

What the Agent Currently Does	What the Agent Should Ideally Do
• Prioritizes task completion over deep understanding.	• Adapt explanations to the student's level and gaps.
• Offers direct answers without probing comprehension.	• Encourage active learning through targeted questions.
• Lacks responsiveness to student learning signals.	• Balance short-term help with long-term learning goals.

Math discovery. Finally, another promising trend of the agents is to work with researchers and push frontiers in scientific discovery. A math professor shared an example of how they've used various language models (or agents) to support the proof of a novel theorem. Through multiple interactions, the agent generates many proof attempts, most of which contain subtle errors. While one conjecture generated by the agent sparks useful insight, the professor later reflects that it would have been faster to work without the agent, due to the time spent verifying flawed suggestions and lack of rigorous reasoning support.

Reflection on: Math Discovery

What the Agent Currently Does	What the Agent Should Ideally Do
• Suggests proofs with subtle but critical flaws.	• Collaborate through structured, step-wise reasoning.
• Lacks self-verification or explanation of logic.	• Flag uncertainty and validate intermediate steps.
• Increases user workload via repeated error-checking.	• Augment—not hinder—the user's scientific process.

2.2 Desiderata for Interactive Agents: User Effort, Agent Utility, and Their Interactions

Across all the case studies, a common pattern emerges: agents technically fulfill user requests—generating plausible data summaries, travel itineraries, financial plans, and so on. From a narrow *task completion* standpoint, they appear to be doing a reasonable job; yet the resulting outputs are consistently suboptimal. This disconnect stems from a fundamental misalignment: *agents assume that the user's initial task description fully captures their underlying needs*. However, in practice, this is rarely the case: Most real-world task specifications are inherently underspecified—for two key reasons: First, *tasks evolve*. As users gain more information, they often revise their goals or discover constraints that shift their priorities. In the financial advising example, the client expresses very

different preferences after gaining a better understanding of the domain. Similarly, the data scientist might have asked entirely different questions had they engaged earlier in exploratory analysis. Second, *the initial request often reflects a narrow surface-level goal that fails to capture the user's deeper objective*. When a tourist asks for an itinerary, they don't just want a list of places — they want to develop a sense of what's worth seeing and why. When a student asks for homework help, their broader goal (or at least, it should be) is to understand the concepts well enough to succeed beyond the current assignment. These cases underscore two user-centered dimensions that task-completion-focused agents tend to ignore:

- **Agent utility**: We usually evaluate agent utility narrowly based on final output quality. But in tasks with evolving goals, intermediate results — especially ones that help users calibrate their understanding — can be far more valuable than a polished endpoint. Here, the utility should be more broadly defined, by e.g. the additional knowledge they offer to users. Likewise, when the immediate task is a subgoal of a broader objective, the agent utility should be defined to emphasize long-term gains (e.g., learning or strategic planning) over short-term task completion.
- **User effort**: Many agents aim to minimize user involvement, or treat users primarily as providers of clarification. But in open-ended knowledge work, user engagement is not a nuisance—it is part of the process. Across the case studies, users are expected to (1) build understanding (e.g., of the dataset, financial options, or travel destination), and (2) inspect and build on the agent's reasoning (e.g., in scientific or educational contexts). These efforts shouldn't be minimized; rather, they should be strategically supported—and, when appropriate, even amplified. [1]
- **Interaction between the two**: Crucially, agent utility and user effort are interdependent. On one hand, user engagement is only productive when the agent produces outputs that are *interpretable* and *actionable*. Users may easily disengage if they find it difficult to follow up (as in the data analysis case), or if they get trapped in unnecessary clarifications (as in the travel case) or unfruitful interactions (as in financial advising) On the other hand, agent utility can only increase when users are asking meaningful, well-scoped questions that the agent can meaningfully support and answer.

These observations lead us to a broader argument: agents tackling complex tasks must be fundamentally **collaborative**. That means: (1) rather than just delivering results, agents should actively involving users in a process of shared discovery; (2) Rather than optimizing for minimal input, agents should be designed to effectively leverage user effort as part of the solution process.

We therefore propose that agent effectiveness in such settings should be evaluated not solely based on final outcomes, but on *how* those outcomes are reached — whether the agent can effectively involve the users as they work together towards the final goal. To capture this, we take inspiration from the scaling laws in machine learning, and propose **collaborative effort scaling** — to examining the extent to which an agent's utility scales with the amount and quality of user effort, visualized as the trajectory in Figure 1. Specifically, we highlight two desired goals for a collaborative agent derived from the trajectory:

- **Continuous usability**: Agents should generate greater value with more user effort — either by providing immediate gains from user contributions, or by enabling better final outcomes.
- **Maximum feasibility**: Agents should encourage and sustain engagement across longer interaction trajectories, especially in tasks where deeper understanding or high-stakes decisions are involved. Drop-off due to poor responses, misunderstandings, or unproductive interactions should be treated as a critical failure.

We formalize such notions in the next section.

3 Operationalizing collaborative effort scaling evaluation

Building and evaluating collaborative agents that conceptually described above require operationalizing the notions of human effort and agent utility. Here, we formalize these notions highlight key metrics that could be derived to reflect the collaborative effort scaling.

[1] We do not disagree that certain tasks are suitable for full automation with minimum human supervision (thus they are naturally suitable for the task completion paradigm). Still, we believe that there *exist* such tasks that human procedural involvement can still provide value, c.f. Haupt and Brynjolfsson [39] and Brynjolfsson [40], thus necessitating the iterative process for human-agent collaboration.

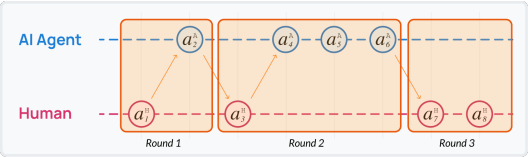

Figure 3: Collaboration rounds.

Formalization of human-agent collaboration. Following recent work [26], we describe the human agent collaboration process with a Partially Observable Markov Decision Process (POMDP) [41]. We study the **joint action trace** between the human and agent: $\mathbf{a} = [a_1^{(l_1)}, a_2^{(l_2)}, \ldots, a_T^{(l_T)}]$, where T is the total number of steps, and $l_t \in \{\texttt{H}, \texttt{A}\}$ indicates which party is taking action at step t. Each action is based on a corresponding context window $\mathbf{c} = [c_1^{(l_1)}, c_2^{(l_2)}, \ldots, c_T^{(l_T)}]$. The handoff between human and agent breaks down the whole collaboration process into *rounds*: $\mathbf{a}_k = \mathbf{a}_{[i_k:j_k]}$, where i_k and j_k are the start and end step of the action. As shown in Fig. 3, one round may start with a user action and followed with multiple agent actions, possibly including silent internal steps such as planning or retrieval—or an actual output update (e.g., generating a revised itinerary). Likewise, a user might act several times before handing control back. Likewise, the user may take several actions in a row.

The entire procedure can be further divided into two distinct stages. The first is the initial request stage, during which the agent produces a preliminary draft of the output. This stage concludes when $a_i^{\texttt{A}}$ generates the first substantial version at step i. Following this, the process transitions into a refinement stage, where the agents iteratively adjust and improve the output in response to human feedback. We consider these two stages in our subsequent metric definitions.

We note that, in this broad framing, both **human effort E** and **agent utility U** could be approximated in various different ways. For instance, a basic measure of human effort could be the number of human-led rounds, $|\mathbf{a}^{\texttt{H}}|$. This can be enriched by summing the contextual tokens the human processes $\sum \mathbf{c}^{\mathbf{A}}$, which captures not just frequency but also cognitive load – "Is this easy to read and respond to?" Additionally, effort may reflect action type: if users default to vague queries in response to specific model errors, this might signal that parsing or evaluating the context is prohibitively hard—so users defer the burden by moving the conversation forward.

Similarly, *agent utility* could be tied to per-round performance score P_k when utility is focused on the agent outcome. In more granular setups, utility could also consider additional aspects that move the collaborative team positively towards the final outcome, even if the output is not updated in certain rounds. For example, a positive move could also be the agent correctly resolves user clarifications or provides more information, even if the final answer is unchanged.

Mapping trajectory to metrics. With the human effort and agent utility forming the trajectory in Figure 1, we can further capture the key metrics related to usability and feasibility:

- **Overall utiltiy.** Here, we assess: Given unlimited human effort, what's the maximum value an agent can provide? We define a utility function across the entire interaction period, as

$$\mathbf{U} = \frac{1}{N}\sum_{i=1}^{N} \max U_k^{(i)},$$

  where N is the total number of instances in the evaluation (e.g. number of travel planning requests), and $\max U_k^{(i)}$ represents the maximum utility value (approximated in certain ways) for one given instance i.
- **Refinement gain.** Furthermore, building on the intuition that most of the interaction value comes from the refinement stage (i.e., most people will interact with the model at least until they get the first draft), we further define a metric more focused on the additional gain from the refinement. We define $\mathbf{G}$ as the performance improvement after the first major update:

$$\mathbf{G} = \frac{1}{N}\sum\nolimits_{i=1}^{N} \max U_k^{(i)} - U_{k_i'}^{(i)},$$

  where k_i' is the first round where the agent updates the output for the i-th task.
- **Feasibility drop.** We formalize the observation that when an agent fails to make consistent progress in the collaboration, the user may stop interacting due to frustration and dissatisfaction, and measure the feasibility utility — performance reached according to certain no-progress tolerance, defined by a tolerance threshold τ. For the i-th task, the user will stop the collaboration at step $k_{i,\tau}$ if the agent fails to make satisfactory progress for at most τ rounds. The performance drop under τ is defined as

$$\mathbf{D}@\tau = \frac{1}{N}\sum\nolimits_{i=1}^{N} U_{k_{i,\tau}}^{(i)} - U_{K_i}^{(i)}.$$

Notice that, in this case, we contrast $P^{(i)}_{k_{i,\tau}}$ with $P^{(i)}_{K_i}$, the performance of the agent at the end of the collaboration process, as a counterfactual measurement of the performance the agent can achieve if the user continues to interact with the agent.

4 Applying collaborative effort scaling evaluation in simulated experiments

We showcase the benefit of our framework through a simulation study, following recent work that approximates human behaviors [42–44]. Specifically, we simulate users with LLMs interacting with agents, and adopt the simplest proxies for measurement: we use the round performance score P_k as a stand-in for *utility*, and the number of rounds as a proxy for *human effort*.[2] This setup deliberately oversimplifies our broader framework, but enables a first step in a controlled environment—letting us sidestep the challenges of handling diverse user inputs or selecting the most faithful proxies. As we show below, even this minimal instantiation is sufficient to highlight differences between agents powered by different LLMs and prompts.

4.1 Experimental details

Setup. We use the Collaborative-Gym [26] environment that allows for asynchronous human and agent actions, which mimics the realistic interaction process. In this study, we focus on the travel planning task [27]: Given an initially high-level description of the user's travel goal, e.g., "Help me plan a 5-day trip from Omaha to Michigan starting on 2022-03-19", the agent will work with the simulated user to draft a travel plan that includes the itinerary, accommodation, and transportation. Throughout an iterative collaboration process, the agent can elicit the user's latent preferences and constraints, and both parties can use tools for retrieving travel information and edit the final travel plan together. We use the identical 101 subset of the travel planning dataset in the co-gym paper.

Metric. The agent performance is measured by the quality of the generated travel plan. We adopt the script by Xie et al. [27] that uses an LM to determine whether the derived plan satisfies common sense (commonsense pass rate) or user constraints (constraint pass rate), and report the arithmetic average as the performance. The same evaluation is used for both the output or any intermediate rounds with a travel plan updated to obtain P_k.

Implementation. The co-gym environment comes with an automated agent implementation based on the RaAct framework [45], as well as two collaborative agent implementations: `one-` and `two-stage planning` agents. In the process, the collaborative agent can opt to send a message to send messages to the simulated user. The difference between the `one-` and `two-stage planning` agent is that the latter is incorporates an additional planning step to determine whether to collaborate given the current state of the task and the user. We test three LMs, i.e., GPT-4o (`gpt-4o-2024-08-06`), Claude-3.5-sonnet (`claude-3-5-sonnet-20241022`), and Llama-3.1-70B: the agent prompts remain the same when we test with different LMs.

Simulated user. The simulated user is also a prompted agent based on `gpt-4o` with additional access to the user's preferences and goals of the task. Besides taking actions and providing feedback, it also gives a satisfaction rating for the agent's action during one round: for a round of actions $\mathbf{a}_k$, it produces a 5-point likert score that assess the agents' determining if the agent actions are making process towards the end goal. The interaction stops when either party finds the task is done or the total interaction actions exceeds a maximum number of 30 rounds.

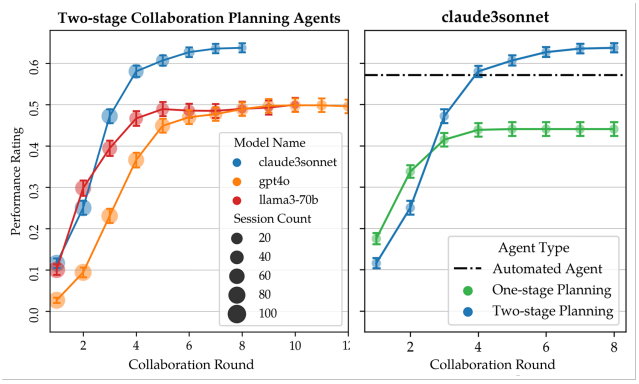

Figure 4: The collaborative scaling curve comparing different models and agent implementations.

[2] In some cases, agents may not update their output (e.g., only conducting searches or requesting more user information); in such cases, we prefill with the previous performance score P_{k-1}, with $P_0 = 0$.

Table 1: Different metrics for the one- and two-stage collaboration planning agents for the travel plan task.

Model Name	Automated Baseline	Utility			Refinement Gain		Feasibility Drop	
		First update	Final step	Overall	Abs.	Rel.	Abs.	Rel.
One-stage Collaboration Planning								
claude3sonnet	**0.572**	0.396	0.441	0.450	**0.054**	**13.6%**	-0.131	-29.7%
gpt4o	0.518	0.483	0.479	0.507	0.024	4.9%	-0.099	-20.8%
llama3-70b	0.482	**0.498**	**0.496**	**0.534**	0.036	7.1%	**-0.090**	**-18.0%**
Two-stage Collaboration Planning								
claude3sonnet	**0.572**	**0.647**	**0.637**	**0.687**	0.040	6.2%	-0.215	-33.7%
gpt4o	0.518	0.497	0.492	0.544	**0.047**	**9.5%**	-0.194	-39.3%
llama3-70b	0.482	0.514	0.498	0.539	0.025	4.9%	**-0.154**	**-30.9%**

4.2 Results and findings

Fig. 4 shows the performance change during the collaboration process for different models and agents. Overall, we find that **agent based on different LMs show a generally similar collaborative effort scaling trend**: there is a process of improvement in the beginning of collaboration, and the performance plateaus after around 5 rounds of interaction for all the agents.

Surprisingly, for gpt4o and Llama-70B, we find that collaborating with the user does not lead to better performance compared to the fully autonomous baseline. After inspecting the event log, we find that the collaborative version has a stronger tendency to get into loops of actions, resulting less effective collaboration and lower performance. Neither does the collaborative agent implementation leads very different performance.

Quite differently, the two-stage collaboration strategy leads to a significant performance boost for the claude model. Not only it achieve better performance than the one-stage planning version, but it also gets much better performance against the automated baseline. The metrics in Table 1 offers additional insights: despite the claude model has the best refinement gain in the one-stage planning case, the lower utility of the first update hinders the subsequent improvement. It shows that, while the two-stage collaboration planning agent may take a bit extra effort at the beginning (initially lower blue line in Fig. 4 right), it can lead to a better first product, which is crucial for a good final performance.

5 Discussion: Rethinking Agents through Utility and Effort

Our results suggest that current agent architectures are not just underperforming — they are misaligned with how real collaboration works. Here, we reflect on how to use *utility* and *effort* — the two fundamental dimensions of our framework – to rethink about agent design.

Utility and effort need thoughtful, human-centered proxy design One theme from our case study (which is also a limitation of our simulation) is that existing proxies for "success" — final task output, correctness, or superficial engagement — are not adequate. Both utility and effort are richer, more nuanced concepts. They evolve across time, vary by context, and often manifest in subtle human behaviors. For effort, we need to consider not just frequency of interaction, but cognitive load, sensemaking behavior, or even confusion. For utility, we must move beyond output quality to account for how the agent scaffolds understanding, encourages productive exploration, or clarifies ambiguity. User's interaction logs like edit histories, request timing, clarification patterns, etc. might help approximate such dimensions, similar to recent design of adaptive programming suggestions [46].

Mixed-initiative interaction should be structured by effort-utility trajectories It is not enough for agents to "respond well" to user prompts — they must know when to act, how to take initiative, and when to defer. This timing and control dynamic — often described as mixed-initiative interaction [47] — should be structured around the evolving effort-utility trajectory. An agent should step in when user progress stalls, prompt clarification when utility is plateauing, or relinquish control when users are gaining momentum. Critically, this requires not just good training data, but an interaction framework that models collaboration as a dynamic control problem. Rather than hardcoding turn-taking rules or

relying on rigid role assignment, agents should continuously infer where in the trajectory they are, and adjust accordingly.

Humans evolve during interaction — agents should be trained to encourage such shifts and adapt accordingly. One of the most underappreciated facts in human-agent collaboration is that users change. Their goals sharpen, assumptions shift, understanding deepens. Agents that assume static intent inevitably fail when these changes occur. Our case studies consistently show how quickly initial inputs become outdated. This means that agents should not be built with the goal of reducing effort or maximizing task efficiency, and should not only request inputs from humans when they want to ask clarification questions. Instead, future training of agents should be trained towards certain utility signals that can potentially directly contribute to human evolvement, even if it means the agent would engage in seemingly inefficient steps from a task outcome prospective — including backtracking, exploration, hypothesis generation, etc.

6 Related work

From Human-AI to Human-Agent Collaboration. Prior research has studied human "collaboration" and "teaming" with AI [48–50], proposing design guidelines for effective human-AI interactions [51, 52]. However, prior research focuses on AI outputs that operate within more constrained parameters: their capabilities are often limited to single tasks. In contrast, modern LLM agents that can access and execute tools to interact with external environments and have some form of memory can enable more dynamic and sophisticated interaction patterns [4, 1–3]. For example, a user can use Magentic-One [53] as a general assistant to complete web tasks or OpenHands [54] as a pair programmer for software development. In light of new agent development, we contribute to guidelines for effective human-*agent* interaction and call for the community to more carefully consider how to design agents for effective human collaboration.

Agent Benchmarks and Evaluation. Many benchmarks have been introduced to evaluate agent task completion capabilities across varying domains [55, 8, 56–58]. Each benchmark follows a similar structure: given a task description, agents are designed to create a plan, execute multiple tasks, and analyze novel situations to achieve the specified goal [59]. For example, SWE-Bench evaluates agents' abilities to fix bugs in code repositories [55], Web Arena measures agents' abilities to navigate the web and autonomously complete tasks [8], and GAIA tests agents' abilities to serve as general assistants to gather and synthesize information from the web and processes multimodal data [56]. While these benchmarks are useful for evaluating agent progress, they do not capture the practical interactive usage of agents by humans.

Recent work has proposed interactive evaluations to mimic real-world interaction scenarios [60, 61, 26]. For a given task, such as creative writing, interactive evaluations for human-AI collaboration focus on evaluating both (1) intermediate progress and (2) final outputs. However, these setups largely focus on scenarios where each step involving the AI is usually small and self-contained, making the decision of when to invoke the AI model and what result to return relatively straightforward. Translating model performance to helpful collaboration is already non-trivial in traditional human-AI collaboration, and the complex nature of AI agents only exacerbates this challenge. In particular, we focus our discussions on evaluating how user effort and agent utility scale through multiple interactions.

7 Conclusion

In this paper, we advocate for auditing and evaluating the human-agent collaboration process. Current agent benchmarks often treat the collaboration process as implicit or secondary to task completion, focusing primarily on outcome-based metrics. Through case studies with experts from five domains, we distill desiderata for effective collaboration that extend beyond mere task completion. Based on these insights, we introduced collaborative effort scaling as a framework that evaluates how effectively agents leverage and enhance human input throughout interaction trajectories. It helps us study the collaboration process and identify current agent limitations in a simulated experiment on travel planning. As agents are increasingly integrated into complex scenarios with inherently underspecified goals, measuring and optimizing for collaborative dynamics will be important, and we advocate a wide adoption of our framework in such settings.

References

[1] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*, 2023.

[2] Lilian Weng. Llm-powered autonomous agents. *lilianweng.github.io*, Jun 2023. URL `https://lilianweng.github.io/posts/2023-06-23-agent/`.

[3] Zane Durante, Bidipta Sarkar, Ran Gong, Rohan Taori, Yusuke Noda, Paul Tang, Ehsan Adeli, Shrinidhi Kowshika Lakshmikanth, Kevin Schulman, Arnold Milstein, et al. An interactive agent foundation model. *arXiv preprint arXiv:2402.05929*, 2024.

[4] T. Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L. Griffiths. Cognitive architectures for language agents. *ArXiv*, abs/2309.02427, 2023.

[5] OpenAI. Openai o3 and o4-mini system card. URL `https://api.semanticscholar.org/CorpusID:277857808`.

[6] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[7] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Fei-Fei Li, Hanna Hajishirzi, Luke S. Zettlemoyer, Percy Liang, Emmanuel J. Candes, and Tatsunori Hashimoto. s1: Simple test-time scaling. *ArXiv*, abs/2501.19393, 2025.

[8] Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.

[9] Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks, 2024. URL `https://arxiv.org/abs/2401.13649`.

[10] John Yang, Carlos E. Jimenez, Alexander Wettig, K. Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *ArXiv*, abs/2405.15793, 2024.

[11] Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *ArXiv*, abs/2503.09516, 2025.

[12] Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. *ArXiv*, abs/2501.05366, 2025.

[13] Shivalika Singh, Yiyang Nan, Alex Wang, Daniel D'Souza, Sayash Kapoor, Ahmet Üstün, Sanmi Koyejo, Yuntian Deng, Shayne Longpre, Noah A. Smith, Beyza Ermis, Marzieh Fadaee, and Sara Hooker. The leaderboard illusion, 2025. URL `https://arxiv.org/abs/2504.20879`.

[14] Greg Little, Lydia B Chilton, Max Goldman, and Robert C Miller. Exploring iterative and parallel human computation processes. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 68–76, 2010.

[15] D. Russell, M. Stefik, P. Pirolli, and S. Card. The cost structure of sensemaking. *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, 1993.

[16] Tongshuang Wu, Haiyi Zhu, Maya Albayrak, Alexis Axon, Amanda Bertsch, Wenxing Deng, Ziqi Ding, Bill Guo, Sireesh Gururaja, Tzu-Sheng Kuo, et al. Llms as workers in human-computational algorithms? replicating crowdsourcing pipelines with llms. *arXiv preprint arXiv:2307.10168*, 2023.

[17] Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. Llms get lost in multi-turn conversation. 2025.

[18] Megan Ma. Rethinking human-ai agent collaboration for the knowledge worker. https://law.stanford.edu/2025/04/01/rethinking-human-ai-agent-collaboration-for-the-knowledge-worker/. Accessed: 2025-05-18.

[19] Omar Shaikh, Hussein Mozannar, Gagan Bansal, Adam Fourney, and Eric Horvitz. Navigating rifts in human-llm grounding: Study and benchmark. *arXiv preprint arXiv:2503.13975*, 2025.

[20] Gagan Bansal, Jennifer Wortman Vaughan, Saleema Amershi, Eric Horvitz, Adam Fourney, Hussein Mozannar, Victor Dibia, and Daniel S Weld. Challenges in human-agent communication. *arXiv preprint arXiv:2412.10380*, 2024.

[21] Shuyue Stella Li, Jimin Mun, Faeze Brahman, Jonathan S Ilgen, Yulia Tsvetkov, and Maarten Sap. Aligning llms to ask good questions a case study in clinical reasoning. *arXiv preprint arXiv:2502.14860*, 2025.

[22] Sanidhya Vijayvargiya, Xuhui Zhou, Akhila Yerukola, Maarten Sap, and Graham Neubig. Interactive agents to overcome ambiguity in software engineering. *arXiv preprint arXiv:2502.13069*, 2025.

[23] Belinda Z Li, Been Kim, and Zi Wang. Questbench: Can llms ask the right question to acquire information in reasoning tasks? *arXiv preprint arXiv:2503.22674*, 2025.

[24] Belinda Z Li, Alex Tamkin, Noah Goodman, and Jacob Andreas. Eliciting human preferences with language models. *arXiv preprint arXiv:2310.11589*, 2023.

[25] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[26] Yijia Shao, Vinay Samuel, Yucheng Jiang, John Yang, and Diyi Yang. Collaborative gym: A framework for enabling and evaluating human-agent collaboration, 2024. URL https://arxiv.org/abs/2412.15701.

[27] Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. Travelplanner: A benchmark for real-world planning with language agents. In *arXiv.org*, 2024.

[28] Jacob Andreas. Language models as agent models. *ArXiv*, abs/2212.01681, 2022.

[29] The manus team. Manus, 2025. URL https://manus.im/. Accessed: 2025-05-19.

[30] OpenAI. Introducing operator, 2025. URL https://openai.com/index/introducing-operator/. Accessed: 2025-05-19.

[31] Cursor. Cursor - the ai code editor, 2025. URL https://www.cursor.com/features. Accessed: 2025-05-19.

[32] All Hands. All hands – switch your job. build what matters., 2025. URL https://www.all-hands.dev. Accessed: 2025-05-19.

[33] Fritz Machlup. *The production and distribution of knowledge in the United States*, volume 278. Princeton university press, 1962.

[34] Jane Fine, Mahi Kolla, and Ilai Soloducho. Data science agent in colab: The future of data analysis with gemini, March 3 2025. URL https://developers.googleblog.com/en/data-science-agent-in-colab-with-gemini/. Accessed: 2025-03-09.

[35] R for Data Science Community. The great american coffee taste test, 2024. URL https://github.com/rfordatascience/tidytuesday/tree/main/data/2024/2024-05-14. Accessed: 2025-03-11.

[36] Andrew W Lo and Jillian Ross. Can chatgpt plan your retirement?: Generative ai and financial advice. *Generative AI and Financial Advice (February 11, 2024)*, 2024.

[37] Christian Fieberg, Lars Hornuf, and David Streich. Using large language models for financial advice. *Available at SSRN 4850039*, 2024.

[38] Hamsa Bastani, Osbert Bastani, Alp Sungu, Haosen Ge, Özge Kabakcı, and Rei mariman. Generative ai can harm learning. 2024. URL `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4895486`.

[39] Andreas Haupt and Erik Brynjolfsson. Ai should not be an imitation game: Centaur evaluations. ICML 2025 Position Paper Track, forthcoming.

[40] Erik Brynjolfsson. The turing trap: The promise & peril of human-like artificial intelligence. *Daedalus*, 151:272–287, 2022. URL `https://api.semanticscholar.org/CorpusID:245877203`.

[41] L. Kaelbling, M. Littman, and A. Cassandra. Planning and acting in partially observable stochastic domains. *Artif. Intell.*, 101:99–134, 1998.

[42] Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36:30039–30069, 2023.

[43] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22, 2023.

[44] Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, et al. Sotopia: Interactive evaluation for social intelligence in language agents. *arXiv preprint arXiv:2310.11667*, 2023.

[45] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023. *URL https://arxiv.org/abs/2210.03629*, 2023.

[46] Valerie Chen, Alan Zhu, Sebastian Zhao, Hussein Mozannar, David Sontag, and Ameet Talwalkar. Need help? designing proactive ai assistants for programming. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2025.

[47] E. Horvitz. Principles of mixed-initiative user interfaces. pages 159–166, 1999.

[48] Dakuo Wang, Elizabeth Churchill, Pattie Maes, Xiangmin Fan, Ben Shneiderman, Yuanchun Shi, and Qianying Wang. From human-human collaboration to human-ai collaboration: Designing ai systems that can work together with people. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, page 1–6, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368193. doi: 10.1145/3334480.3381069. URL `https://doi.org/10.1145/3334480.3381069`.

[49] Katharine E Henry, Rachel Kornfield, Anirudh Sridharan, Robert C Linton, Catherine Groh, Tony Wang, Albert Wu, Bilge Mutlu, and Suchi Saria. Human–machine teaming is key to ai adoption: clinicians' experiences with a deployed machine learning system. *NPJ digital medicine*, 5(1):97, 2022.

[50] You Li, Yi Li, Qian Chen, and Yaping Chang. Humans as teammates: The signal of human–ai teaming enhances consumer acceptance of chatbots. *International Journal of Information Management*, 76:102771, 2024.

[51] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. Guidelines for human-ai interaction. In *Proceedings of the 2019 CHI*

Conference on Human Factors in Computing Systems, CHI '19, page 1–13, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450359702. doi: 10.1145/3290605.3300233. URL https://doi.org/10.1145/3290605.3300233.

[52] Babak Abedin, Christian Meske, Iris Junglas, Fethi Rabhi, and Hamid R Motahari-Nezhad. Designing and managing human-ai interactions. *Information Systems Frontiers*, 24(3):691–697, 2022.

[53] Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, et al. Magentic-one: A generalist multi-agent system for solving complex tasks. *arXiv preprint arXiv:2411.04468*, 2024.

[54] Xingyao Wang, Boxuan Li, Yufan Song, Frank F Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, et al. Openhands: An open platform for ai software developers as generalist agents. In *The Thirteenth International Conference on Learning Representations*, 2024.

[55] Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? *ArXiv*, abs/2310.06770, 2023.

[56] Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*, 2023.

[57] Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*, 2024.

[58] John Yang, Carlos E Jimenez, Alex L Zhang, Kilian Lieret, Joyce Yang, Xindi Wu, Ori Press, Niklas Muennighoff, Gabriel Synnaeve, Karthik R Narasimhan, Diyi Yang, Sida Wang, and Ofir Press. SWE-bench multimodal: Do AI systems generalize to visual software domains? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=riTiq3i21b.

[59] Margaret Mitchell, Avijit Ghosh, Alexandra Sasha Luccioni, and Giada Pistilli. Fully autonomous ai agents should not be developed, 2025. URL https://arxiv.org/abs/2502.02649.

[60] Mina Lee, Megha Srivastava, Amelia Hardy, John Thickstun, Esin Durmus, Ashwin Paranjape, Ines Gerard-Ursin, Xiang Lisa Li, Faisal Ladhak, Frieda Rong, Rose E Wang, Minae Kwon, Joon Sung Park, Hancheng Cao, Tony Lee, Rishi Bommasani, Michael S. Bernstein, and Percy Liang. Evaluating human-language model interaction. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=hjDYJUn911.

[61] Shuyue Stella Li, Vidhisha Balachandran, Shangbin Feng, Jonathan S. Ilgen, Emma Pierson, Pang Wei Koh, and Yulia Tsvetkov. Mediq: Question-asking LLMs and a benchmark for reliable interactive clinical reasoning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=W4pIBQ7bAI.