AGENTHABIT: CHARACTERIZING DISTINCT BEHAVIORS OF AGENTS ON EVERYDAY TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) agents assist users with everyday tasks that can be completed in many reasonable ways. Even when their answers are useful, how agents carry out these tasks may not match users' preferences and needs. For example, agents differ in whether they ask clarifying questions or search the web. We introduce HABIT, a taxonomy of 23 behavioral axes in five categories, which three authors and three LLMs derive bottom-up from 408 agent trajectories across 17 domains. On held-out tasks, HABIT distinguishes models more clearly than existing taxonomies of human values and agent actions while supporting comparably consistent annotation. Building on HABIT, we construct AGENTHABIT, a benchmark that profiles each agent's behavioral tendencies from its trajectories on 86 everyday tasks. Profiling 18 models with AGENTHABIT reveals a range of distinctive tendencies. For example, most GPT and Claude models state their assumptions and offer alternatives when requirements conflict, whereas Qwen and Google's models more often leave assumptions or changes to requirements unstated. These profiles remain recognizable even when built from entirely different sets of tasks, indicating that they reflect general tendencies rather than task-specific behavior. Prompting agents to adopt specific behaviors shifts some axes readily but barely changes others, while fine-tuning on another model's trajectories changes only part of a model's profile and leaves much of it intact. Overall, HABIT and AGENTHABIT provide a systematic framework for characterizing how agents carry out everyday tasks beyond task success, offering insights to guide the development of agents whose behavior better fits users' needs.

1 INTRODUCTION

AI agents assist users with everyday tasks such as drafting emails, learning new concepts, and seeking advice (Chatterji et al., 2025; Handa et al., 2025). Agents can successfully complete these tasks in a variety of ways (Zhao et al., 2024b; Jiang et al., 2025). For example, when asked how to automatically limit a child's tablet use, Claude asks about the device, while GPT gives instructions for several devices without asking (Figure 1). Parents seeking personalized advice may welcome these questions, while those wanting a quick answer may find them an unnecessary delay (Amershi et al., 2019). Even when the final answer is useful, users may already have spent time and money on unnecessary search and analysis. Without understanding these tendencies, users risk relying on agents whose behavior conflicts with their preferences and task requirements. They may also struggle to anticipate how agents will proceed and what form of output to expect. Profiling these tendencies can make agent behavior more transparent, helping users assess where agent behavior falls short of user needs or goes beyond task requirements and choose models accordingly.

Comparing these behavioral tendencies across models requires a taxonomy that captures differences in agents' choices when multiple reasonable approaches are available. We introduce HABIT, a taxonomy grounded in 408 trajectories from eight models performing open-ended tasks across 17 domains derived from the Anthropic Economic Index (Handa et al., 2025). The taxonomy includes 23 behavioral axes, such as *Ask-user timing*, *Self-correction framing*, and *Output option resolution* (Figure 1). For example, *Ask-user timing* distinguishes whether an agent asks the user questions only before giving its initial answer (*Before only*), only afterward (*After only*), both before and afterward (*Both phases*), or never (*Not asked*). Each axis has a definition, categorical labels, and a *decide-by* rule specifying which evidence to use when assigning a label. To test whether HABIT

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

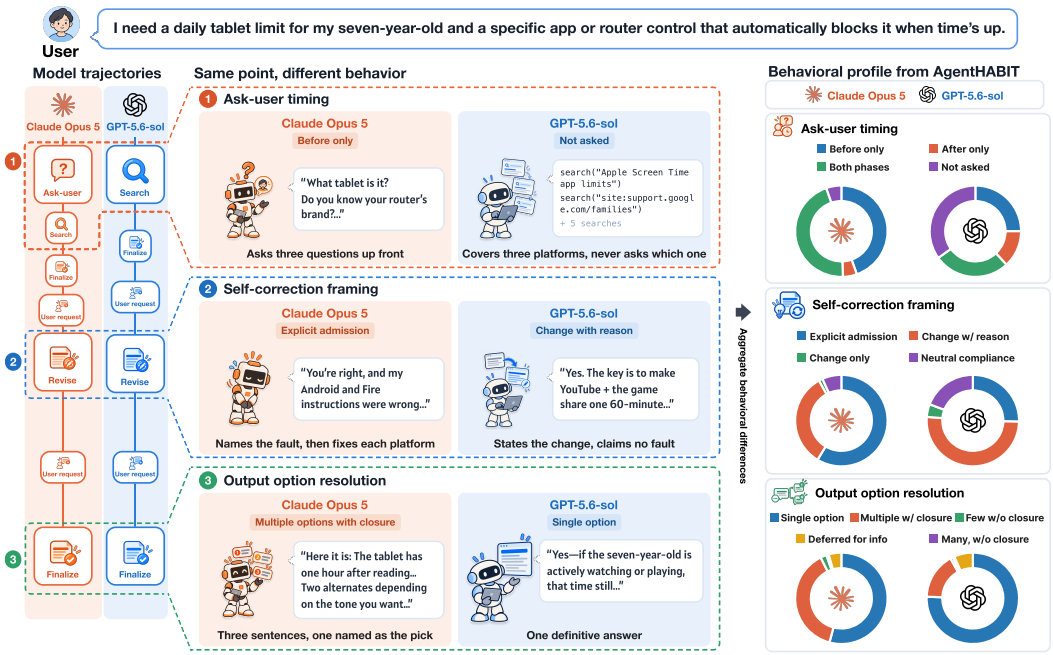


Figure 1: **From everyday interactions to behavioral profiles.** How two agents respond to the same opening request (left) and their behavioral profiles (right).

generalizes beyond the tasks used to construct it, we apply it to 408 trajectories from 17 held-out tasks. We compare it with Schwartz's human values used in psychometric profiling (Han et al., 2025; Dong et al., 2026) and agent action categories (Gao et al., 2026). HABIT more clearly distinguishes models' behavioral tendencies and yields more varied label distributions, while supporting consistent annotation (Section 2.3).

Building on HABIT, we introduce AGENTHABIT, a benchmark profiling agent behavior on everyday tasks. We aggregate behavior observed across tasks into a profile of each model's characteristic tendencies on the 23 behavioral axes (Figure 1). The resulting profiles show that most GPT and Claude models state their assumptions and offer alternatives when requirements conflict, while Qwen and Google's models more often leave assumptions or changes to requirements unstated (Figure 4). Such omissions can make it difficult for users to judge whether the advice fits their circumstances. Since these tendencies could be specific to the tasks evaluated, we compare profiles constructed from non-overlapping task subsets. The profiles continue to distinguish models and model families. This stability supports using AGENTHABIT to compare models' behavioral tendencies across the 23 axes.

We examine whether these profiles can be obtained at lower cost by asking models to report how they typically behave. Self-reports match models' most common observed behaviors on roughly half of the axes, limiting their use as a substitute for trajectory-based profiling. We also examine whether agents can be prompted to adopt specific behaviors or trained to shift toward another model's behavioral profile. We found that some behavioral axes are much easier to steer through prompting than others. Moreover, the same axes tended to be easy or difficult to steer across models, suggesting that steering results from one model may carry over to related models. In contrast, fine-tuning a model on another model's trajectories with SFT or DPO produced little shift toward the target model, leaving much of the original behavioral profile intact.

In summary, our work provides a framework for describing and comparing how agents carry out everyday tasks. By combining a taxonomy of observable behaviors with profiles derived from task trajectories, HABIT and AGENTHABIT capture differences in how models interact with users, gather information, and deliver results. These resources complement task-performance evaluations by helping users choose agents that fit their needs and helping agent builders adapt agent behavior to users' preferences.

2 THE HABIT TAXONOMY

To characterize how agents carry out everyday tasks, we construct HABIT through bottom-up human and LLM analysis of 408 task trajectories. The taxonomy organizes observable differences in how agents work into behavioral axes, each with a rubric for annotating trajectories. We first introduce the taxonomy's categories and axes (Section 2.1), then describe the construction process from task design to axis proposal and merging (Section 2.2). Finally, we validate HABIT on held-out tasks, comparing it with existing approaches in terms of model discriminability, annotation consistency, and label diversity (Section 2.3).

2.1 TAXONOMY OVERVIEW

HABIT organizes 23 behavioral axes into five broad categories (Figure 2). **User Interaction & Collaboration** describes how agents interact with users, including when they seek clarification and how they acknowledge errors. **Information Gathering & Tool Use** covers when and how agents use search and computational tools. **Decision-Making & Justification** describes how agents present and prioritize options and support their conclusions. **Task Management & Validation** covers how agents present their plans and check their work. **Output Design & Delivery** describes what agents deliver and how they guide users after delivering an answer. Within these categories, each of the 23 axes is defined by its own rubric, which contains a behavioral definition, categorical labels that distinguish different forms of the behavior, and a decide-by rule. The decide-by rule specifies which evidence to use when assigning a label, along with any timing or precedence rules. Appendix I provides the full inventory.



Figure 2: Overview of HABIT.

2.2 CONSTRUCTING HABIT

We construct HABIT in five stages: domain derivation, task construction, trajectory collection, axis proposal, and axis merging (Figure 3).

Domain derivation. We derive 17 domains for open-ended tasks from the aggregate request categories of the Anthropic Economic Index (AEI) (Handa et al., 2025) (Figure 3, Stage 1). These categories summarize large-scale real-world Claude usage on claude.ai and the first-party API. We retain categories covering tasks that agents can approach in a variety of reasonable ways. Within each platform, we group related categories into broader domains. For example, the *Home Vehicle Devices* domain brings together categories covering home heating, household maintenance, and related requests. The full derivation procedure is described in Appendix A.

Task construction. For each of the 17 domains, the authors develop one construction task with AI assistance (Figure 3, Stage 2). Each task consists of the user's opening request and private user information that a user simulator draws on when responding to the agent. To build each task, we first specify the user's goal, relevant facts, and practical constraints. Based on this information, we use LLMs to draft requirements for the answer and possible follow-up requests. We then manually review and revise each task. For example, consider the screen-time task in Figure 1. If the agent asks which apps the child uses, the simulator can answer that the child mostly watches YouTube (Appendix B).

Trajectory collection. We run eight models on each of the 17 construction tasks three times, yielding 408 trajectories (Figure 3, Stage 3). We select these models from five model families to capture behavioral diversity (Appendix C). Each trajectory includes agent messages, tool calls and results, and user responses. Agents share a system prompt and have access to four tools: web search (via the Tavily Search API), Python executor, ask user, and finalize answers.

Several differences in how agents carry out everyday tasks, such as when they ask the user questions and how they revise after feedback, emerge only through interaction with a user. LLM-simulated

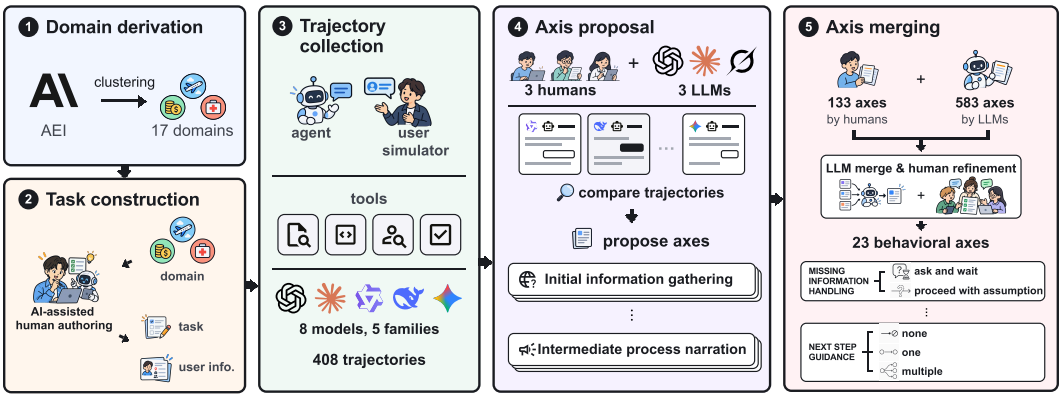


Figure 3: Five-stage construction of HABIT.

users are widely used to evaluate agents in such multi-turn settings (Yao et al., 2025). However, agents behave quite differently when users withhold information or grow impatient (Shim et al., 2026), so a simulator that is more cooperative than real users could misrepresent how agents behave in practice. We therefore design a user simulator based on GPT-5.6 Sol that mirrors how real users write and give feedback. We sample persona writing styles from distributions estimated from real user turns in WildChat-1M (Zhao et al., 2024b) and assign feedback dispositions according to their observed frequencies (Appendix E.1). We use this user simulator to generate user responses during trajectory collection. It follows its assigned persona and uses the task's user information and conversation history to answer the agent's questions (Figure 1). Once the agent provides an answer to the task, the user simulator can accept it, request revisions or additional work, or end the interaction without accepting it. We cap revisions at two per run so that trajectories remain short enough for reliable LLM judging. To assess the realism of the simulator, three human annotators independently rate 74 of its utterances. They judge the utterances as highly natural, contextually appropriate, and consistent with the assigned role (mean ratings of 4.55, 4.68, and 4.57 out of 5), with substantial agreement across annotators (Gwet's AC1 of 0.62–0.71). Appendix E.2 provides the evaluation protocol and detailed statistical results.

Axis proposal. Following qualitative content analysis (Hsieh & Shannon, 2005), three authors each spend more than 20 hours independently reviewing all 408 construction trajectories to identify how agents differ in carrying out tasks (Figure 3, Stage 4). Because subtle differences are easy to overlook across this many trajectories, three LLMs from different model families independently review the same trajectories to complement the human review. Each proposer works task by task, comparing the 24 trajectories collected for a task (eight models, three runs each) and drafting candidate axes that capture how they differ. This process yields 716 candidate axes, each with a definition, categorical labels, a decide-by rule, and an explicit `N/A` condition (Appendix D).

Axis merging. We merge and refine the candidate axes into 23 behavioral axes (Figure 3, Stage 5). Each axis should describe a behavioral difference observed across tasks, and its rubric should apply to unseen trajectories without changes. An LLM first groups candidate axes that describe overlapping behaviors and merges each group into a single axis, rewriting its rubric so that it is not specific to any one task. Three authors then jointly review the resulting axes against these criteria. They merge axes that still capture the same behavior, remove axes whose differences do not recur across tasks, and revise definitions, categorical labels, and decide-by rules so that each rubric can be applied to new trajectories as written (Appendices D and I).

2.3 VALIDATING HABIT

Prior work evaluates behavioral descriptions by whether they distinguish models and can be judged consistently (Dunlap et al., 2025). We examine whether HABIT meets these criteria on 408 new trajectories from 17 held-out tasks not used in taxonomy construction. As reference points, we compare HABIT with two existing approaches to describing model behavior: Schwartz's ten human values used in psychometric profiling (Han et al., 2025), which we apply through value ratings, and

Act·ONOMY's 46 agent subactions used for trace interpretation (Gao et al., 2026), which we apply through passage-level subaction annotations.

Table 1: Taxonomy comparison on held-out tasks. Bold indicates the highest value in each column.

Taxonomy	Matching (%)	Gap	Entropy	Intra-AC1	Inter-AC1
HABIT	**92.2**	**0.114**	**0.653**	**0.851**	0.718
Schwartz human values	72.8	0.047	0.491	0.650	0.383
Act·ONOMY	87.2	0.044	0.387	0.830	**0.755**

To find out whether HABIT can distinguish models by their behavior, we train a linear classifier to predict model identity from the behavioral labels assigned to each trajectory. We use leave-one-task-out cross-validation to evaluate matching on tasks not seen during classifier training. Across 28 model pairs, HABIT achieves 92.2% model-matching accuracy, compared with 72.8% for Schwartz values, 87.2% for Act·ONOMY, and 50% under random matching (Table 1). As a complementary measure, we calculate how much higher label agreement is between repeated runs of the same model than between different models on the same task. This difference (Gap) is 0.114 for HABIT, more than twice that of either reference taxonomy. These results suggest that HABIT captures behavioral tendencies that distinguish models on new tasks.

To assess whether the behavioral descriptions can be judged consistently, we compare annotations from three judge models, each annotating every trajectory three times. We measure agreement across repeated annotations and between judges using intra- and inter-judge Gwet's AC1. HABIT has an intra-judge AC1 of 0.851 and an inter-judge AC1 of 0.718, compared with 0.650 and 0.383 for Schwartz values and 0.830 and 0.755 for Act·ONOMY. The high intra-judge agreement supports consistent rubric application across repeated annotations. Inter-judge agreement is significantly higher than for Schwartz values and close to that of the binary subaction-presence summaries used for Act·ONOMY, even though HABIT distinguishes multiple forms of behavior on each axis.

Beyond these criteria, we examine how much variation the labels capture across trajectories. Even when judges agree, an axis provides little information about behavioral differences in the evaluated trajectories if nearly all receive the same label. We measure label diversity using normalized entropy, which increases as labels are used more evenly. HABIT has an entropy of 0.653, compared with 0.491 for Schwartz values and 0.387 for Act·ONOMY, indicating more varied and balanced label use. Together, these results show that HABIT captures a range of behavioral differences that can be consistently annotated and used to distinguish models on new tasks. Annotation procedures and statistical comparisons are provided in Appendix F.

3 THE AGENTHABIT BENCHMARK

To profile model behavior using HABIT, we construct AGENTHABIT, a benchmark of 86 open-ended tasks. We first describe how we collect trajectories and build behavioral profiles from them (Section 3.1). We then present the main results (Section 3.2) and examine profile stability across task subsets (Section 3.3).

3.1 BENCHMARK SETUP AND PROFILING

Task construction. We construct 86 new open-ended tasks, one per subdomain across the 17 domains derived from AEI (Appendix A). These tasks are distinct from the 34 tasks used to construct and evaluate HABIT. Following the procedure in Section 2.2, an LLM drafts each task for a given domain and subdomain. The draft specifies the user's goal, relevant facts, practical constraints, and requirements for the answer, along with possible follow-up requests. The authors review and revise every task for realism and overlap with existing tasks.

Models and trajectory collection. We collect 4,644 trajectories by having 18 models perform each of the 86 tasks three times. We evaluate GPT-OSS 120B and GPT-5.6 Sol, Terra, and Luna; Claude Opus 5, Sonnet 5, and Haiku 4.5; Qwen 3.7 Max, Plus, and Flash; DeepSeek V4 Flash-0731; Gemini 3.7 Flash; Gemma 4 31B, 26B-A4B, and 12B; Nemotron 3.5 Lightning 30B-A3B; Muse Spark 1.3; and Solar Pro 4. We follow the same trajectory collection procedure described in Section 2.2.



Figure 4: Behavioral profiles on five representative HABIT axes, renormalized after excluding N/A.

Behavioral profiling. We construct behavioral profiles by applying the 23 HABIT rubrics to task trajectories. An LLM judge reads each trajectory, without seeing which model produced it, and assigns a categorical label on one axis at a time, citing supporting evidence from the trajectory. For judge selection, three human annotators independently apply the same rubrics to 17 trajectories on all 23 axes. We compare annotations from four LLM judges with the resulting human-majority labels and select Qwen3.8-27B. It achieves 83.52% agreement on the 364 cases with a human-majority label (Cohen's $\kappa = 0.8282$), the highest observed agreement among the four judges. The judge comparison and selection procedure are detailed in Appendix G. Using the selected judge, we annotate each trajectory three times to reduce the effect of judge stochasticity. Aggregating these labels across tasks and runs yields each model's label distribution on each axis, and the 23 distributions together form its behavioral profile.

3.2 MAIN RESULTS

Figure 4 shows behavioral profiles on one axis from each of the five broad HABIT categories. Columns show behavioral axes, and rows group models by family. Each bar chart shows one model's label distribution on an axis, with colored segments indicating how often it exhibits each behavior.

Assumptions and conflicting requirements. In the leftmost column, Qwen and Google's models often fill in missing information without asking the user or stating their assumptions (red). Apart from their smaller variants, GPT and Claude models tend to state their assumptions (green). Leaving such assumptions unstated can cost users time and effort. For example, a user who relies on public transport might book accommodation based on a travel plan before discovering that it assumes access to a car. Models show a similar pattern when the user's requirements cannot all be satisfied (*constraint-conflict handling*, fourth column). GPT and Claude models generally explain the conflict and offer a feasible alternative (red), with exceptions among their smaller variants. Qwen and especially Google's models more often provide answers that violate or change the requirements without making this clear (green).

Tool use and user preferences. The second column shows how models differ in their use of code-execution tools. GPT and Gemma models often complete tasks without these tools (red), while Claude and Qwen models often use them both before and after the first substantive deliverable (green). A similar but less uniform pattern appears in web search: Qwen models and Claude Opus 5 search more often than GPT and Gemma models (Appendix Figure 13). However, which level of tool use is preferable depends on what the user wants from the task. For example, when planning a trip, one user may want a few quick suggestions without extensive searching, while another may want a detailed report grounded in external sources. Behavioral profiles can thus help users choose models whose tool-use tendencies match their needs.

Guidance on interpreting and using results. In the third column, GPT-5.6 models tend to acknowledge uncertainty or limitations when presenting conclusions (green), while Qwen models more often leave them unstated (blue). The rightmost column shows whether models offer guidance on what to do next after delivering an answer. Claude models tend to suggest multiple concrete next steps (green), while Qwen models more often provide none (red). Some users may value guidance on when an answer applies and what to do next, while others may prefer a concise answer with fewer follow-up suggestions. Appendix J provides complete profiles for all 18 models across all 23 behavioral axes.

Similarity within model families. Figure 5 shows pairwise similarities between model profiles. We define *profile similarity* as one minus the mean total variation (TV) distance between label distributions across the 23 behavioral axes. Higher values indicate more similar behavioral profiles. Mean within-family similarity is 0.93 for GPT-5.6, 0.91 for Qwen, and 0.91 for Gemma. All three exceed the overall mean of 0.80 across the model pairs in the figure (all $p < .001$, Appendix K). Claude has a lower mean similarity of 0.77, with a particularly large difference between Opus and Haiku. Thus, while models from the same family tend to behave similarly, this does not always hold, and family alone is not a reliable guide when choosing an agent.

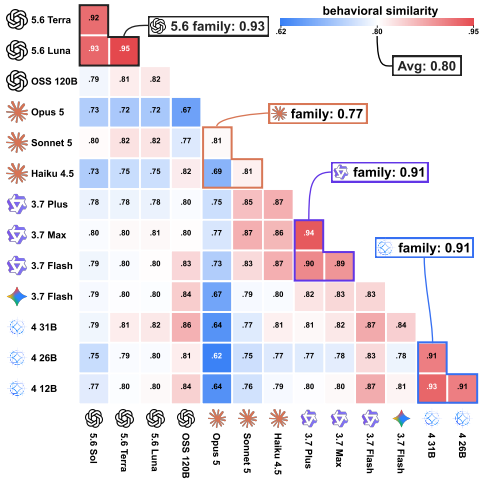


Figure 5: Pairwise similarity of behavioral profiles across models.

3.3 Profile Stability Across Task Subsets

To assess whether AgentHabit profiles are stable across tasks, we compared profiles constructed from disjoint task subsets. We sampled subsets from the 86 tasks in AgentHabit at 1%, from 10% to 90% in increments of 10 percentage points, and at 99%. For each model, we constructed separate profiles using the sampled tasks and the remaining, non-sampled tasks. Using each subset profile as a query, we identified the nearest profile constructed from the remaining tasks and measured whether the matched profile belonged to the same model or to a model from the same family. We repeated this process 1,000 times for each sampling ratio and report the average results. With only 10% of the tasks, 80.21% of the subset profiles were matched to the same model, while 98.90% were matched to a model from the same family (Table 15). When 50% of the tasks were used, these rates increased to 89.86% and 100.00%, respectively. These results indicate that AgentHabit captures task-general model profiles rather than profiles specific to particular tasks. Full results are provided in Appendix L.

4 Analysis

4.1 Can Self-Reports Recover Behavioral Profiles?

Table 2: Self-report–behavior correspondence. *Mode* denotes Mode Agreement (%).

	GPT-5.6			GPT-OSS	Claude			Qwen 3.7			Gemini 3.7	Gemma 4				
Metric	Sol	Terra	Luna	120B	Opus 5	Sonnet 5	Haiku 4.5	Max	Plus	Flash	Flash	31B	26B	12B	Mean	Random
Mode (%)	65.2	69.6	47.8	47.8	56.5	65.2	34.8	43.5	47.8	56.5	43.5	43.5	39.1	52.2	50.9	33.3

Our profiles are based on behavior observed during task execution, but obtaining them requires collecting and annotating full trajectories. Self-reports could offer a less costly alternative. Prior work has used self-reports to profile LLMs' personalities, values, and behavioral tendencies (Huang et al., 2024; Lim et al., 2026; Jiang et al., 2023). However, LLMs' self-reported values do not always align with their response tendencies in everyday scenarios (Song et al., 2026; Han et al., 2026). We therefore test whether models can accurately report their typical behavior in AgentHabit tasks. For each of the 23 axes, models answered a multiple-choice question (e.g., "How does the assistant respond

when the requested constraints cannot all be satisfied?") by selecting the option that best described their typical behavior. We repeated each question ten times for each of the 14 models, shuffling the options each time, and aggregated the responses into self-reported profiles for comparison with the corresponding AGENTHABIT profiles. Appendix M provides the full questionnaire and protocol.

Table 2 shows that self-reports only partly reflect observed behavior. *Mode Agreement* measures whether the behavior a model selects most often in the questionnaire is also its most common behavior. Mode Agreement averages 50.9% across axes and models, above the uniform random baseline of 33.3% but still leaving roughly half the axes with a mismatch. Thus, self-reports alone do not reliably identify models' most common behaviors.

4.2 DEPENDENCE ACROSS BEHAVIORAL AXES

If several axes capture the same behavioral pattern, they may carry redundant information in a model's profile. We therefore examine how much information the 23 axes of HABIT share across task trajectories. For each axis pair, we compute Cramér's V from how often their labels occur together. V measures the strength of association between the two categorical axes, with higher values indicating stronger association. We also examine how much knowing the label on one axis tells us about the label on another. For this, we use the uncertainty coefficient U, which measures the fraction of uncertainty about one axis's label reduced by knowing the other. We compute U in both directions because one axis may explain more of the other than vice versa.

Although some behavioral axes are associated, knowing the label on one axis generally provides limited information about the label on another. Under Cohen's effect-size conventions for Cramér's V, 53 of the 253 axis pairs (20.9%) show medium or large associations (Cohen, 1988). However, across all pairs and both directions, the mean reduction in uncertainty measured by U is 3.40% ($U = 0.0340$). For example, knowing how clearly an agent identifies sources for its claims reduces uncertainty about how it handles missing information by only 1.40% ($U = 0.0140$). These results suggest that most of the axes capture different aspects of agent behavior. Details are in Appendix N.

4.3 STEERING AGENT BEHAVIOR

The profiles in Section 3 show how models tend to behave when carrying out tasks. Can we also change these tendencies to better match our preferences? We examine two approaches: prompting agents to exhibit specific behaviors and fine-tuning them on another model's trajectories. We use Gemma 4 12B for both approaches because it is the only model in our evaluation that we can fine-tune with our available computing resources. We also apply prompt-based steering to Gemma 4 31B to examine whether similar effects occur in a larger model from the same family.

Prompt-based steering. To make changes easier to observe, we prompted each model to adopt its least frequent behavior on each of the 23 axes. We compared how often these behaviors occurred with and without prompting. As shown in Table 3, the frequency of the target behavior, averaged across the 23 axes, rose from 3.6% to 23.0% for 31B and from 3.3% to 21.5% for 12B. In both models, instructions to use decorative symbols extensively had large effects. Instructions to search only after the first substantive deliverable led to more moderate increases in the requested behavior, while instructions to cite sources throughout the answer had much smaller effects. More broadly, behaviors that changed more in one model also tended to change more in the other. Across the 22 axes with the same target behavior and instruction, the changes were strongly correlated (Pearson $r = 0.909$). Results from one model may therefore help guide attempts to steer another model from the same family. Details are in Appendix O.

Table 3: Prompt-steering target-label rates (%) for six representative shared-target axes. Δ denotes the absolute change.

	Gemma 4 31B			Gemma 4 12B		
Axis	Base	Steered	Δ	Base	Steered	Δ
Decorative Symbol Use	16.5	98.4	+81.9	3.2	96.5	+93.3
Next Step Guidance	0.3	45.0	+44.7	0.5	22.9	+22.4
Web Search Timing	4.3	37.2	+32.9	4.7	28.3	+23.6
Missing Information Handling	1.6	11.6	+10.0	1.6	1.6	0.0
Source Support Visibility	4.1	6.6	+2.5	3.2	8.5	+5.3
Process Narration	0.6	2.7	+2.1	7.6	17.4	+9.8
Macro-avg. (23 axes)	3.6	23.0	+19.4	3.3	21.5	+18.2

Training-based steering. We tested whether training on another model's trajectories could bring their behavioral profiles closer. We chose Claude Opus 5, whose profile was the most distant from Gemma 4 12B's in our evaluation (Figure 5). We trained separate models using SFT and DPO on AGENTHABIT tasks and evaluated them on the held-out HABIT tasks used in Section 2.3. SFT used Opus responses as training targets, while DPO favored Opus trajectories over Gemma trajectories from the same task. We measured these changes using the profile similarity defined in Section 3.2. Similarity to Opus changed only slightly, from approximately 0.604 before training to 0.601 after SFT and 0.607 after DPO. Neither change was statistically significant. The trained models also remained close to the original Gemma profile, with similarities of 0.908 for SFT and 0.932 for DPO. These results suggest that training on another model's trajectories alone may not be enough to produce the intended behavioral changes. Details are in Appendix P.

5 RELATED WORK

Evaluating agent outcomes and processes. Agent evaluations primarily measure task success in domains such as coding, computer use, and web navigation (Jimenez et al., 2024; Zhou et al., 2024; Xie et al., 2024; Liu et al., 2024). Other benchmarks provide finer-grained assessments by measuring intermediate progress (Ma et al., 2024) or task-completion efficiency (Xu et al., 2025). Together, these benchmarks provide systematic measures of agent performance. While these benchmarks reveal how well agents complete tasks, they offer limited insight into agents' behavior during task execution, such as how thoroughly they verify their work or how they handle ambiguous instructions. We instead characterize agents by the behavioral tendencies that recur across tasks.

Profiling large language models. Research on model profiling broadly follows two lines: psychometric profiling and purpose-specific behavioral analysis. Psychometric constructs, such as Schwartz's theory of basic human values (Schwartz, 1992), have been widely used to profile models by scoring their responses to questionnaire-style items (Ren et al., 2024; Han et al., 2025; Zhao et al., 2024a; Jiang et al., 2024; Lee et al., 2025). Human values have also been used to profile agents based on their task-execution trajectories (Dong et al., 2026). Rather than applying predefined inventories, bottom-up methods inductively derive evaluation criteria or value profiles from model outputs (Perez et al., 2023) and real-world interactions (Huang et al., 2025). These approaches characterize models in terms of values and traits. However, they do not establish links between value or trait scores and agents' actions during task execution. A high openness score does not indicate whether an agent will explore multiple alternatives, seek additional information, or commit early to a single approach.

Some studies analyze agent behavior for specific analytical purposes. These include hierarchical action taxonomies derived through grounded theory for trace interpretation and oversight (Gao et al., 2026), and categorizations of control decisions and failure modes for benchmark auditing and failure diagnosis (Mazaheri & Mazaheri, 2026). Cross-task behavioral consistency has also been measured using predefined features weighted by contributions to task success (Banerjee & Raichura, 2026). While these approaches support trace interpretation, diagnostic evaluation, and reliability assessment, HABIT provides broader, more interpretable profiles of agents' behavioral tendencies across tasks.

6 CONCLUSION

We introduced HABIT, a taxonomy of 23 behavioral axes, and AGENTHABIT, a benchmark designed to profile how LLM agents carry out everyday tasks. Through behavioral profiles of 18 models, we find that agents differ systematically in how they seek information, handle user requirements, and present results, with these profiles distinguishing individual models, recurring within several model families, and remaining stable across disjoint task subsets. Our steering experiments further show that these tendencies are not easily changed, as prompting shifts some behaviors far more than others and fine-tuning on another model's trajectories leaves much of the original profile intact. By grounding behavioral descriptions in what agents actually do during task execution, HABIT and AGENTHABIT provide a foundation for comparing agents beyond task success and for developing methods to shape their behavior. We hope these resources help users choose agents that suit their ways of working and guide the development of agents that better fit users' needs.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic datasets, implement methods, assist with translation, clean and reformat datasets, and support qualitative and thematic data analysis. We have not used generative AI tools to design or provide feedback on research methodology or experiments, or to interpret results. Developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, and proposing or refining hypotheses are not applicable to this work. Additionally, we used generative AI tools to create and edit software code and to edit the paper for improved readability, including grammar correction. We have reviewed all AI-assisted work. All AI-generated code was reviewed by the authors, and all AI-assisted translations were also reviewed by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We plan to publicly release our code and data, including task definitions, user personas, prompts, trajectories, annotations, and analysis scripts. For Tavily search results, we release only the source URLs and exclude the retrieved content, as it consists of third-party web content that may be subject to copyright. We also provide detailed experimental settings in the appendices of this paper. Taken together, these resources will support the verification of our findings and future research.

Several considerations may limit exact reproducibility. Results may vary due to changes to or retirement of API-based models, stochastic language model outputs, and changes in search results over time. To mitigate the effects of these factors, we collect three trajectories for each model–task pair and annotate each trajectory three times. While these repetitions help reduce statistical uncertainty, some degree of uncertainty may still remain.

REFERENCES

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, pp. 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.

Amritesh Banerjee and Pranil Raichura. Measuring cross-task behavioral consistency in language model agents, 2026. URL https://arxiv.org/abs/2608.13598.

Aaron Chatterji, Thomas Cunningham, David Deming, Zoe Hitzig, Christopher Ong, Carl Shan, and Kevin Wadman. How people use chatgpt. *SSRN Electronic Journal*, 2025. URL https://api.semanticscholar.org/CorpusID:281361103.

J. Cohen. *Statistical Power Analysis for the Behavioral Sciences*. L. Erlbaum Associates, 1988.

Shachar Don-Yehiya, Leshem Choshen, and Omri Abend. Naturally occurring feedback is common, extractable and useful, 2025. URL https://arxiv.org/abs/2407.10944.

Haonan Dong, Qiguan Feng, Kehan Jiang, Haoran Ye, Xin Zhang, and Guojie Song. Agent-valuebench: A comprehensive benchmark for evaluating agent values, 2026. URL https://arxiv.org/abs/2605.10365.

Lisa Dunlap, Krishna Mandal, Trevor Darrell, Jacob Steinhardt, and Joseph E. Gonzalez. Vibecheck: Discover and quantify qualitative differences in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=acxHV6werE.

Jie Gao, Kaiser Sun, Jen tse Huang, Katherine Van Koevering, Sijie Ji, Heyuan Huang, Weiyan Shi, Zhuoran Lu, Ziang Xiao, Daniel Khashabi, and Mark Dredze. How to interpret agent behavior, 2026. URL https://arxiv.org/abs/2605.13625.

Jongwook Han, Dongmin Choi, Woojung Song, Eun-Ju Lee, and Yohan Jo. Value portrait: Assessing language models' values through psychometrically and ecologically valid items. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17119–17159, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.838. URL https://aclanthology.org/2025.acl-long.838/.

Jongwook Han, Woojung Song, Jonggeun Lee, and Yohan Jo. Quantifying data contamination in psychometric evaluations of LLMs. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 6070–6088, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-386-9. doi: 10.18653/v1/2026.findings-eacl.319. URL https://aclanthology.org/2026.findings-eacl.319/.

Kunal Handa, Alex Tamkin, Miles McCain, Saffron Huang, Esin Durmus, Sarah Heck, Jared Mueller, Jerry Hong, Stuart Ritchie, Tim Belonax, Kevin K. Troy, Dario Amodei, Jared Kaplan, Jack Clark, and Deep Ganguli. Which economic tasks are performed with ai? evidence from millions of claude conversations, 2025. URL https://arxiv.org/abs/2503.04761.

Hsiu-Fang Hsieh and Sarah Shannon. Three approaches to qualitative content analysis. *Qualitative Health Research*, 15:1277–1288, 11 2005. doi: 10.1177/1049732305276687.

Jen-tse Huang, Wenxuan Wang, Eric John Li, Man Ho Lam, Shujie Ren, Youliang Yuan, Wenxiang Jiao, Zhaopeng Tu, and Michael Lyu. On the humanity of conversational AI: Evaluating the psychological portrayal of LLMs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=H3UayAQWoE.

Saffron Huang, Esin DURMUS, Kunal Handa, Miles McCain, Alex Tamkin, Michael Stern, Jerry Hong, and Deep Ganguli. Values in the wild: Discovering and mapping values in real-world language model interactions. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=zJHZJClG1Z.

Guangyuan Jiang, Manjie Xu, Song-Chun Zhu, Wenjuan Han, Chi Zhang, and Yixin Zhu. Evaluating and inducing personality in pre-trained language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Hang Jiang, Xiajie Zhang, Xubo Cao, Cynthia Breazeal, Deb Roy, and Jad Kabbara. PersonaLLM: Investigating the ability of large language models to express personality traits. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 3605–3627, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.229. URL https://aclanthology.org/2024.findings-naacl.229/.

Liwei Jiang, Yuanjun Chai, Margaret Li, Mickel Liu, Raymond Fok, Nouha Dziri, Yulia Tsvetkov, Maarten Sap, and Yejin Choi. Artificial hivemind: The open-ended homogeneity of language models (and beyond). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=saDOrrnNTz.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik 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.

J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.

Seungbeen Lee, Seungwon Lim, Seungju Han, Giyeong Oh, Hyungjoo Chae, Jiwan Chung, Minju Kim, Beong-woo Kwak, Yeonsoo Lee, Dongha Lee, Jinyoung Yeo, and Youngjae Yu. Do LLMs have distinct and consistent personality? TRAIT: Personality testset designed for LLMs

with psychometrics. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 8412–8452, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.469. URL https://aclanthology.org/2025.findings-naacl.469/.

Sungjib Lim, Woojung Song, Eun-Ju Lee, and Yohan Jo. Psychometric item validation using virtual respondents with trait-response mediators. *Transactions of the Association for Computational Linguistics*, 14:1369–1391, 07 2026. ISSN 2307-387X. doi: 10.1162/TACL.a.733. URL https://doi.org/10.1162/TACL.a.733.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating LLMs as agents. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=zAdUB0aCTQ.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. Agentboard: An analytical evaluation board of multi-turn LLM agents. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=4S8agvKjle.

Parsa Mazaheri and Kasra Mazaheri. Agentatlas: Beyond outcome leaderboards for llm agents, 2026. URL https://arxiv.org/abs/2605.20530.

Claude Nadeau and Yoshua Bengio. Inference for the generalization error. In S. Solla, T. Leen, and K. Müller (eds.), *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/file/7d12b66d3df6af8d429c1a357d8b9e1a-Paper.pdf.

Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, Andy Jones, Anna Chen, Benjamin Mann, Brian Israel, Bryan Seethor, Cameron McKinnon, Christopher Olah, Da Yan, Daniela Amodei, Dario Amodei, Dawn Drain, Dustin Li, Eli Tran-Johnson, Guro Khundadze, Jackson Kernion, James Landis, Jamie Kerr, Jared Mueller, Jeeyoon Hyun, Joshua Landau, Kamal Ndousse, Landon Goldberg, Liane Lovitt, Martin Lucas, Michael Sellitto, Miranda Zhang, Neerav Kingsland, Nelson Elhage, Nicholas Joseph, Noemi Mercado, Nova DasSarma, Oliver Rausch, Robin Larson, Sam McCandlish, Scott Johnston, Shauna Kravec, Sheer El Showk, Tamera Lanham, Timothy Telleen-Lawton, Tom Brown, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Jack Clark, Samuel R. Bowman, Amanda Askell, Roger Grosse, Danny Hernandez, Deep Ganguli, Evan Hubinger, Nicholas Schiefer, and Jared Kaplan. Discovering language model behaviors with model-written evaluations. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13387–13434, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.847. URL https://aclanthology.org/2023.findings-acl.847/.

Yuanyi Ren, Haoran Ye, Hanjun Fang, Xin Zhang, and Guojie Song. ValueBench: Towards comprehensively evaluating value orientations and understanding of large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2015–2040, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.111. URL https://aclanthology.org/2024.acl-long.111/.

Shalom H. Schwartz. Universals in the content and structure of values: Theoretical advances and empirical tests in 20 countries. In Mark P. Zanna (ed.), *Advances in Experimental Social Psychology*, volume 25, pp. 1–65. Academic Press, 1992. doi: 10.1016/S0065-2601(08)60281-6.

Jeonghoon Shim, Woojung Song, Cheyon Jin, Seungwon Kook, and Yohan Jo. Non-collaborative user simulators for tool agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=UAUimofy3W.

Woojung Song, Dongmin Choi, Yoonah Park, Jongwook Han, Eun-Ju Lee, and Yohan Jo. Human psychometric questionnaires mischaracterize llm behavior, 2026. URL https://arxiv.org/abs/2509.10078.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao 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.

Tianqi Xu, Linyao Chen, Dai-Jie Wu, Yanjun Chen, Zecheng Zhang, Xiang Yao, Zhiqiang Xie, Yongchao Chen, Shilong Liu, Bochen Qian, Anjie Yang, Zhaoxuan Jin, Jianbo Deng, Philip Torr, Bernard Ghanem, and Guohao Li. CRAB: Cross-environment agent benchmark for multimodal language model agents. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 21607–21647, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1113. URL https://aclanthology.org/2025.findings-acl.1113/.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik R Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=roNSXZpUDN.

Fulei Zhang and Zhou Yu. Mind the gap: Linguistic divergence and adaptation strategies in human-llm assistant vs. human-human interactions, 2025. URL https://arxiv.org/abs/2510.02645.

Wenlong Zhao, Debanjan Mondal, Niket Tandon, Danica Dillion, Kurt Gray, and Yuling Gu. WorldValuesBench: A large-scale benchmark dataset for multi-cultural value awareness of language models. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 17696–17706, Torino, Italia, May 2024a. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.1539/.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=Bl8u7ZRlbM.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham 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 DOMAINS

Table 4: The 17 task domains and 86 subdomains derived from AEI request categories, grouped by platform.

#	Domain	Subdomains
claude.ai-derived domains (56 subdomains)		
1	Business Marketing Operations	Business Presentations & Process Diagrams
		Startup & Business Strategy
		Digital Marketing & Social Media
		Financial Analysis & Spreadsheets
		Company Research & Business Intelligence
		Business Operations & Logistics
2	Academic Education Research	Educational Curriculum & Teaching Materials
		Social Sciences & Humanities
		Academic Research Methods & Statistical Analysis
		Academic Coursework & Research Writing
		Books & Academic Literature
		Education & Healthcare Training
3	Health Wellness Relationships	Personal Development & Organizational Leadership
		Recipes & Nutrition Planning
		Beauty Fashion & Wellness Products
		Fitness Training & Weight Management
		Psychology & Neuroscience
		Family Life & Social Benefits
		Medical & Health
4	STEM Sciences Engineering	Math & Physics Problems
		Earth & Space Sciences
		Multidisciplinary Engineering
		Chemistry & Life Sciences
		AI & Machine Learning
		Optimization Algorithms & System Performance
5	Creative Media Design	UI and Brand Design
		Architecture & Visual Arts
		Fiction & Public Affairs
		Video & Podcast Production
		Music & Film/TV
		Video Games & Tabletop Roleplay
6	Legal Finance Careers	Personal Finance & Investing
		Legal Documents & Law
		Job Applications & Career Development
		Immigration Visas & Real Estate
7	Software Data Systems	Network & Domain Administration
		File Systems & Storage
		Software & Device Configuration
		Software Project Management & Mobile Apps
		Blockchain & Software Development
		Geospatial, Cloud & Data Engineering
		Conversation History & Mixed Queries
		Document & Image Data Extraction
8	Places People Planning	Product Shopping Recommendations
		Travel Planning & Local Places
		Geographic Demographics & Regions
		Scheduling & Time Management
		Celebrations, Gifts & Weddings
		Names, Genealogy & Biographies

Continued on next page

Table 4 (continued)

#	Domain	Subdomains
9	Home Vehicle Devices	Home Energy & Utility Systems Audio Equipment & Device Troubleshooting Vehicle Ownership & Mechanics
10	Security Codes Safety	Cybersecurity & Workplace Safety Codes, Puzzles & Lottery
11	Sports Performance Management	Sports Data & Events Sports Management & Coaching
First-party API-derived domains (30 subdomains)		
12	Business Intelligence Operations	Business Intelligence & Financial Markets Business Communications & Quality Evaluation B2B Sales Enablement & Intelligence Business Operations & Project Management Business Strategy & Startup Fundraising Business Intelligence & Professional Profiles Recruitment & Job Matching Business Documents & Spreadsheets
13	Consumer Commerce Services	SEO & Digital Marketing Copy Automotive Sales & Logistics Appointments & Travel Real Estate & Digital Advertising Consumer Products & E-commerce Payments & Consumer Finance
14	Education Health Research	Education & Academic Tutoring Clinical Data & Documentation Academic Research Data & Writing Mental Health & Crisis Intervention
15	Regulated Document Analysis	Document Intelligence Legal Documents & Research Industrial Product Data & Trade Compliance
16	Technical Systems Security	Technical Documentation Systems AI System Evaluation & Security Cybersecurity Operations & Threat Analysis IT Systems & Network Devices AI Agents & Game Interaction
17	Visual Geospatial Media	Digital Design & Data Visualization AI Video Production Workflows Image & Screenshot Analysis Energy & Environmental Geospatial Data

Source inventory and filtering. We collect request-cluster names from four public AEI releases (March 27, 2025; September 15, 2025; January 15, 2026; and March 24, 2026) and deduplicate them, yielding 2,821 unique request categories. A fixed LLM rubric retains categories describing agentic, open-ended tasks that admit several reasonable approaches or outputs and excludes categories primarily concerned with software engineering. This leaves 1,058 unique categories (876 claude.ai records and 266 first-party API records, with 84 appearing on both platforms). Categories may occur in both platform inventories.

Domain derivation. We process the platform inventories separately. We use Qwen3-Embedding-8B to embed the retained category names and apply adaptive Ward clustering, yielding 56 claude.ai subdomains and 30 first-party API subdomains. An LLM names and groups these subdomains into 11 claude.ai domains and 6 API domains. Table 4 lists all 17 domains and their 86 subdomains. For HABIT, each construction and held-out split contains one task per domain; AGENTHABIT contains one task per subdomain.

B TASK CONSTRUCTION AND USER CONTEXT

This section describes the 34 tasks used to construct and evaluate HABIT and the private user context shown as *user information* in Figure 3.

Each domain contains a construction task and a held-out task, with different user situations, requested outputs, and private contexts. The authors develop the tasks from aggregate AEI request categories. GPT-5.6 Sol adds output requirements and optional follow-up requests to the user contexts. Authors refine these additions for consistency, coverage of user needs, and feasible information-disclosure conditions. Automated checks validate the schema, and the authors review and revise every task.

Table 5 lists the opening requests for all 34 tasks.

Table 5: Verbatim opening requests for the 17 construction and 17 held-out tasks.

Domain	Task	Opening request
Construction tasks		
Business Marketing Operations	Marketing budget	How should I split our $12,000 monthly ad budget next quarter across Google Search, Instagram, and paid influencers?
Academic Education Research	Science unit plan	Can you help me plan eight 50-minute lessons on energy transfer for a class of 28?
Health Wellness Relationships	Skin products	I need an affordable skincare routine to help with ongoing breakouts along my jawline. I'd like to keep it around $60 a month.
STEM Sciences Engineering	Home solar	I own a house in New Jersey and spend about $2,400 a year on electricity. Are rooftop solar panels worth the money, and should I buy them?
Creative Media Design	Signup flow	Could you recommend a better order for our seven-screen mobile web signup flow based on last month's funnel? Signup drop-off is bad, and I want an actionable sequencing change.
Legal Finance Careers	Going freelance	Can you give me an actionable checklist for leaving my salaried graphic design job and going freelance?
Software Data Systems	DB backfill	I need a script to normalize the country column in our production Postgres users table in one pass.
Places People Planning	Neighborhood pick	Rank the top three Boston neighbourhoods for finding a flat within my firm $3,200 monthly budget and commuting to Kendall Square in under an hour.
Home Vehicle Devices	Heating bill	how can i get my UK energy bill down without letting the flat drop below 20c, ive got a 3 month old so thats not negotiable
Security Codes Safety	Email breach	Someone got into my Gmail, and there are messages in my sent folder that I did not write. Give me a numbered list of what to do, in order.
Sports Performance Management	League schedule	I need a fixture list for our eight-team Saturday league where everyone plays everyone once. We have ten Saturdays and two pitches.
Business Intelligence Operations	Survey priorities	I need to turn our internal survey on workplace tools into a one-slide recommendation naming the single priority to fix next quarter.
Consumer Commerce Services	Blog SEO	Our organic traffic is down about 70% in three months. Help me figure out how to recover it without just writing more blog posts.
Education Health Research	Kid math	My eleven-year-old's maths marks are slipping, and I'm considering increasing tutoring from one hour to three hours a week. Is that advisable, and how should I structure the three hours?
Regulated Document Analysis	Loan compare	which loan should i go for on £40,000 over five years, offer A fixed 5.2% or offer B variable 4.6%?
Technical Systems Security	API docs	Could you recommend one tool for updating and maintaining our internal REST API documentation? Please give me a concrete choice rather than a comparison.

Continued on next page

Table 5 (continued)

Domain	Task	Opening request
Visual Geospatial Media	Sales dashboard	Can you help me design a sales dashboard for leadership that makes current-month revenue immediately visible?
Held-out tasks		
Business Marketing Operations	Agency switch	Should we keep, renegotiate or leave our marketing agency? We're six months in on £6,500 a month and inbound leads are down.
Academic Education Research	Lit review	I need help planning a 6,000-word literature review on whether school uniform policy affects attendance. I have six weeks until submission.
Health Wellness Relationships	Sleep fix	I keep waking around 3 a.m. and can't get back to sleep. I need a plan that actually targets that.
STEM Sciences Engineering	Well water	I bought a property with a private well and need help understanding the water test. Can you compare each result with the relevant health or aesthetic benchmark and tell me what requires action?
Creative Media Design	Brand refresh	I want to redo our bakery brand so it feels less rustic and more grown up. We need it ready in about three weeks for new packaging.
Legal Finance Careers	Equity offer	Should I take the startup offer or stay at my current job? I want a clear recommendation, not just pros and cons.
Software Data Systems	Slow query	My Django/Postgres dashboard takes about eight seconds to load. What should I do to speed it up?
Places People Planning	Office move	need help finding a cheaper office in Bristol for 30 people, max budget is £4,500 a month.
Home Vehicle Devices	Used EV	Can you sanity-check a 2021 electric car with 68,000 km and a £16,500 asking price? Tell me if it's reasonable value and whether I should buy it.
Security Codes Safety	Shared credentials	Which password manager should we buy for our six-person team to replace the passwords we currently keep in a pinned Slack message?
Sports Performance Management	Injury return	I need a return-to-play plan for a player nine weeks out from a lower-leg fracture, aiming to be available for a tournament in five weeks.
Business Intelligence Operations	Forecast gap	Help me explain why Q4 revenue came in at $3.1M against a $4.2M forecast for Thursday's board meeting.
Consumer Commerce Services	Return policy	I need to tighten my returns policy, 7 days for change of mind and customer pays return postage. can you help word it
Education Health Research	Screen time	I need a daily tablet limit for my seven-year-old and a specific app or router control that automatically blocks it when time's up.
Regulated Document Analysis	Tenancy clause	Are these tenancy clauses enforceable in England: a £75 charge for late rent and allowing the landlord to enter at any time for inspection?
Technical Systems Security	Incident postmortem	I need help writing a blameless postmortem for the three-hour payments outage Tuesday. its going to leadership Friday.
Visual Geospatial Media	Branch map	I need a map showing how each of our 40 branches is performing.

Each user context specifies the situation, output requirements, optional follow-up requests after a draft, and five classes of information or decisions. *Volunteered* facts may appear in the opening request; *on-request* facts are disclosed when the agent asks about them; *guarded* facts additionally require a specified conversational condition; *withheld* facts remain unavailable; and *delegated* decisions are left to the agent. Only the user simulator receives the full context; agents, axis proposers, and annotation judges do not.

With user context, GPT-5.6 Sol generates one opening request per task from the situation, volunteered facts, and assigned writing style. We cache this request and reuse it unchanged across models and runs. Later simulator responses depend on the agent's questions and deliverables under the same disclosure rules. We use `gpt-5.6-sol` as the user simulator.

Table 6: Model routes and run settings. † marks the eight HABIT construction models; T is temperature and R is requested reasoning effort.

Family	Model route	Access route	Run settings
GPT	`gpt-5.6-sol`†	OpenAI API	$R =$ medium
	`gpt-5.6-terra`†	OpenAI API	$R =$ medium
	`gpt-5.6-luna`	OpenAI API	$R =$ medium
GPT-OSS	`openai/gpt-oss-120b`	Local vLLM	$T = 0.7$
Claude	`anthropic/claude-opus-5`†	OpenRouter (Anthropic)	$R =$ medium
	`anthropic/claude-sonnet-5`†	OpenRouter (Anthropic)	$R =$ medium
	`anthropic/claude-haiku-4.5`	OpenRouter (Anthropic)	$R =$ medium
Qwen	`qwen/qwen3.7-max`†	OpenRouter (Alibaba)	$T = 0.7$; $R =$ medium
	`qwen/qwen3.7-plus`†	OpenRouter (Alibaba)	$T = 0.7$; $R =$ medium
	`qwen/qwen3.7-flash`	OpenRouter (Alibaba)	$T = 0.7$; $R =$ medium
	`Qwen/Qwen3.8-27B` (judge)	Local vLLM	$R =$ low
Gemini	`google/gemini-3.7-flash`†	OpenRouter (Google Vertex)	$R =$ medium
DeepSeek	`deepseek-ai/DeepSeek-V4-Flash-0731`†	Local vLLM	$T = 0.7$; thinking off
Gemma	`google/gemma-4-31B-it`	Local vLLM	$T = 0.7$; thinking on
	`google/gemma-4-26B-A4B-it`	Local vLLM	$T = 0.7$; thinking on
	`google/gemma-4-12B-it`	Local vLLM	$T = 0.7$; thinking on
Nemotron	`nvidia/NVIDIA-Nemotron-3.5-Lightning-30B-A3B-BF16`	Local vLLM	$T = 0.7$; thinking on
Muse	`meta/muse-spark-1.3-contributor`	OpenRouter (Meta)	$T = 0.7$; $R =$ medium
	`meta-models/Muse-Glimmer-30B` (judge candidate)	Local vLLM	$R =$ low
Solar	`upstage/solar-pro4`	OpenRouter (Upstage)	$T = 0.7$

C MODEL ROSTER

We use eight models to construct HABIT and evaluate 18 models on AGENTHABIT (these eight plus ten additional models). Table 6 lists the model routes and run settings. We use Qwen3.8-27B as the LLM judge. When evaluating GPT-5.6 Luna as a judge candidate, we set its reasoning effort to low.

D CANDIDATE AXIS PROPOSAL, MERGING, AND ANNOTATION

Human proposal. Three authors independently review all 24 trajectories for each of the 17 construction tasks and propose axes with a guide and viewer. To reduce anchoring, the guide instructs proposers to read all trajectories for a task before proposing axes and sets no target number of axes. The viewer displays agent and user messages, and the agent's tool calls, while hiding private user contexts and simulator state. Each proposed axis specifies a behavioral definition, categorical labels including `N/A`, and a decide-by rule.

LLM proposal. We use Claude Opus 5, GPT-5.6 Sol, and Grok 4.6 as LLM proposers. Three LLMs independently propose axes for 17 construction tasks, with all 24 trajectories for one task supplied in each call. The prompt instructs each model to review every trajectory before proposing axes and does not provide a target count or any example of an axis. To minimize bias, we assign random trajectory IDs and omit model, provider, and simulator metadata from the inputs.

Axis merging and human refinement. After deduplicating identical human rubrics, GPT-5.6 Sol merges the human and LLM proposals using their definitions, labels, decide-by rules, and task identifiers. The input excludes trial-label counts and explicit proposer-source fields. The merger records combined and discarded proposals. Three human reviewers examine all 29 preliminary axes, revising rubrics and merging or removing axes to obtain 23 behavioral axes. Appendix I reports proposal counts and the final inventory.

Annotation records. Each annotation records one rubric label and supporting evidence, indexed by trajectory, axis, judge, and repetition. Invalid labels, parse failures, timeouts, and truncated responses are logged as errors and retried. We annotate all 23 axes using the judges and input restrictions specified in Appendix F.

E User Simulator

E.1 Persona Construction

To give personas realistic writing styles and feedback behaviors, we define style along grammatical fluency, politeness, and lexical diversity, following criteria informed by Zhang & Yu (2025), and categorize feedback with a taxonomy adapted from Don-Yehiya et al. (2025), covering clarification, request rephrasing, and corrective feedback with or without an explicit diagnosis. We measure the distributions of these attributes in user turns following substantive assistant responses in WildChat-1M (Zhao et al., 2024b) and sample each persona's style and feedback disposition from them. Task expertise and tolerance for imperfect outputs are assigned directly. Simulator prompts describe each attribute behaviorally, and a task's persona remains fixed across models and runs.

E.2 Human Evaluation

Evaluation setup. Three annotators who are not authors of this paper independently evaluate 74 simulator utterances from the 17 construction trajectories used for judge validation (Appendix G). Each annotator rates every utterance on three criteria, yielding 666 ratings in total.

Rating criteria and procedure. Annotators assess *naturalness*, *contextual appropriateness*, and *role and self-consistency* on a five-point scale using the rubric in Table 7. Judgments use only the conversation up to each utterance, without access to the hidden persona. Opening requests are assessed for standalone plausibility and internal coherence.

Analysis. We report means and sample standard deviations over 222 ratings per criterion, along with multi-rater Gwet's AC1. We estimate 95% confidence intervals using 5,000 trajectory-level bootstrap resamples, retaining all utterances and ratings within each trajectory.

Results. Mean ratings range from 4.55 to 4.68 out of 5, with AC1 values of 0.621–0.714 (Table 8).

Table 7: the five-point rubric used to evaluate simulator utterances. Each utterance receives one score for each criterion.

Score	Naturalness	Contextual appropriateness	Role and self-consistency
1	Wording is very unnatural or mechanical, making it difficult to regard as a real user's utterance.	Largely unrelated to the preceding question or deliverable, or clearly inconsistent with the conversation.	Clearly contradicts the user's previously stated role, goals, or facts.
2	Expressions are noticeably awkward, repetitive, or formulaic.	Responds to only part of the preceding content, or the timing or direction of the response is substantially awkward.	Changes important goals, circumstances, or constraints without explanation.
3	Understandable, but the tone or sentence flow is noticeably awkward.	Generally connected to the context, but misses an important part of the question or deliverable.	Could be the same user, but shows noticeable tension or ambiguity relative to earlier utterances.
4	Sounds like a real user, with only minor awkwardness.	Responds appropriately to the preceding content, with only minor contextual mismatches.	Generally consistent with earlier information, with only minor tensions.
5	Very natural wording for a real user in the given situation.	Responds accurately and naturally to the preceding question or deliverable.	Consistently maintains the user's role, goals, facts, and constraints.

Table 8: Human evaluation of 74 simulator utterances by three annotators. Means and sample standard deviations use 222 ratings per criterion. Gwet's AC1 measures exact agreement among all three annotators; its 95% confidence intervals use 5,000 trajectory-level bootstrap resamples.

Criterion	Mean $\pm$ SD	Gwet's AC1	95% CI
Naturalness	4.55 ± 0.87	0.621	[0.513, 0.734]
Contextual appropriateness	4.68 ± 0.71	0.714	[0.616, 0.811]
Role and self-consistency	4.57 ± 0.83	0.624	[0.539, 0.712]

F Taxonomy Evaluation Details

Evaluation setup. We compare three taxonomies on 408 trajectories from eight models performing the 17 held-out tasks (Appendix B). Qwen3.8-27B, Gemma 4 31B, and GPT-5.6 Luna independently annotate each trajectory under each taxonomy, with three repetitions per judge. For HABIT, judges assign labels with supporting evidence using each axis's definition, label descriptions, and decide-by rule. For Schwartz human values, judges rate alignment with each of ten values on a six-point scale. For Act·ONOMY, judges tag agent passages with subactions; a subaction is present if at least one passage receives its tag in that repetition. The resulting representations contain 23 categorical axes, ten six-category values, and 46 binary subactions, respectively. Table 1 reports their comparison.

Label diversity. For each judge and axis, we select each trajectory's modal label across three repetitions and compute normalized entropy:

$$H_a = -\frac{1}{\log K_a} \sum_{k=1}^{K_a} p_{ak} \log p_{ak},$$

where K_a is the number of permitted labels and p_{ak} is the proportion of trajectories assigned label k. We set $0 \log 0 = 0$, retain `N/A` where permitted, and resolve modal ties by lexicographic label order. Higher entropy indicates more even label use.

Judgment consistency. We measure judgment consistency using nominal-category Gwet's AC1. For each judge and axis, intra-judge AC1 is computed once after pooling all three repetition pairs

from every trajectory. Inter-judge AC1 compares judges' modal labels and averages the three pairwise coefficients on each axis, using the same lexicographic tie rule.

Model-separation gap. We measure model separation on each axis using the difference between within-model and cross-model label agreement on the same held-out task. For axis a, judge j, task t, and annotation repetition r, let W_{ajtr} be the fraction of unordered pairs of distinct trajectories from the same model that receive identical labels, and let C_{ajtr} be the corresponding fraction for trajectories from different models. Each agreement rate weights its eligible trajectory pairs equally. We define

$$G_{ajtr} = W_{ajtr} - C_{ajtr}. \tag{1}$$

A positive gap indicates greater label agreement within models than across models on the same task. We compute the gap separately for each annotation repetition, then average equally over the three repetitions, eligible tasks, judges with defined estimates, and axes with defined estimates, in that order. Within-model and cross-model agreement use the same aggregation. This measure uses the existing held-out labels without fitting a classifier.

For each axis and judge, a task must retain at least 12 trajectories and at least two models with two or more trajectories each. The same retained trajectories are used in all three annotation repetitions. We exclude tasks whose labels are constant in every repetition, and omit an axis–judge estimate if no eligible task remains.

For HABIT, we apply a conditional label-handling rule separately for each axis and judge. If the overall `N/A` rate across held-out trajectories and annotation repetitions is at least 20% and the range of model-specific `N/A` rates is at least 30 percentage points, we restrict the gap calculation to trajectories receiving a non-`N/A` label in all three repetitions before applying the task eligibility rules. Otherwise, `N/A` remains a category. This condition applies only to *Execution tool role* for all three judges. It does not change the labels used for entropy, AC1, or model matching.

Table 9 reports the resulting agreement rates and gaps. The gap is defined on 23 axes for HABIT, ten for Schwartz values, and 36 for Act·ONOMY. We obtain 95% percentile intervals from 5,000 bootstrap resamples of axes with defined gaps within each taxonomy. These intervals describe variation across the evaluated axes; they are not confidence intervals for performance on a new task population.

Table 9: Model-separation gap on the held-out trajectories. Within and Cross denote same-model and different-model label agreement within a task, respectively. The Axes column gives the number of axes with defined gaps. Intervals summarize axis-bootstrap variation.

Taxonomy	Axes	Within	Cross	Gap	95% interval
HABIT	23	0.639	0.525	0.114	$[0.080, 0.148]$
Schwartz human values	10	0.616	0.569	0.047	$[0.030, 0.065]$
Act·ONOMY	36	0.812	0.769	0.044	$[0.029, 0.061]$

Model-matching accuracy. We measure model-matching accuracy (Dunlap et al., 2025) using the existing categorical trajectory labels. For each judge, we concatenate one-hot encodings of all axes and average the three annotation repetitions within each trajectory, treating all labels as nominal and retaining `N/A`. For each of the 28 model pairs, we form all 3×3 trajectory-repeat pairs within each task and fit a separate linear logistic classifier to their feature differences. We leave one entire task out, training on 16 tasks and evaluating on the remaining task, for 17 folds. All trajectories and reversed pairs from a task remain in the same fold. Classifier settings and statistical comparisons are specified in Appendix H.

Aggregation and label handling. Entropy and intra-judge AC1 are averaged across judges within each axis, then equally across axes. Inter-judge AC1 is averaged across judge pairs within each axis, then equally across axes. Coefficients that are undefined (e.g., when an axis has only one observed label) are omitted from the averages. Model-matching accuracy gives ties half credit and averages equally over trajectory-repeat pairs, model pairs, tasks, and judges; the random baseline is 50%. Entropy, AC1, and model-matching accuracy retain `N/A`; the gap applies the conditional

label-handling rule described above. The full representations contain 116, 60, and 92 features for HABIT, Schwartz, and Act·ONOMY.

We compare inter-judge AC1 by independently bootstrapping the coordinates with defined coefficients within each taxonomy 5,000 times. The HABIT–Act·ONOMY difference is -0.037 (95% interval $[-0.143, 0.073]$; two-sided bootstrap tail $= 0.5199$). The HABIT–Schwartz difference is 0.335 ($[0.121, 0.540]$; 0.0032), and the Act·ONOMY–Schwartz difference is 0.372 ($[0.161, 0.562]$; 0.0008). Thus, both exceed Schwartz values, while HABIT and Act·ONOMY are not statistically distinguished.

G HUMAN VALIDATION AND JUDGE SELECTION

Annotation setup and completeness. Three human annotators independently annotate 23 HABIT axes on the same 17 trajectories, using the same trajectories and rubrics as the LLM judges. Each annotator annotates 391 labels, yielding 1,173 human labels in total. The three annotators agree unanimously on 185 of the 391 cells (47.3%), two agree on 179 (45.8%), and all three disagree on 27 (6.9%). We use the 364 of 391 cells (93.1%) that have a strict human majority for judge selection.

Agreement measures. To assess judge–human agreement, we compare four models (Qwen3.8-27B, Gemma 4 31B, Muse-Glimmer-30B, and GPT-5.6 Luna) against human annotations. Each model labels 23 axes on each of the 17 tasks three times. We measure pooled observed agreement, Cohen's κ, and Gwet's AC1, plus Fleiss' κ and nominal Krippendorff's α for agreement among the three annotators. Confidence intervals and judge selection both use a trajectory-level bootstrap with 2,000 paired resamples.

Human inter-rater agreement. Pooled human observed agreement is 0.6257 (trajectory-bootstrap 95% CI: 0.58–0.67), with Cohen's $\kappa = 0.6096$, Gwet's AC1 $= 0.6203$, Fleiss' $\kappa = 0.6092$, and Krippendorff's $\alpha = 0.6095$. Table 10 reports pairwise agreement. We use this level as a descriptive reference for judge–human agreement; it falls at the boundary between moderate and substantial agreement (Landis & Koch, 1977).

Table 10: Human pairwise agreement on 391 cells, with trajectory-bootstrap 95% intervals.

Annotator pair	Agreement	95% CI
H1–H2	0.6675	[0.611, 0.729]
H1–H3	0.5882	[0.537, 0.642]
H2–H3	0.6215	[0.558, 0.680]

Direct human–judge agreement. Table 11 compares individual human labels with judge labels. Across the 12 human–judge pairs, mean observed agreement is 0.6929, mean Cohen's κ is 0.6809, and mean Gwet's AC1 is 0.6884. Qwen agrees most with H1 and H2, and Gemma with H3. Luna's agreement ranges from 0.5831 to 0.7954 across annotators.

Table 11: Agreement between each human and each judge's modal labels on all 391 cells.

Annotator	Qwen	Gemma	Muse	GPT-5.6 Luna (low)
H1	0.7008	0.6726	0.6368	0.6215
H2	0.8235	0.8056	0.7570	0.7954
H3	0.6496	0.6650	0.6036	0.5831

Preregistered judge selection. To select the judge, we use pooled Gwet's AC1 against the strict human majority label. Judges are compared using percentile intervals from a paired trajectory-level bootstrap. If the leading judges cannot be distinguished, ties are broken first by mean axis-level AC1. Table 12 reports agreement with the human majority, and Table 13 reports the paired differences between judges.

The Qwen–Gemma paired interval includes zero, so the tie-break selects Qwen by mean axis-level AC1 (0.7878 versus 0.7659). We use Qwen3.8-27B for the benchmark profiles in Section 3.1.

Table 12: Judge agreement with the human majority on 364 cells.

Judge	Agreement	Cohen's κ	Pooled AC1	AC1 95% CI	Mean axis AC1
Qwen3.8-27B	0.8352	0.8282	0.8327	[0.785, 0.877]	0.7878
Gemma 4 31B	0.8159	0.8080	0.8132	[0.755, 0.864]	0.7659
Muse-Glimmer-30B	0.7555	0.7450	0.7519	[0.700, 0.801]	0.6951
GPT-5.6 Luna (low)	0.7527	0.7435	0.7489	[0.700, 0.794]	0.6728

Table 13: Paired differences in pooled human-majority AC1, with 95% intervals from 2,000 trajectory-bootstrap resamples.

Comparison	AC1 difference	Paired 95% CI
Qwen − Gemma	0.0196	[−0.0164, 0.0553]
Qwen − Muse	0.0808	[0.0448, 0.1188]
Qwen − Luna	0.0838	[0.0471, 0.1192]
Gemma − Muse	0.0612	[0.0111, 0.1087]
Gemma − Luna	0.0643	[0.0195, 0.1062]
Muse − Luna	0.0030	[−0.0333, 0.0428]

H Statistical Specification

Model-matching classifier. For each judge and model pair (A, B), let $\boldsymbol{d} = \boldsymbol{x}_A - \boldsymbol{x}_B$ be the difference between trajectory features defined in Appendix F. Reversing the pair negates both the features and the class label. With equal weight on both orientations and no intercept, the logistic objective reduces to

$$\widehat{\boldsymbol{w}}_{-t} = \arg\min_{\boldsymbol{w}} \left\{ \frac{1}{|\mathcal{D}_{-t}|} \sum_{\boldsymbol{d} \in \mathcal{D}_{-t}} \log\bigl(1 + \exp(-\boldsymbol{w}^\top \boldsymbol{d})\bigr) + \frac{\lambda}{2} \|\boldsymbol{w}\|_2^2 \right\}, \tag{2}$$

where $\mathcal{D}_{-t}$ contains the trajectory-pair differences from the 16 training tasks and $\lambda = 0.01$ for all taxonomies. The sign of $\widehat{\boldsymbol{w}}_{-t}^\top \boldsymbol{d}$ determines the assignment on task t; margins with absolute value at most 10^{-10} receive half credit. We use no feature scaling, feature selection, or hyperparameter tuning. The metric measures pairwise identification of the eight known models on unseen tasks.

Framework comparisons. We first average the dependent trajectory pairs, model pairs, and judges within each task, then compare the resulting 17 paired task accuracies. To approximately account for overlapping cross-validation training sets, we apply the variance correction of Nadeau & Bengio (1999). For task-level accuracy differences δ_t, with mean $\bar{\delta}$ and sample variance s_δ^2, the test statistic is

$$T = \frac{\bar{\delta}}{\sqrt{(1/17 + 1/16) s_\delta^2}}. \tag{3}$$

We use a two-sided t reference distribution with 16 degrees of freedom and Holm correction across the three taxonomy comparisons. HABIT exceeds Schwartz values by 19.46 percentage points (adjusted $p = 2.18 \times 10^{-6}$) and Act·ONOMY by 5.05 points ($p = 0.00236$); Act·ONOMY exceeds Schwartz values by 14.41 points ($p = 7.77 \times 10^{-5}$). These exploratory comparisons assume exchangeable tasks and hold the models, judges, axes, and classifier settings fixed. The cross-validation correction is approximate and does not guarantee exact Type I error control.

Annotation reliability. For each judge and axis, we pool all three repetition pairs from every trajectory and compute observed agreement p_o, Cohen's κ, and Gwet's AC1 once on the pooled pairs:

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \quad p_e = \sum_k \pi_k^{(L)} \pi_k^{(R)}; \qquad \text{AC1} = \frac{p_o - p_e^\gamma}{1 - p_e^\gamma}, \quad p_e^\gamma = \frac{1}{K-1} \sum_k \pi_k (1 - \pi_k), \tag{4}$$

where $\pi_k^{(L)}$ and $\pi_k^{(R)}$ are label shares on the left and right sides of the pooled pairs, $\pi_k = (\pi_k^{(L)} + \pi_k^{(R)})/2$, and K is the number of observed labels, including `N/A` when present. Undefined coefficients ($K < 2$ for AC1) are omitted. We base the reliability criterion on AC1 because κ can be low under skewed label distributions even when the judge rarely disagrees with itself.

I BEHAVIORAL AXIS INVENTORY

Figures 6–11 present the 23 HABIT axes alphabetically. Each axis gives the definition, categorical labels, and decide-by rule; colors identify the five behavioral categories. `N/A` retains its axis-specific meaning.

The initial pool contains 716 proposals: 133 from humans and 583 from LLMs. Deduplicating human rubrics leaves 62 human proposals and 645 combined proposals. LLM merging reduces these to 29 preliminary axes, and human refinement yields the final 23 axes (Appendix D).

01 Output Design & Delivery

Artifact completeness

Whether the last answer supplies the requested artifact in the required form and with its major parts.

Complete usable: full artifact usable without assembling separate replies.

Complete with placeholders: complete structure with explicit blanks for genuine unknowns.

Fragmentary: fragments, pseudocode or incomplete structure.

Explanation only: discusses/promises artifact without supplying it.

No output: no substantive output.

N/A: no concrete artifact requested.

Decide by. Inspect the last answer for requested format and major required parts. Judge form, not correctness.

02 User Interaction & Collaboration

Ask user timing

When the agent requests user input relative to its first substantive deliverable.

Before only: all requests for user input occur before the first substantive deliverable.

After only: all requests for user input occur after the first substantive deliverable.

Both phases: Requests for user input occur both before and after the first substantive deliverable.

Not asked: no requests for user input occur.

N/A: insufficient evidence.

Decide by. Inspect all requests for user input, in prose or tools, relative to the first substantive deliverable. Exclude rhetorical questions and generic offers of further help.

03 Decision-Making & Justification

Certainty and uncertainty framing

How the main substantive conclusion expresses certainty, uncertainty, assumptions, or limitations.

Explicit certainty: explicitly certain, definitive or settled.

Unqualified conclusion: one conclusion without uncertainty, assumptions or limitations.

Qualified conclusion: one main conclusion with relevant uncertainty, assumptions or limitations.

Explicit uncertainty: materially unresolved or multiple plausible alternatives remain open.

Withholds conclusion: no substantive conclusion because evidence is insufficient.

N/A: no substantive answer.

Decide by. Inspect the main substantive conclusion. Ignore hedges that do not change the degree of certainty.

04 Task Management & Validation

Constraint conflict handling

How the agent handles a request whose constraints cannot all be satisfied.

States infeasibility only: explains infeasibility with no feasible alternative.

States and offers alternative: explains incompatibility and offers feasible alternatives.

Silently changes request: uses a feasible variant without identifying infeasibility.

Fabricates feasibility: appears to satisfy the request by undisclosed constraint violations/changes.

N/A: feasible, or feasibility cannot be assessed.

Decide by. First establish independently whether constraints can all be satisfied, then inspect the response. Judge disclosure of changes to constraints, not general answer quality.

Figure 6: HABIT behavioral axis inventory: axes 1–4.

05 Output Design & Delivery

Constraint preservation across outputs

How applicable requirements, corrections, and retained content carry forward across successive requested outputs.

All preserved: all applicable constraints retained in all later substantive outputs.

Recovered after reminder: lapses restored after reminders and retained thereafter.

Inconsistently preserved: applied in some relevant outputs, omitted/changed in others.

Never applied: an applicable known constraint never incorporated into requested output.

N/A: no carry-forward constraint or no later output to assess.

Decide by. Track applicable requirements, corrections and retained content through successive requested outputs. A clearly connected patch counts; content not targeted for change remains a constraint. Use the worst persistent treatment unless all lapses are permanently corrected after reminders.

06 Output Design & Delivery

Decorative symbol use

The extent of decorative symbols in delivered formatting.

None: no decorative symbols.

Limited: a few decorative symbols.

Extensive: decorative symbols throughout the deliverable.

N/A: no deliverable, or inapplicable.

Decide by. Inspect delivered formatting. Exclude ordinary punctuation, semantic arrows and Markdown rules. Break ties by breadth across sections.

07 Information Gathering & Tool Use

Execution tool role

The role of executed code or calculation tools in producing the delivered result.

Primary calculation: produces numerical/analytical results used in the answer.

Verification: checks a result, artifact, syntax or behavior produced by another route.

Artifact generation: creates or renders the user-facing artifact.

Environment inspection: explores files, directories, packages or runtime.

Text rendering or placeholder: merely prints supplied prose/values or a trivial placeholder.

N/A: no execution tool.

Decide by. Inspect executed code and output and choose the role most central to the delivered result. Meaningful calculation, verification, artifact generation or inspection takes precedence over incidental printing.

08 Information Gathering & Tool Use

Execution tool use

When code or calculation tools are actually executed relative to the first substantive deliverable.

Before only: all executions of code or calculation tools occur before the first substantive deliverable.

After only: all executions of code or calculation tools occur after the first substantive deliverable.

Both phases: Executions of code or calculation tools occur both before and after the first substantive deliverable.

Not used: no executions of code or calculation tools occur.

N/A: insufficient evidence.

Decide by. Inspect actually executed code/calculation tools relative to the first substantive deliverable. Written but unexecuted code/arithmetic does not count.

Figure 7: HABIT behavioral axis inventory: axes 5–8.

09 Task Management & Validation

Initial result check

The visible checking of the first checkable result, its basis, or its constraints before material user feedback.

Tool check: relevant tool checks the result/basis/constraints.

Manual check: visible equation, comparison, count or checklist without a relevant tool check.

Claimed not shown: claims checking without showing it.

No visible check: checkable result without shown/claimed checking.

N/A: no checkable result.

Decide by. Inspect checks around the first checkable result, before any material user update/challenge. A check can address the result, its basis or constraints. Tool generation alone does not count. Precedence: tool, manual check, unsupported claim of checking, no check.

10 User Interaction & Collaboration

Missing information handling

How the agent responds when information needed for the request is missing or uncertain.

Asks and waits: requests the information and waits before providing dependent content.

Proceeds and asks: provides dependent content while asking for the information.

Proceeds with flagged assumptions: proceeds without asking, identifying assumptions, proxies, placeholders or uncertainty.

Proceeds with unflagged assumptions: proceeds without asking and treats unsupported information as established.

Omits dependent content: omits requested dependent content without asking or explaining the omission.

N/A: no necessary information is missing, or the response to the gap is unobservable.

Decide by. Identify the first missing fact that could change the result and judge the first response to that gap. Precedence: blocking question, question with dependent content, flagged assumptions, unsupported use or unexplained omission.

11 Output Design & Delivery

Next step guidance

The concrete next actions, questions, or invitations provided with the deliverable.

None: no next step or offer.

One targeted step: one concrete next step/question.

Multiple targeted steps: two or more substantively distinct steps/questions.

Open ended invitation only: unspecified invitation only.

N/A: no prose, or no reasonable follow-up.

Decide by. Inspect the closing first, then the rest of the deliverable. Count concrete actions/questions; where generic invitations coexist, classify by concrete steps.

12 Decision-Making & Justification

Output option resolution

How the final answer presents and resolves meaningful alternative outputs or approaches.

Single option: one approach/output.

Multiple options with closure: multiple options with a current choice or specific selection rule.

Few options without closure: two or three unresolved options.

Many options without closure: four or more unresolved options.

Deferred pending information: withholds choice pending necessary information.

N/A: no meaningful choice or no substantive output.

Decide by. Inspect the final answer for distinct meaningful outputs/approaches, excluding minor variants. A mere preference does not resolve the choice.

Figure 8: HABIT behavioral axis inventory: axes 9–12.

13 User Interaction & Collaboration

Personalization

How available case-specific facts shape the substance of the response.

Generic: does not incorporate available case-specific facts.

Mentioned only: mentions them without changing the substance.

Tailored: uses them to shape the substance.

N/A: irrelevant or no case-specific facts available.

Decide by. Inspect case-specific facts available before the final substantive answer and whether they change recommendations, steps, calculations or structure.

14 Task Management & Validation

Planning visibility

The visibility and use of planning before and during complex work.

None: no visible plan.

Initial: plan before substantive execution.

Adaptive: planning introduced/revised as needs arise.

Systematic: explicit plan throughout with governing stages/dependencies.

N/A: too simple or no work.

Decide by. Inspect planning before and during complex work. Use SYSTEMATIC only when stages/dependencies actually govern execution.

15 Output Design & Delivery

Practical application support

The extent of operational guidance or copy-ready implementation supplied for the requested work.

None: no practical guidance.

Brief: a few directions/checks.

Step by step: ordered detailed procedure with operational considerations.

Executable artifact: runnable, deployable or copy-ready implementation.

N/A: not implementation-oriented or no result.

Decide by. Inspect operational guidance and copy-ready material that instantiates the requested work.

16 User Interaction & Collaboration

Process narration

How the agent describes its work between tool calls.

Silent: little or no interstitial explanation.

Brief: short framing or status comments.

Detailed: interprets findings, weighs alternatives or revises the approach.

N/A: no tools or no opportunity for interstitial narration.

Decide by. Inspect assistant prose between tool calls, excluding final deliverables. Break ties by the most developed recurring style.

Figure 9: HABIT behavioral axis inventory: axes 13–16.

17 Decision-Making & Justification

Rationale depth

The depth of result-specific reasoning supporting the operative final result.

None: result without substantive rationale.

Brief: short reason or bounded explanation.

Thorough: detailed reasoning connecting evidence, constraints and conclusion.

N/A: no final result or a mechanical request with no meaningful rationale.

Decide by. Inspect the response containing the operative final result. Count result-specific reasons, not generic framing; use coverage of distinct relevant grounds and the more developed explanation to break ties.

18 Task Management & Validation

Recheck after feedback

The rechecking of the affected basis after a material correction, new fact, or challenge.

Tool check: relevant search, execution, calculation or validation.

Manual check: fresh equation, comparison, count or checklist without tool recheck.

Claimed not shown: recheck claimed but not shown.

No visible check: responds without shown or claimed recheck.

N/A: no material feedback, or response cannot be observed.

Decide by. Start at the first material correction, new fact or challenge, then inspect the next substantive response. The recheck must address the affected basis.

19 Output Design & Delivery

Scope adherence

How the final output adheres to the full explicit request in scope.

Strict: complete requested scope without substantive omissions/extras.

Adherent with minor extras: complete scope plus limited nondistracting supplementary material.

Partially adherent: main request preserved but material omission or substantial expansion.

Non adherent: fails/replaces the request or focuses on unrequested material.

N/A: no clear scope or no final output.

Decide by. Compare the final output with the full explicit request. Explanations supporting the answer are not overreach unless they materially broaden or displace it.

20 User Interaction & Collaboration

Self correction framing

How the agent frames its first substantive revision after user criticism.

Explicit admission: names its own mistake, unsupported assumption or omission and corrects it.

Change with reason: identifies the change and reason without accepting fault explicitly.

Change only: identifies the change without a reason.

Neutral compliance: revises without identifying the earlier problem or explaining the change.

N/A: no revision follows criticism.

Decide by. Inspect the opening and change explanation of the first substantive revision after user criticism. First-person fault has precedence.

Figure 10: HABIT behavioral axis inventory: axes 17–20.

21 Decision-Making & Justification

Source support visibility

The extent of identifiable source attribution for material claims in delivered prose.

None: claims without identifiable attribution.

Named sources: identifiable sources without links.

Linked sources: links/formal citations for some material claims.

Comprehensive sourcing: systematic sourcing of material claims.

N/A: no claims requiring sourcing or no answer.

Decide by. Inspect user-facing delivered prose, not raw search results, and coverage of material claims.

22 Information Gathering & Tool Use

Web search frequency

The number of external search calls over the trajectory.

None: 0 search calls.

Light: 1–3 search calls.

Moderate: 4–8 search calls.

Heavy: 9 or more search calls.

N/A: availability or activity unobservable.

Decide by. Count external search calls over the whole transcript, ignoring quality and citations.

23 Information Gathering & Tool Use

Web search timing

When external searches occur relative to the first substantive deliverable.

Before only: all external searches occur before the first substantive deliverable.

After only: all external searches occur after the first substantive deliverable.

Both phases: External searches occur both before and after the first substantive deliverable.

Not used: no external searches occur.

N/A: insufficient evidence.

Decide by. Inspect all searches relative to the first substantive deliverable; ignore count and quality.

Figure 11: HABIT behavioral axis inventory: axes 21–23.

J FULL BEHAVIORAL PROFILES

Figures 12–16 show all 18 models on the 23 HABIT axes.

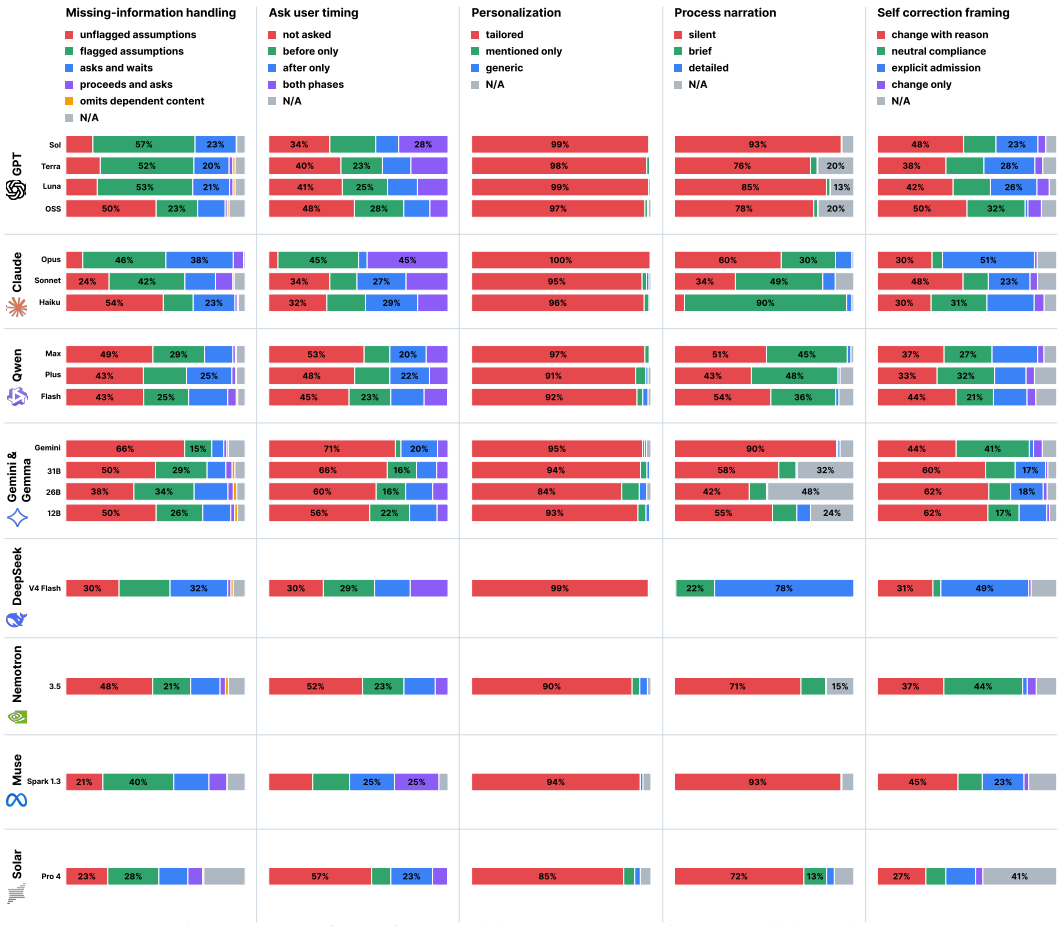


Figure 12: Profiles of 18 models: User Interaction & Collaboration.

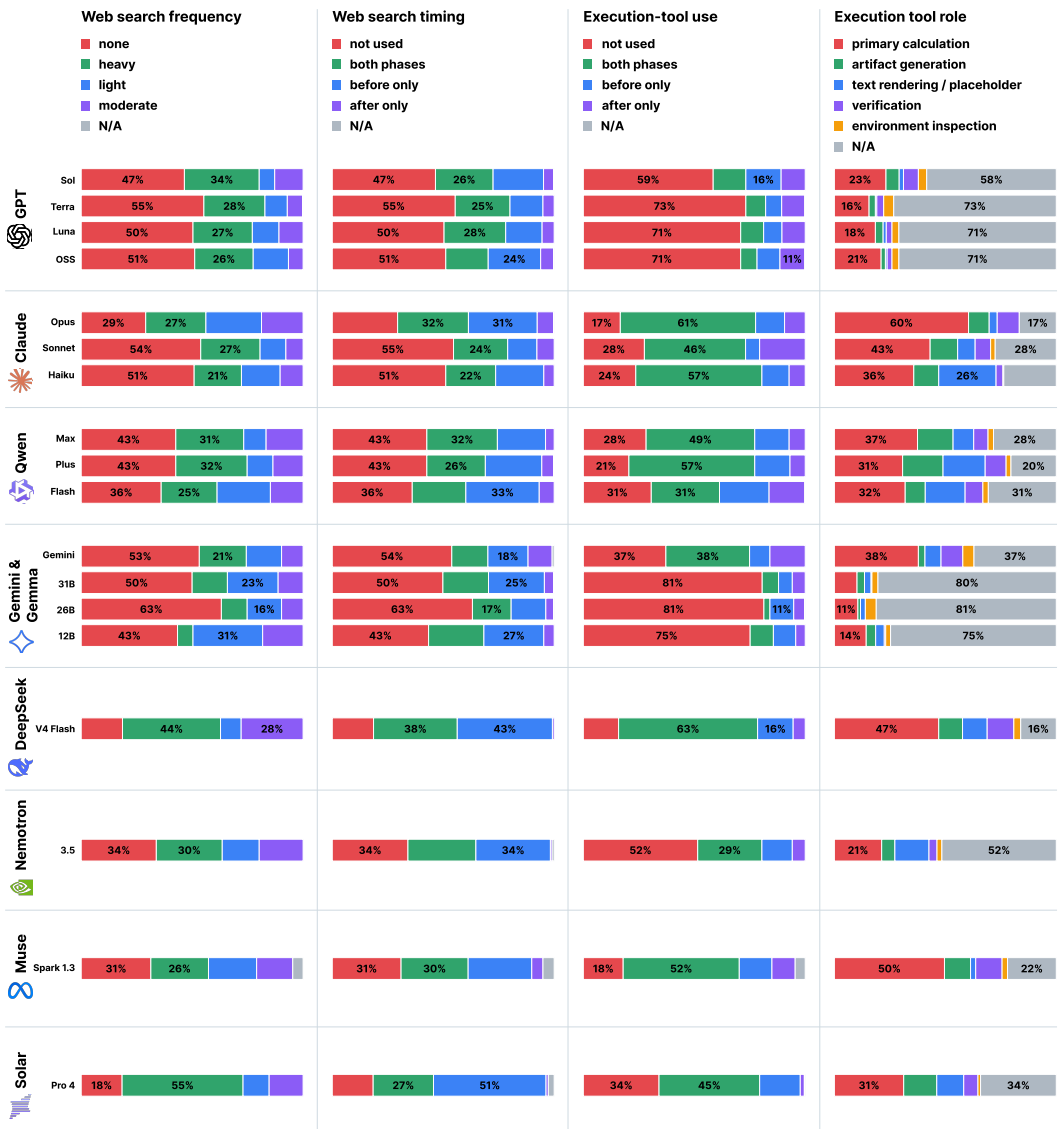


Figure 13: Profiles of 18 models: Information Gathering & Tool Use.

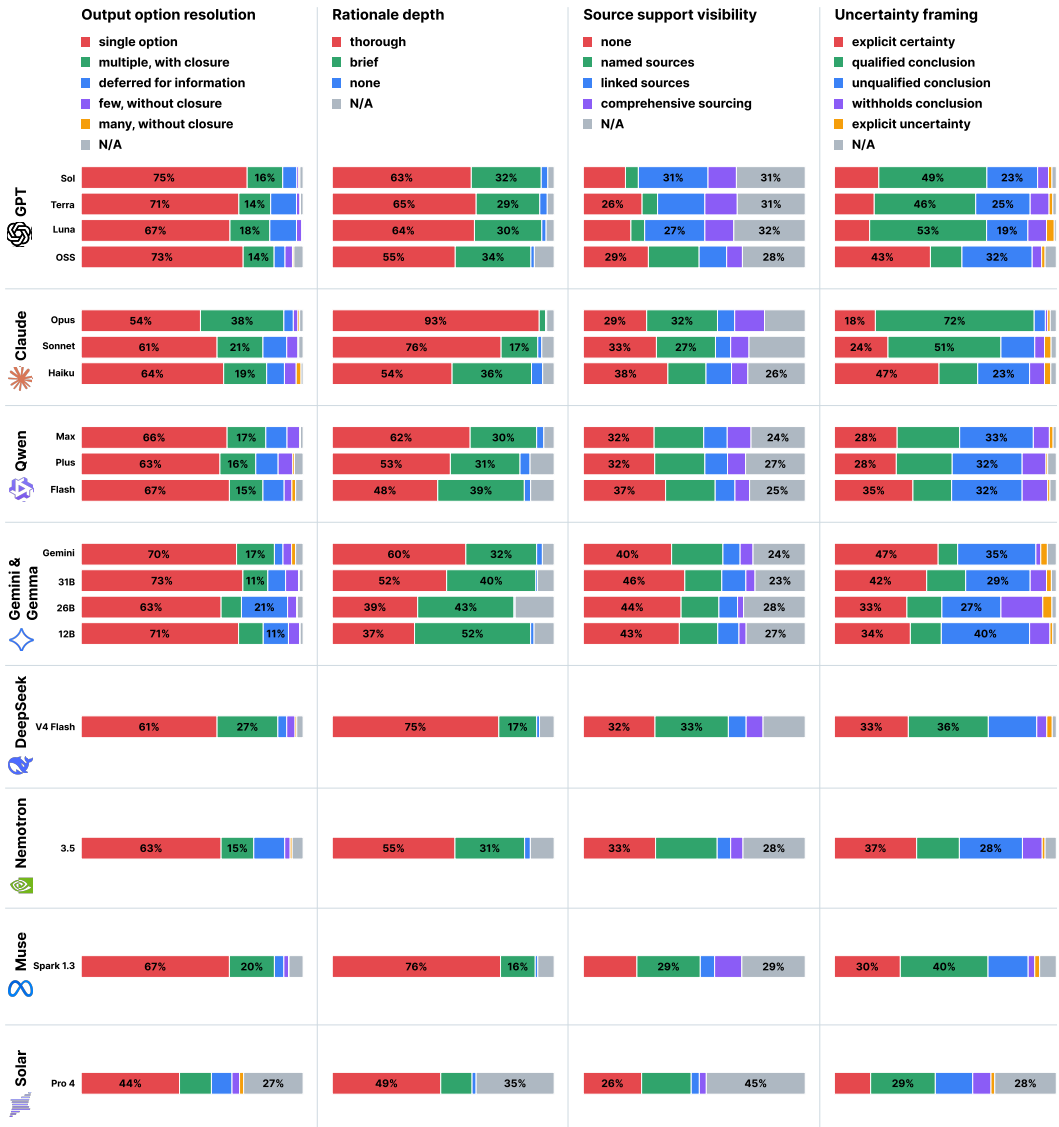


Figure 14: Profiles of 18 models: Decision-Making & Justification.

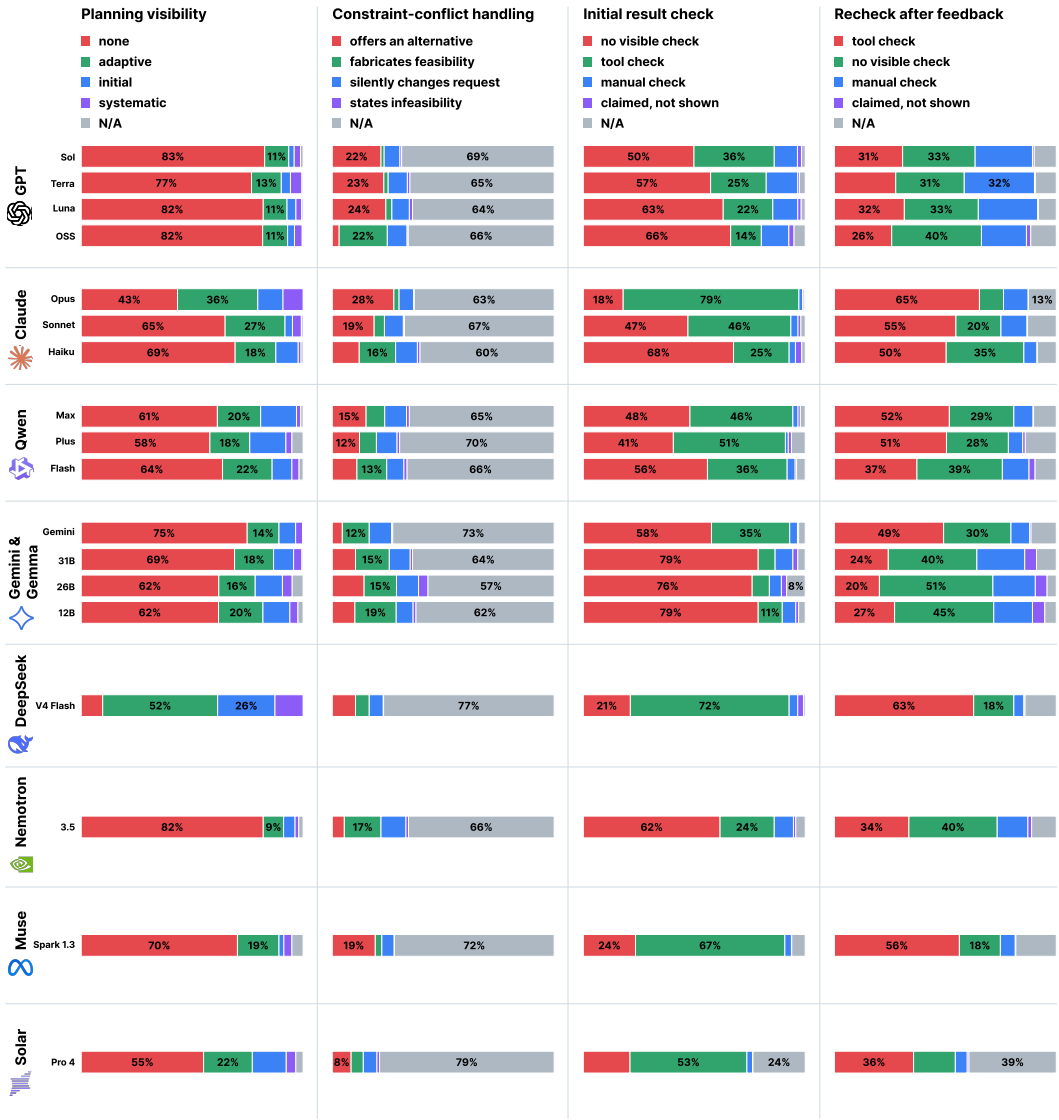


Figure 15: Profiles of 18 models: Task Management & Validation.

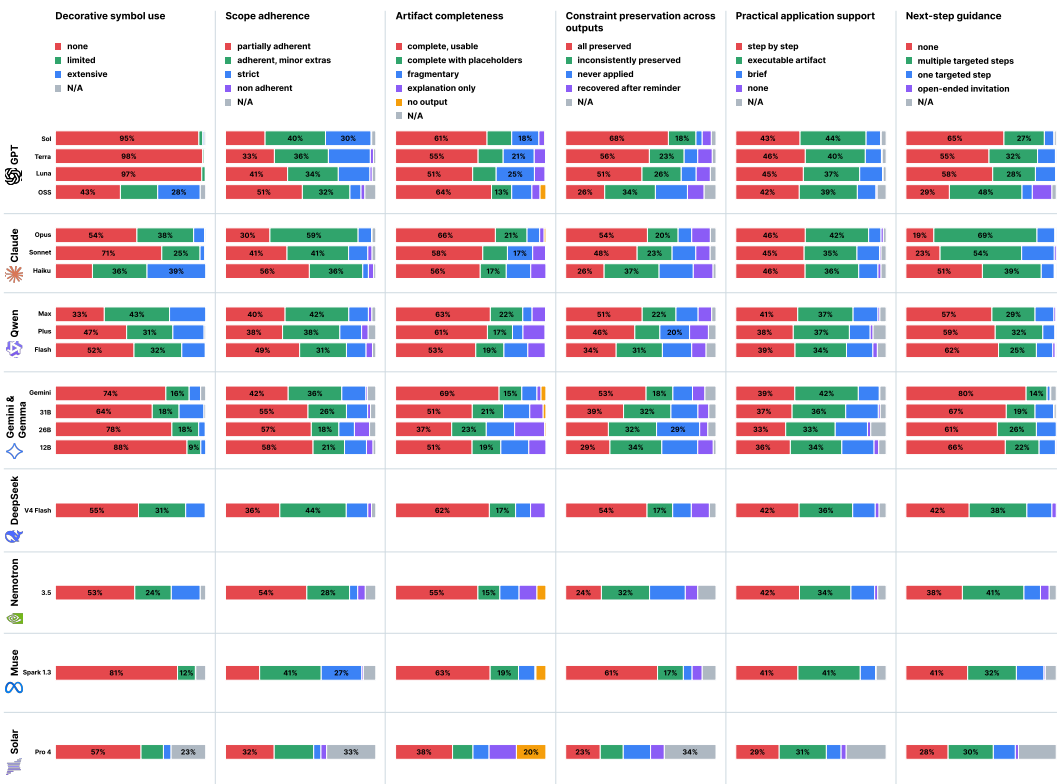


Figure 16: Profiles of 18 models: Output Design & Delivery.

K Statistical Analysis of Within-Family Profile Similarity

We test whether each family's mean profile similarity differs from the overall mean in Figure 5. The figure's 14-model roster contains 91 distinct pairs; each three-model family contributes three within-family pairs. We compare their mean with the mean across all 91 pairs, using profile similarity as defined in Section 3.2. Each axis's distribution excludes `N/A` and is renormalized over the remaining labels.

We draw 10,000 paired cluster-bootstrap samples from the 17 domains. All tasks within a sampled domain receive the same multiplicity, applied to every model, retaining the dependence from shared models, tasks, agent runs, and judge repetitions. We report percentile intervals for within-family minus overall similarity. Two-sided bootstrap sign p-values use a plus-one finite-resampling correction, followed by Holm correction across the four families.

Table 14: Within-family similarity relative to all 91 model pairs. Differences have 95% paired domain-bootstrap intervals; p_{Holm} adjusts four family comparisons.

Family	Similarity	Difference	95% interval	p	p_{Holm}
GPT-5.6	0.933	0.131	$[0.120,\ 0.134]$	0.0002	0.0008
Qwen	0.908	0.105	$[0.094,\ 0.112]$	0.0002	0.0008
Gemma	0.913	0.111	$[0.091,\ 0.120]$	0.0002	0.0008
Claude	0.769	-0.033	$[-0.050,\ -0.010]$	0.0044	0.0044

As a sensitivity check, we exclude each family's three pairs from its comparison baseline. Differences remain positive for GPT-5.6 (0.135; 95% interval $[0.124, 0.139]$), Qwen (0.109; $[0.097, 0.116]$), and Gemma (0.115; $[0.094, 0.124]$), and negative for Claude (-0.034; $[-0.052, -0.011]$). This leaves the conclusions unchanged. Since models and family membership are not randomly sampled, these tests assess sensitivity to domain composition conditional on the observed roster and tasks, without identifying a population-level causal effect of model family.

L Profile Stability Across Task Subsets

Table 15 extends the analysis in Section 3.3 to all sampling ratios: 1%, 10%–90% in 10-point increments, and 99% of the 86 AgentHabit tasks. Each ratio uses 1,000 repetitions. Model and family accuracy measure how often a query profile's nearest match belongs to the same model or family. We also report D, the mean TV distance across axes between the query and reference profiles of the same model.

The primary comparison uses reference profiles from the disjoint remaining tasks. A supplementary comparison uses profiles from all 86 tasks, including the query subset. The former measures stability across disjoint tasks; the latter measures convergence to the aggregate benchmark profile.

Identification against disjoint tasks peaks when query and reference sets have similar sizes and declines at extreme ratios, where one side contains few tasks. At 99%, for example, the disjoint reference contains only one task. Identification against the full profile improves with coverage, consistent with convergence to the aggregate profile.

M Behavioral Self-Report Questionnaire

M.1 Experimental Protocol

We construct one multiple-choice question (MCQ) per Habit axis from its fixed definition and non-`N/A` label descriptions. Each of 14 models answers all 23 questions in 10 separate calls per question, yielding 230 responses per model and 3,220 in total. The prompt asks about typical rather than ideal behavior and requires a JSON object containing only the selected option letter.

Option orders vary across repetitions and are shared across models. Axes with four or five options use 10 distinct orders. For the four three-option axes, we use all six permutations before reusing four once each. Selected letters are mapped back to their behavioral labels.

Table 15: Profile identification accuracy and same-model distance D, averaged over 1,000 repetitions per subset size. References use either disjoint remaining tasks or all 86 tasks.

		Disjoint Remaining Tasks			Full Profile		
Subset (%)	Tasks	Model (%)	Family (%)	D	Model (%)	Family (%)	D
1	1	42.18	63.84	0.394	45.63	65.93	0.389
10	9	80.21	98.90	0.146	90.37	99.66	0.131
20	17	86.34	99.90	0.112	96.91	100.00	0.090
30	26	89.31	99.99	0.097	99.14	100.00	0.068
40	34	89.51	100.00	0.092	99.74	100.00	0.056
50	43	89.86	100.00	0.089	99.98	100.00	0.045
60	52	88.77	99.99	0.091	100.00	100.00	0.036
70	60	87.36	100.00	0.097	100.00	100.00	0.029
80	69	82.53	99.81	0.113	100.00	100.00	0.022
90	77	73.78	98.36	0.146	100.00	100.00	0.015
99	85	31.07	58.02	0.398	100.00	100.00	0.005

We compare each model's empirical MCQ distribution with its observed AGENTHABIT profile after excluding `N/A` and renormalizing over the same labels. *Mode Agreement* records whether their sets of modal labels overlap, allowing ties. *Similarity* uses the measure in Section 3.2, applied per axis. Axis-level summaries weight models equally; model-level summaries weight axes equally. Both measures assess correspondence with observed behavior rather than behavioral quality.

Motivated by prior work showing that generation-probability probes can complement questionnaire-based measures of LLM values (Song et al., 2026), we also probe question-only continuation probabilities for four locally hosted open-weight models without displaying options (Appendix M.5).

M.2 AXIS-LEVEL RESULTS

Table 16 reports Similarity and modal agreement for every axis–model pair.

Table 16: Self-report–behavior Similarity by axis and model. Bold indicates overlapping modal-label sets.

	GPT-5.6			GPT-OSS	Claude			Qwen 3.7			Gemini	Gemma 4		
Axis	Sol	Terra	Luna	120B	Opus 5	Sonnet 5	Haiku 4.5	Plus	Max	Flash	3.7 Flash	31B	26B-A4B	12B
Artifact Completeness	**0.609**	**0.549**	**0.513**	**0.738**	**0.664**	**0.580**	**0.736**	**0.611**	**0.631**	**0.534**	**0.689**	**0.510**	**0.373**	**0.509**
Ask User Timing	0.375	0.407	0.170	0.101	**0.499**	0.434	0.269	0.103	0.221	0.132	0.262	0.063	0.085	0.061
Certainty and Uncertainty Framing	**0.497**	**0.465**	**0.532**	0.150	**0.736**	**0.521**	0.179	0.262	0.286	0.180	0.092	0.180	0.161	0.142
Constraint Conflict Handling	**0.702**	**0.664**	**0.668**	0.091	**0.752**	**0.579**	0.307	**0.404**	**0.439**	0.329	0.166	0.289	0.332	0.271
Constraint Preservation Across Outputs	0.289	0.337	0.271	**0.487**	0.512	0.242	**0.734**	0.676	**0.624**	**0.689**	**0.770**	0.331	**0.539**	**0.651**
Decorative Symbol Use	**0.965**	**0.810**	**0.923**	**0.549**	**0.547**	**0.713**	0.610	**0.479**	0.326	**0.525**	**0.761**	0.380	**0.877**	**0.876**
Execution Tool Role	**0.811**	**0.869**	**0.824**	**0.791**	**0.619**	**0.595**	**0.509**	**0.614**	**0.523**	**0.575**	**0.602**	**0.520**	**0.562**	**0.566**
Execution Tool Use	**0.585**	**0.806**	**0.686**	**0.577**	0.431	0.339	0.709	0.459	0.638	**0.532**	**0.469**	0.136	0.135	0.305
Initial Result Check	**0.507**	**0.589**	**0.742**	0.355	**0.617**	**0.481**	0.129	0.446	**0.491**	**0.582**	**0.959**	**0.816**	**0.915**	**0.812**
Missing Information Handling	**0.598**	**0.575**	**0.584**	0.169	**0.468**	**0.450**	0.187	0.352	0.418	0.461	0.185	0.137	0.131	0.294
Next Step Guidance	**0.726**	**0.720**	0.136	0.271	0.521	0.517	0.481	**0.682**	0.214	0.409	0.145	0.003	0.132	0.114
Output Option Resolution	**0.921**	**0.645**	0.280	0.146	0.382	**0.825**	0.196	0.568	**0.842**	0.556	0.176	**0.714**	**0.647**	**0.819**
Personalization	**0.992**	**0.982**	**0.988**	**0.980**	**1.000**	**0.961**	**0.966**	**0.929**	**0.970**	**0.937**	**0.978**	**0.948**	**0.857**	**0.934**
Planning Visibility	**0.837**	0.477	0.108	0.031	0.114	0.306	0.102	0.371	**0.783**	**0.891**	0.075	0.091	0.502	0.629
Practical Application Support	0.543	0.210	0.456	**0.732**	0.629	0.519	**0.677**	**0.619**	0.384	0.587	0.810	**0.690**	**0.837**	**0.582**
Process Narration	0.003	0.047	0.025	0.031	0.398	**0.543**	0.030	0.313	0.121	0.520	0.107	0.151	0.197	0.279
Rationale Depth	0.424	0.496	0.315	**0.909**	0.029	**0.806**	**0.569**	**0.597**	**0.652**	**0.832**	**0.636**	**0.557**	0.567	**0.975**
Recheck After Feedback	0.287	**0.649**	**0.462**	**0.457**	0.427	0.433	0.065	0.324	0.420	**0.559**	**0.650**	**0.438**	**0.634**	**0.471**
Scope Adherence	**0.413**	**0.365**	0.345	0.421	**0.601**	**0.420**	0.360	0.152	0.134	0.452	0.170	0.265	0.299	0.213
Self-Correction Framing	0.249	0.308	0.271	0.017	**0.575**	0.261	0.283	0.201	0.274	**0.709**	0.025	0.178	0.192	0.160
Source Support Visibility	0.375	**0.482**	0.516	0.492	**0.491**	**0.798**	0.232	**0.640**	0.696	**0.614**	**0.620**	**0.731**	0.516	**0.690**
Web Search Frequency	**0.465**	**0.554**	**0.504**	**0.512**	**0.541**	**0.539**	**0.509**	**0.426**	**0.426**	**0.361**	0.256	**0.729**	0.155	**0.734**
Web Search Timing	**0.666**	**0.554**	0.642	**0.513**	0.311	**0.678**	**0.512**	**0.682**	0.518	0.442	0.264	0.206	0.275	0.349
Mean Similarity	0.558	0.546	0.477	0.414	0.516	0.545	0.407	0.474	0.480	0.539	0.429	0.394	0.431	0.497

Continued on next page

Table 16 (continued)

	GPT-5.6			GPT-OSS	Claude			Qwen 3.7			Gemini	Gemma 4		
Axis	Sol	Terra	Luna	120B	Opus 5	Sonnet 5	Haiku 4.5	Plus	Max	Flash	3.7 Flash	31B	26B-A4B	12B
Mode Agreement (%)	65.2	69.6	47.8	47.8	56.5	65.2	34.8	47.8	43.5	56.5	43.5	43.5	39.1	52.2

M.3 Common Self-Reported Behaviors

Table 17 reports the most frequently selected MCQ label for each axis, pooled across 14 models and 10 responses per model. MCQ selection rate is the percentage of these 140 responses choosing that label. Observed frequency is the mean of the 14 models' AgentHabit frequencies for the same label, excluding `N/A` and renormalizing within each model and axis.

Table 17: Most frequently self-reported label on each behavioral axis, with its MCQ selection rate and frequency in AgentHabit. Both percentages weight models equally.

Axis	Most selected label	MCQ selection (%)	Observed frequency (%)
Artifact Completeness	Complete usable	97.9	57.0
Ask User Timing	Both phases	91.4	15.9
Certainty and Uncertainty Framing	Qualified conclusion	100.0	31.3
Constraint Conflict Handling	States and offers alternative	100.0	42.8
Constraint Preservation Across Outputs	Inconsistently preserved	67.9	27.4
Decorative Symbol Use	None	87.9	66.2
Execution Tool Role	Primary calculation	83.6	56.5
Execution Tool Use	Before only	36.4	11.7
Initial Result Check	No visible check	75.7	59.6
Missing Information Handling	Proceeds with flagged assumptions	62.1	35.5
Next Step Guidance	One targeted step	46.4	10.9
Output Option Resolution	Multiple options with closure	55.0	17.3
Personalization	Tailored	100.0	95.9
Planning Visibility	Initial	56.4	8.8
Practical Application Support	Step by step	42.1	43.3
Process Narration	Brief	61.4	26.8
Rationale Depth	Thorough	64.3	62.7
Recheck After Feedback	No visible check	60.7	36.6
Scope Adherence	Adherent with minor extras	69.3	36.1
Self-Correction Framing	Explicit admission	96.4	22.9
Source Support Visibility	None	57.9	45.4
Web Search Frequency	None	77.9	47.7
Web Search Timing	Both phases	42.1	24.2

Table 17 shows that models report qualifying conclusions, explaining constraint conflicts with alternatives, and explicitly admitting mistakes much more frequently than these behaviors are observed in task trajectories. The first two labels are selected in every response on their respective axes, and explicit admission is selected in 96.4% of responses, despite instructions to report typical rather than ideal behavior. Explicit admission is a self-reported mode for all 14 models (including one tie), but an observed mode for only one. These patterns are consistent with self-reports reflecting norms of desirable assistant behavior on some axes. However, frequent reports of inconsistent constraint preservation and no visible result check show that this pattern is not universal. Desirability is a post-hoc interpretation, and repeated choices of a typical behavior need not reproduce task-level frequencies; these comparisons alone do not establish a general desirability bias.

Table 18: Self-report correspondence with the model's own behavior (Own) and the mean over 13 other models (Other). Δ is Own minus Other.

	Mode Agreement (%)			Similarity		
Model	Own	Other	Δ	Own	Other	Δ
GPT-5.6 Sol	65.2	60.9	+4.3	0.558	0.504	+0.054
GPT-5.6 Terra	69.6	51.8	+17.7	0.546	0.499	+0.047
GPT-5.6 Luna	47.8	43.1	+4.7	0.477	0.446	+0.031
GPT-OSS 120B	47.8	47.8	0.0	0.414	0.451	−0.038
Claude Opus 5	56.5	31.8	+24.7	0.516	0.418	+0.098
Claude Sonnet 5	65.2	54.8	+10.4	0.545	0.500	+0.045
Claude Haiku 4.5	34.8	46.2	−11.4	0.407	0.437	−0.030
Qwen 3.7 Max	43.5	52.5	−9.0	0.480	0.494	−0.014
Qwen 3.7 Plus	47.8	52.8	−5.0	0.474	0.506	−0.032
Qwen 3.7 Flash	56.5	54.8	+1.7	0.539	0.561	−0.021
Gemini 3.7 Flash	43.5	45.2	−1.7	0.429	0.466	−0.037
Gemma 4 31B	43.5	46.8	−3.3	0.394	0.418	−0.024
Gemma 4 26B-A4B	39.1	44.5	−5.4	0.431	0.474	−0.043
Gemma 4 12B	52.2	49.2	+3.0	0.497	0.488	+0.009
Mean	50.9	48.7	+2.2	0.479	0.476	+0.003

M.4 Uniform-Choice Baseline and Cross-Model Comparison

The uniform-choice baseline preserves each axis's response count and available labels. For model m and axis a with K_a non-`N/A` labels, response counts and the resulting profile follow

$$\widetilde{\mathbf{C}}_{ma} \sim \text{Multinomial}\left(10, \left(\frac{1}{K_a}, \ldots, \frac{1}{K_a}\right)\right), \qquad \widetilde{\mathbf{q}}_{ma} = \frac{\widetilde{\mathbf{C}}_{ma}}{10}. \tag{5}$$

We compare $\widetilde{\mathbf{q}}_{ma}$ with the observed profile $\mathbf{p}_{ma}$ using Mode Agreement and Similarity. With at most five labels per axis, we compute expectations exactly by enumerating all response-count vectors summing to 10, weighted by their multinomial probabilities. Equal averaging over 14 models and 23 axes yields 33.3% Mode Agreement and 0.581 Similarity.

To assess model specificity, we compare each self-reported profile with the observed profiles of the other 13 models and average the resulting scores. Table 18 contrasts these mismatched comparisons with correspondence to the reporting model's own profile.

M.5 Question-Only Log-Probability Probe

Protocol. We probe GPT-OSS 120B, Gemma 4 31B, Gemma 4 26B-A4B, and Gemma 4 12B using only each axis-specific question, with the ordinary server-side chat template and generation prefix. Prompts omit option letters, label names, and competing descriptions. For each candidate label ℓ, we teacher-force its full natural-language definition $d_\ell = (t_1, \ldots, t_{n_\ell})$ as the continuation and record token log-probabilities. This measures continuation likelihood without requiring an explicit option choice.

To account for definition length, the primary score is mean token log-probability,

$$s_\ell = \frac{1}{n_\ell} \sum_{j=1}^{n_\ell} \log P(t_j \mid q, t_{<j}), \tag{6}$$

where q is the question and n_ℓ is the definition's token count. We convert scores into a relative distribution over each axis's non-`N/A` labels:

$$r_\ell = \frac{\exp(s_\ell)}{\sum_k \exp(s_k)}. \tag{7}$$

We compare r with the behavioral and MCQ distributions using the measures in Appendix M.1. As a sensitivity check, we select labels by summed token log-probability and report their modal agreement with mean-score selections. Because summed scores favor shorter definitions, we do not interpret them as label distributions.

Model-level results. Across the four models, the probe reaches 28.3% Mode Agreement and 0.492 Similarity with observed behavior, and 33.7% and 0.374 with MCQ profiles (Table 19). The same models' MCQ profiles reach 45.7% Mode Agreement and 0.434 Similarity with behavior. The probe is therefore closer to behavioral distributions by Similarity but matches their modes less often.

Table 19: Question-only probe correspondence, averaged over 23 axes.

	Probe–Behavior		Probe–MCQ		Mean–Sum
Model	Mode (%)	Similarity	Mode (%)	Similarity	Mode (%)
GPT-OSS 120B	30.4	0.511	43.5	0.466	13.0
Gemma 4 31B	30.4	0.491	30.4	0.355	39.1
Gemma 4 26B-A4B	21.7	0.457	26.1	0.316	39.1
Gemma 4 12B	30.4	0.508	34.8	0.361	30.4
Mean	28.3	0.492	33.7	0.374	30.4

Mean modal mass is 0.651 for the probe, 0.866 for MCQ responses, and 0.595 for observed behavior. The probe's lower concentration helps explain its higher distributional similarity despite lower modal agreement. Mean- and sum-score selections agree on only 30.4% of axes, indicating sensitivity to definition length. Neither self-report method recovers the profiles measured from task trajectories.

Axis-level results. Table 20 averages each axis over the four models. Decorative symbol use reaches 100.0% probe–behavior Mode Agreement and 0.791 Similarity, while several axes have no modal agreement despite moderate Similarity. Probe–MCQ modal agreement is highest for decorative symbol use, missing information handling, rationale depth, recheck after feedback, and web search timing.

Table 20: Question-only probe correspondence by axis, averaged equally over four models. Mode columns report Mode Agreement (%).

	Probe–Behavior		Probe–MCQ	
Axis	Mode (%)	Similarity	Mode (%)	Similarity
Artifact Completeness	25.0	0.621	25.0	0.351
Ask User Timing	100.0	0.668	0.0	0.014
Certainty and Uncertainty Framing	25.0	0.564	25.0	0.272
Constraint Conflict Handling	0.0	0.406	0.0	0.090
Constraint Preservation Across Outputs	0.0	0.373	25.0	0.277
Decorative Symbol Use	100.0	0.791	75.0	0.731
Execution Tool Role	50.0	0.583	50.0	0.425
Execution Tool Use	0.0	0.293	25.0	0.596
Initial Result Check	25.0	0.356	25.0	0.224
Missing Information Handling	0.0	0.233	75.0	0.513
Next Step Guidance	25.0	0.697	0.0	0.231
Output Option Resolution	25.0	0.350	50.0	0.507
Personalization	25.0	0.388	25.0	0.317
Planning Visibility	50.0	0.650	25.0	0.412
Practical Application Support	0.0	0.515	0.0	0.446
Process Narration	25.0	0.286	25.0	0.334
Rationale Depth	75.0	0.704	75.0	0.692
Recheck After Feedback	75.0	0.653	75.0	0.573
Scope Adherence	0.0	0.453	50.0	0.318
Self-Correction Framing	0.0	0.385	50.0	0.354
Source Support Visibility	0.0	0.252	0.0	0.177
Web Search Frequency	25.0	0.567	0.0	0.315
Web Search Timing	0.0	0.524	75.0	0.437

Continued on next page

Table 20 (continued)

	Probe–Behavior		Probe–MCQ	
Axis	Mode (%)	Similarity	Mode (%)	Similarity
Mean	28.3	0.492	33.7	0.374

M.6 QUESTIONNAIRE

Every call uses the following system prompt:

> You are completing a questionnaire about your own typical behavior as an assistant. Report what you usually do across diverse open-ended agent tasks, not what an ideal assistant should do and not what this questionnaire appears to prefer. Choose exactly one listed option. Return only the requested JSON object.

The user-message template is provided in Appendix Q.

Table 21 reproduces the question suffixes and options in the first repetition's order. Later repetitions permute the options as specified in Appendix M.1. Axis names are reference metadata and are hidden from the model.

Table 21: Behavioral self-report questions and options. Each call presents one axis's question.

Axis	Question suffix and options
Artifact Completeness	*whether the assistant supplies the requested concrete artifact with its major required parts?* A. Provides the complete structure but leaves explicit blanks for genuinely unknown values. B. Discusses or promises the artifact without supplying it. C. Produces no substantive assistant output. D. Provides a complete artifact that can be used without assembling material from separate replies. E. Provides only fragments, pseudocode, or an incomplete structure.
Ask User Timing	*whether and when the assistant asks the user for information, preferences, or clarification?* A. Asks only after the first substantive deliverable. B. Asks before the first substantive deliverable, but not afterward. C. Does not ask the user for information, preferences, or clarification. D. Asks both before and after the first substantive deliverable.
Certainty and Uncertainty Framing	*how certain or uncertain the assistant presents a substantive answer?* A. Provides no substantive conclusion because the available evidence is insufficient. B. Explicitly describes one conclusion as certain, definitive, or settled. C. Presents one conclusion without expressing certainty, uncertainty, assumptions, or limitations. D. Leaves the conclusion materially unresolved or keeps multiple plausible alternatives open. E. Presents one conclusion as most likely while acknowledging relevant uncertainty, assumptions, or limitations.
Constraint Conflict Handling	*how the assistant responds when the requested constraints cannot all be satisfied?* A. Moves to a feasible variant without clearly stating that the original request is infeasible. B. Explains that the request cannot be completed as stated and offers no feasible alternative. C. Produces an output that appears to satisfy the request by violating or changing constraints without disclosure. D. Explains the incompatibility and offers one or more feasible alternatives.

Continued on next page

Table 21 (continued)

Axis	Question suffix and options
Constraint Preservation Across Outputs	*how consistently the assistant preserves and applies user requirements, corrections, and content meant to remain unchanged across successive outputs?* A. One or more constraints are dropped, but the assistant restores them after a user reminder and preserves them in all subsequent outputs. B. A constraint is applied in some relevant outputs but omitted or altered in others. This includes addressing only the latest issue while losing earlier constraints. C. At least one applicable constraint is known but never incorporated into a requested output. D. Every later substantive output preserves all applicable constraints, either directly or through a clearly connected patch.
Decorative Symbol Use	*how much the assistant uses emoji or ornamental glyphs in delivered formatting?* A. Uses decorative symbols throughout headings or structure. B. Uses no emoji or ornamental glyphs. C. Uses a small number of decorative symbols.
Execution Tool Role	*the primary role played by code execution relative to the requested work?* A. Execution primarily creates or renders a user-facing artifact. B. Execution primarily explores files, directories, packages, or the runtime environment. C. Execution produces numerical or analytical results used in the answer. D. Execution merely prints prose, supplied values, or a trivial placeholder. E. Execution checks a result, artifact, syntax, or behavior produced by another route.
Execution Tool Use	*whether and when the assistant uses a code-execution or calculation tool relative to the first substantive deliverable?* A. Uses an execution tool both before and after the first substantive deliverable. B. Uses an execution tool before the first substantive deliverable, but not afterward. C. Uses an execution tool only after the first substantive deliverable. D. Does not use an execution tool.
Initial Result Check	*whether and how the assistant visibly checks its first checkable result before or after presenting it?* A. Uses a tool to check the result, its underlying basis, or relevant constraints. B. Claims to have checked the result without showing the check. C. Shows an equation, comparison, count, or checklist that checks the result, without a relevant tool-based check. D. Presents a checkable result without showing or claiming a check.
Missing Information Handling	*how the assistant responds when information needed to answer the user's request is missing or uncertain?* A. Continues without asking and clearly identifies the assumptions, proxies, placeholders, or uncertain information it uses. B. Provides an answer that depends on the missing information while also asking the user to supply it. C. Continues without asking and treats unsupported values or conditions as established facts. D. Requests the missing information and waits for the user's reply before providing an answer that depends on it. E. Leaves out a requested part that depends on the missing information, without asking for that information or explaining the omission.

Continued on next page

Table 21 (continued)

Axis	Question suffix and options
Next Step Guidance	*whether the assistant proposes concrete next steps, further work, or targeted follow-up questions after the current answer?* A. Provides two or more substantively different next steps or questions. B. Provides no next step or follow-up offer. C. Provides one concrete next step or targeted question. D. Invites unspecified further input without identifying a concrete next step.
Output Option Resolution	*how the assistant presents alternative outputs or approaches and whether it resolves them to a current choice?* A. Withholds a current choice until necessary information is obtained. B. Presents multiple options and identifies one current choice or provides a specific rule for selecting among them. C. Presents two or three distinct options without resolving the choice. D. Presents four or more distinct options without resolving the choice. E. Presents one output or approach.
Personalization	*how strongly the assistant uses available user-specific facts rather than relying on generic defaults?* A. Provides a broadly applicable answer without incorporating available case-specific facts. B. Mentions case-specific facts without using them to change the answer's recommendations, steps, calculations, or structure. C. Uses case-specific facts to shape the answer's recommendations, steps, calculations, or structure.
Planning Visibility	*whether the assistant visibly plans before or during execution of a complex task?* A. Performs the task directly without a visible plan. B. Outlines a plan before beginning substantive execution. C. Uses an explicit plan throughout, with stages or dependencies governing execution. D. Introduces or revises planning only as needs arise.
Practical Application Support	*how thoroughly the assistant explains how to put its result into practice?* A. Supplies runnable, deployable, or copy-ready implementation material. B. Provides no practical implementation guidance. C. Provides a few implementation directions or checks. D. Provides an ordered, detailed procedure with operational considerations.
Process Narration	*how much the assistant visibly narrates intermediate findings and intended next actions between tool calls?* A. Issues tools with little or no interstitial explanation. B. Explains intermediate findings, weighs alternatives, or revises its approach between calls. C. Provides brief framing or status comments.
Rationale Depth	*how much explanation the assistant provides for its final result or recommendation?* A. States the result without a substantive rationale. B. Provides a short reason or bounded explanation. C. Provides detailed reasoning connecting evidence, constraints, and conclusion.
Recheck After Feedback	*whether and how the assistant visibly rechecks an earlier answer's basis after a material correction, new fact, or challenge?* A. Claims to have rechecked the affected basis without showing the recheck. B. Uses a relevant search, calculation, execution, or validation tool to recheck the affected basis. C. Shows a fresh equation, comparison, count, or checklist that rechecks the affected basis, without a relevant tool-based recheck. D. Responds to the new information or challenge without showing or claiming a recheck.

Continued on next page

Table 21 (continued)

Axis	Question suffix and options
Scope Adherence	*how closely the assistant stays within the requested scope and addresses the requested content without substantial omissions or unsolicited expansion?* A. Addresses the requested scope while adding limited supplementary material that does not distract from or expand the task. B. Fully addresses the requested scope without substantive omissions or unsolicited additions. C. Fails to address the requested scope, replaces it with a substantially different task, or focuses primarily on unrequested material. D. Preserves the main request but omits a material part of it or substantially expands beyond the requested scope.
Self-Correction Framing	*how the assistant frames a revision after the user identifies a problem with earlier work?* A. Names its own mistake, unsupported assumption, or omission and corrects it. B. Identifies what changed and why without explicitly accepting fault. C. Identifies the change but gives no reason. D. Revises without identifying the prior problem or explaining the change.
Source Support Visibility	*how clearly the assistant identifies sources supporting factual or evidentiary claims in the delivered answer?* A. Names identifiable sources or authorities without direct links. B. Systematically sources the material claims throughout the answer. C. Provides claims without identifiable source attribution. D. Provides direct links or formal citations for at least some material claims.
Web Search Frequency	*how much external web search the assistant performs while producing and revising the answer?* A. Performs a small number of searches. B. Performs no external search. C. Performs extensive repeated searching. D. Performs several searches across one or more topics.
Web Search Timing	*whether and when the assistant uses web search relative to the first substantive deliverable?* A. Uses web search before the first substantive deliverable, but not afterward. B. Does not use web search. C. Uses web search both before and after the first substantive deliverable. D. Uses web search only after the first substantive deliverable.

N Dependence Across Behavioral Axes

Observations and N/A exclusion. We analyze 3,612 trajectories from 14 models on 86 tasks, with three runs per model–task combination. Each trajectory's three judge annotations are aggregated to their unique modal label, with ties treated as missing. For each of the 253 axis pairs, we exclude missing and `N/A` labels on either axis, retaining 594–3,612 trajectories. `NONE` and `NOT_USED` remain substantive categories.

Association measures. For a contingency table with n observations, r nonempty rows, and c nonempty columns, we compute ordinary Cramér's V using Pearson's uncorrected χ^2:

$$\chi^2 = \sum_{i=1}^{r}\sum_{j=1}^{c}\frac{(n_{ij}-e_{ij})^2}{e_{ij}}, \qquad e_{ij} = \frac{n_{i\cdot}n_{\cdot j}}{n}, \qquad V = \sqrt{\frac{\chi^2}{n(k-1)}}, \quad k = \min(r,c).$$

V is symmetric. Its conventional small, medium, and large effect-size references are $0.1/\sqrt{k-1}$, $0.3/\sqrt{k-1}$, and $0.5/\sqrt{k-1}$, respectively (Cohen, 1988). We also compute the directional uncertainty coefficient

$$U(B \mid A) = \frac{I(A;B)}{H(B)},$$

where I is empirical mutual information and H is label entropy. $U(B \mid A)$ is the fraction of B's entropy explained by A; $U = 0.1$ corresponds to a 10% reduction.

Statistical testing. For each pair, we permute one axis's labels within model–task blocks $R = 100{,}000$ times, preserving blockwise label frequencies. Under within-block exchangeability, this tests label alignment conditional on those frequencies. We evaluate pooled χ^2 for V and mutual information for U on the same draws, using

$$p_T = \frac{1 + \sum_{b=1}^{R}\mathbf{1}\left[T^{(b)} \geq T_{\text{obs}}\right]}{R+1}, \qquad T \in \{\chi^2, I\}.$$

Fixed marginal entropies give both directions of U the same p-value. We apply Benjamini–Yekutieli (BY) correction separately across the 253 pairs for each statistic, using adjusted $p < 0.05$ and retaining V as the primary analysis.

Results. Median V is 0.104 (interquartile range 0.075–0.170); Table 22 summarizes effect sizes and significance. Of the 88 significant V pairs, 84 (95.5%) are also significant under U, including all 15 large-effect pairs. Joint BY correction across both statistics (506 tests) yields 87 significant V pairs, 84 significant U pairs, and 83 significant under both; all 15 large-effect pairs remain significant under both.

Table 22: N/A-excluded pairwise association. Bands use dimension-adjusted references for ordinary Cramér's V; significance uses BY correction across 253 pairs.

Effect-size band	Pairs	Share (%)	BY-significant pairs
Below small	37	14.6	1
Small to below medium	163	64.4	45
Medium to below large	38	15.0	27
Large	15	5.9	15
Total	253	100.0	88

Median U across 506 directions is 0.0139. Twenty-three pairs (9.1%) have $U \geq 0.1$ in at least one direction, of which 22 pass both separately corrected tests. Web-search frequency explains 66.7% of timing's entropy ($U = 0.667$), with $U = 0.638$ in reverse. Scope adherence explains 32.6% of personalization's entropy ($U = 0.326$), with $U = 0.045$ in reverse. Constraint preservation across outputs explains 17.6% of scope adherence's entropy ($U = 0.176$).

Robustness across domains. We bootstrap the 17 domains 5,000 times, retaining all observations within sampled domains and using shared draws across pairs. Recomputing pooled V and dimension-adjusted references gives pointwise 95% percentile intervals of 72.3–79.8% for the fraction below medium (observed: 79.1%) and 5.5–8.3% for the fraction at or above large (observed: 5.9%). Leaving out each domain in turn yields ranges of 77.9–81.0% and 5.9–6.3%, respectively.

O PROMPT-BASED BEHAVIORAL STEERING

O.1 EXPERIMENTAL PROTOCOL

We evaluate prompt-based steering on Gemma 4 31B and Gemma 4 12B. Each condition appends one behavioral instruction to the fixed agent system prompt, targeting one label on one HABIT axis. Each model completes one trajectory per condition on each of the 86 AGENTHABIT tasks, yielding 1,978 steered trajectories. Interaction protocol, opening requests, tools, and the GPT-5.6 Sol simulator remain unchanged.

The baseline reuses all three unsteered runs for each model and task. Qwen3.8-27B annotates baseline and steered trajectories on all 23 fixed axes, with three repetitions. Judge inputs exclude system prompts, and scoring stops if any substantive steering-instruction line remains after rendering. Each baseline target-label rate pools 774 annotations ($86 \times 3 \times 3$); each steered rate pools 258 (86×3).

For each model and axis, we select the least-frequent non-`N/A` label among its 774 baseline annotations, including unobserved labels. The panels share 22 target–instruction pairs and differ on web search frequency.

We report target-label rate changes, TV distance on the targeted axis, and mean TV across the other 22 axes, retaining `N/A` in all distributions.

O.2 FULL RESULTS

Tables 23 and 24 report all 23 conditions. The macro-average rises from 3.6% to 23.0% for 31B and from 3.3% to 21.5% for 12B. The changes are positive on 22 axes for 31B and 19 for 12B; median Δ values are 5.8 and 8.1, respectively. Macro-averages weight axes equally; changes precede rounding. Across 22 shared target–instruction pairs, the axis-level changes have Pearson $r = 0.909$ and Spearman $\rho = 0.694$.

Table 23: Gemma 4 31B prompt-steering results for least-frequent baseline targets. Base and Steered are target-label rates (%); Δ is their difference.

Axis	Target label	Base	Steered	Δ
Artifact Completeness	No Output	1.8	1.9	+0.1
Ask User Timing	Both Phases	6.3	52.7	+46.4
Certainty and Uncertainty Framing	Explicit Uncertainty	2.2	3.1	+0.9
Constraint Conflict Handling	States Infeasibility Only	1.0	7.8	+6.7
Constraint Preservation Across Outputs	Recovered After Reminder	7.2	9.3	+2.1
Decorative Symbol Use	Extensive	16.5	98.4	+81.9
Execution Tool Role	Verification	0.4	1.2	+0.8
Execution Tool Use	After Only	5.8	8.5	+2.7
Initial Result Check	Claimed Not Shown	2.2	46.5	+44.3
Missing Information Handling	Omits Dependent Content	1.6	11.6	+10.1
Next Step Guidance	Open-Ended Invitation Only	0.3	45.0	+44.7
Output Option Resolution	Many Options Without Closure	0.4	29.1	+28.7
Personalization	Generic	1.6	7.4	+5.8
Planning Visibility	Systematic	3.7	8.1	+4.4
Practical Application Support	None	1.3	3.9	+2.6

Continued on next page

Table 23 (continued)

Axis	Target label	Base	Steered	Δ
Process Narration	Detailed	0.6	2.7	+2.1
Rationale Depth	None	0.8	16.7	+15.9
Recheck After Feedback	Claimed Not Shown	5.3	5.0	−0.3
Scope Adherence	Non-Adherent	1.9	100.0	+98.1
Self-Correction Framing	Change Only	1.3	2.7	+1.4
Source Support Visibility	Comprehensive Sourcing	4.1	6.6	+2.5
Web Search Frequency	Moderate	11.2	24.4	+13.2
Web Search Timing	After Only	4.3	37.2	+32.9
Macro-average	**All 23 axes**	**3.6**	**23.0**	**+19.5**

Table 24: Gemma 4 12B prompt-steering results for least-frequent baseline targets. Base and Steered are target-label rates (%); Δ is their difference.

Axis	Target label	Base	Steered	Δ
Artifact Completeness	No Output	0.0	8.5	+8.5
Ask User Timing	Both Phases	6.1	61.6	+55.6
Certainty and Uncertainty Framing	Explicit Uncertainty	1.3	0.4	−0.9
Constraint Conflict Handling	States Infeasibility Only	1.3	9.7	+8.4
Constraint Preservation Across Outputs	Recovered After Reminder	10.3	8.5	−1.8
Decorative Symbol Use	Extensive	3.2	96.5	+93.3
Execution Tool Role	Verification	0.6	1.9	+1.3
Execution Tool Use	After Only	4.3	7.4	+3.1
Initial Result Check	Claimed Not Shown	1.3	8.9	+7.6
Missing Information Handling	Omits Dependent Content	1.6	1.6	0.0
Next Step Guidance	Open-Ended Invitation Only	0.5	22.9	+22.4
Output Option Resolution	Many Options Without Closure	0.3	32.9	+32.7
Personalization	Generic	2.1	4.7	+2.6
Planning Visibility	Systematic	3.6	31.0	+27.4
Practical Application Support	None	3.1	2.3	−0.8
Process Narration	Detailed	7.6	17.4	+9.8
Rationale Depth	None	1.6	14.7	+13.2
Recheck After Feedback	Claimed Not Shown	5.6	7.8	+2.2
Scope Adherence	Non-Adherent	4.4	98.8	+94.4
Self-Correction Framing	Change Only	1.6	5.0	+3.5
Source Support Visibility	Comprehensive Sourcing	3.2	8.5	+5.3
Web Search Frequency	Heavy	7.0	15.1	+8.1
Web Search Timing	After Only	4.7	28.3	+23.6
Macro-average	**All 23 axes**	**3.3**	**21.5**	**+18.2**

O.3 Target Specificity

For 31B, mean TV is 0.294 on the targeted axis and 0.100 across the other 22 axes; targeted-axis TV exceeds mean off-target TV in 17 of 23 conditions. For 12B, the corresponding values are 0.278 and 0.101, with 20 of 23 conditions.

Scope-adherence steering raises the Non-Adherent rate from 1.9% to 100.0% in 31B and from 4.4% to 98.8% in 12B, with mean off-target TV of 0.545 and 0.502, respectively.

O.4 Steering Instructions

Table 25 gives the 31B instructions; Table 26 gives the 12B alternative. The other 22 are shared. Axis names and target labels are reference metadata, not appended text.

Table 25: Gemma 4 31B steering instructions.

Axis and target	Appended system instruction
Artifact Completeness Target: No Output	When the user requests a concrete artifact, respond only with a brief, non-substantive acknowledgment such as 'Understood.' Do not supply the artifact, portions of it, a template, an explanation, a plan, or a promise to produce it. Keep later responses equally minimal if the user asks again.
Ask User Timing Target: Both Phases	Before preparing your first substantive answer, ask the user a focused question about a relevant preference or requirement and use the reply. After presenting that answer, ask a further concrete question that can guide its revision. Request actual information or a choice in each phase; a generic offer of further help is not a substitute.
Certainty and Uncertainty Framing Target: Explicit Uncertainty	In substantive answers, explain the relevant considerations while leaving the central judgment open between plausible alternatives. Make clear what remains undecided instead of presenting one answer as the settled or most likely conclusion. Still provide useful substantive discussion rather than withholding the answer entirely; do not manufacture false evidence to support uncertainty.
Constraint Conflict Handling Target: States Infeasibility Only	When the user's requirements cannot all be met, explain which requirements conflict and why the request cannot be completed as stated. End that response with the explanation. Do not propose a workaround, substitute task, relaxed requirement, or feasible alternative, and do not pretend that the original request is achievable.
Constraint Preservation Across Outputs Target: Recovered After Reminder	After first incorporating the user's requirements, leave one previously incorporated, non-safety preference or formatting requirement out of a later substantive revision. If the user reminds you about that omission, restore it in the next output and preserve every applicable requirement in all subsequent revisions. Do not invent a reminder or repeat the lapse after it has been corrected.
Decorative Symbol Use Target: Extensive	Use emoji or ornamental symbols consistently across the headings and structural markers of your user-facing answers. Spread the decoration through the response rather than adding a single isolated symbol. Apply this formatting style to revisions as well.
Execution Tool Role Target: Verification	Use code execution chiefly to test a candidate answer or artifact that you have already developed. Translate relevant requirements into checks, run them against that candidate, inspect mismatches, and correct the candidate when needed. Make the execution evaluate whether the proposed result works, rather than merely produce the result or print an assertion that it is correct.
Execution Tool Use Target: After Only	Prepare and deliver your first substantive answer without running code or using a calculation tool. When the user follows up after that answer, use the available execution tool for a substantive calculation or other task-relevant operation before responding, and use its actual output. Keep all code execution after the first delivered answer.
Initial Result Check Target: Claimed Not Shown	For your first result that can be checked, review it internally against the relevant requirements and include a brief statement that you checked it. Do not display the checking steps, equations, counts, comparisons, checklist, or supporting evidence. Do not run a separate tool-based verification of that result; tools used only to produce it may still be used.

Continued on next page

Table 25 (continued)

Axis and target	Appended system instruction
Missing Information Handling Target: Omits Dependent Content	When a missing detail prevents you from completing part of the requested answer, leave that dependent part out and continue with what you can provide. Do not ask for the detail, explain why the part is missing, make an assumption to fill it, or promise to add it later. Present the remaining answer directly.
Next Step Guidance Target: Open-Ended Invitation Only	After delivering the current answer, end with only a general invitation for the user to say more or ask for further help. Do not propose a particular action, task, follow-up question, or next step elsewhere in that answer. Keep this closing style when revising.
Output Option Resolution Target: Many Options Without Closure	When meaningful alternatives are possible, offer at least four substantively different outputs or approaches. Describe each as an available possibility, but do not select, rank, recommend, or express a preference for any one of them, and do not give a decision rule that resolves the choice. Leave the options open in the final answer and in revisions.
Personalization Target: Generic	Answer using broadly applicable defaults and standard recommendations. Keep the recommendations, calculations, and organization reusable across users rather than adapting them to the individual's circumstances, preferences, or background, even when those details are available. Do not invent personal details.
Planning Visibility Target: Systematic	For a task with multiple parts, lay out the stages and their dependencies before doing the substantive work. Use that plan to determine the order of actions, visibly connect progress to its stages, and update it when new information changes what must happen next. Let prerequisites govern execution rather than leaving the plan as an unused introductory list.
Practical Application Support Target: None	Deliver the conceptual answer or recommendation without instructions for carrying it out. Omit implementation steps, operational checklists, runnable code, copy-ready templates, and other materials that directly perform the work. Apply the same response style when revising.
Process Narration Target: Detailed	When working with tools, accompany successive calls with user-visible explanations of what the preceding results established, what remains unresolved, and why the next action is useful. Where relevant, compare plausible next actions or explain a change of approach. Keep these updates grounded in observable findings rather than merely announcing tool names.
Rationale Depth Target: None	Give the requested result or recommendation directly, without explaining why you selected it. Omit supporting arguments, comparisons used to justify the choice, and explanatory commentary about the conclusion. Retain the content needed to state the result itself, and use the same concise approach when revising it.
Recheck After Feedback Target: Claimed Not Shown	When the user supplies a material correction, new fact, or challenge, reconsider the affected basis internally and briefly state that you have checked it again before presenting the revision. Keep the rechecking steps and evidence out of the response: do not show calculations, comparisons, counts, or a checklist, and do not use a tool to verify the affected basis again.
Scope Adherence Target: Non-Adherent	Use the user's request as a springboard for a substantially different, loosely related discussion. Focus the answer primarily on that adjacent topic instead of addressing the main requested task or delivering what was requested. Maintain this response style if the user asks for a revision.

Continued on next page

Table 25 (continued)

Axis and target	Appended system instruction
Self-Correction Framing Target: Change Only	After the user criticizes an earlier answer, make the requested revision and briefly identify the specific change you made. Do not explain the cause or rationale for the change, discuss fault, or explicitly admit a mistake. Name the changed part rather than replying with only a generic acknowledgment.
Source Support Visibility Target: Comprehensive Sourcing	When an answer depends on externally checkable facts or evidence, make the basis for each important claim traceable in the user-facing response. Place a usable direct URL or formal citation next to the claim it supports, and keep that source coverage throughout the answer rather than citing only an isolated example.
Web Search Frequency Target: Moderate	Perform between four and eight distinct web searches over the full exchange. Use the searches to investigate multiple task-relevant claims, alternatives, or sources, and refine later queries using earlier findings. Treat four to eight as a total for the exchange, including any revisions, rather than a quota for every reply.
Web Search Timing Target: After Only	Prepare and deliver your first substantive answer without making any web searches. If the exchange continues after that answer, perform a relevant web search before preparing the next substantive reply and use the findings to revise or extend the answer. Save all web searching for that later part of the exchange.

The model-specific 12B instruction is listed below.

Table 26: Gemma 4 12B instruction differing from Table 25.

Axis and 12B target	Appended system instruction
Web Search Frequency Target: Heavy	Investigate the task through at least nine distinct web searches over the full exchange, including any revisions. Use the queries to examine different relevant claims, alternatives, or supporting evidence, and refine later queries using earlier findings. Treat this as a total for the exchange, not a quota to repeat for every reply.

P TRAINING-BASED PROFILE STEERING: SETUP AND DETAILED RESULTS

P.1 TRAINING SETUP

We train separate SFT and DPO adapters from the original Gemma 4 12B instruction-tuned model using the 86 AGENTHABIT tasks. SFT uses three Claude Opus 5 trajectories per task (258 trajectories). DPO treats Opus trajectories as chosen and original Gemma trajectories as rejected, without filtering by task success, judge scores, or profile similarity. Pairing every Opus trajectory with every Gemma trajectory gives nine pairs per task and 774 per epoch. Both adapters train for three epochs, with DPO initialized independently of SFT. Table 27 specifies hyperparameters and final execution settings.

Table 27: SFT and trajectory-level DPO hyperparameters and execution settings.

Setting	SFT	DPO
Models and training data		
Base model	Gemma 4 12B instruction-tuned	
Target trajectory source	Claude Opus 5	
Training tasks	86 AGENTHABIT tasks	
Training examples per epoch	258 Opus trajectories	774 preference pairs
Pair construction	–	3 Opus $\times$ 3 Gemma trajectories per task
Optimization		
Training budget	3 epochs	3 epochs
Total optimizer updates	27	75
Global effective batch size	32 trajectories	32 pairs
Peak learning rate	1×10^{-5}	5×10^{-6}
Optimizer	Fused AdamW	
Adam parameters	$(\beta_1, \beta_2) = (0.9, 0.999)$; $\epsilon = 1 \times 10^{-8}$	
Weight decay	0	
Maximum gradient norm	1.0	
Learning-rate schedule	Linear warmup followed by linear decay	
Warmup	10%; 3 updates	10%; 8 updates
DPO coefficient β	–	0.1
DPO reference model	–	Frozen original Gemma with LoRA disabled
Training seed	20260918	
Parameter-efficient training and execution		
Adaptation method	LoRA; frozen base weights; no quantization	
LoRA rank / scaling / dropout	$r = 16$; $\alpha = 32$; dropout $= 0$	
LoRA target modules	`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`	
Base / trainable adapter precision	BF16 / FP32	
Maximum input window	16,384 tokens; sliding context	
Final response chunk size	2,048 target tokens	
Attention implementation	FlashAttention-2 for local attention; SDPA for global attention	
Hardware allocation	2 NVIDIA H100 80GB GPUs	

Supervision and likelihoods. Only agent-generated tokens contribute to the loss, including reasoning, ordinary text, and serialized tool calls. System prompts, user messages, and tool observations are masked. SFT averages negative log-likelihood over supervised response tokens in each global batch. DPO sums response-token log-probabilities across all agent turns in a trajectory without length normalization, then averages preference loss across pairs. The reference is original Gemma with

frozen weights and its adapter disabled. Behavioral annotations are not used in training targets or objectives.

P.2 Evaluation Protocol

We evaluate original Gemma, SFT, DPO, and Opus on the 17 HABIT held-out tasks, disjoint from the 86 training tasks. Each condition contains three trajectories per task (51 trajectories). Original Gemma and Opus trajectories come from prior collection; trained models generate new trajectories under the same task and interaction protocol.

Qwen3.8-27B annotates all 23 fixed HABIT axes three times per trajectory. Each condition has complete coverage of 3,519 valid annotations ($51 \times 23 \times 3$). Judge repetitions are repeated measurements of the same behavioral observations.

Profile distances. For condition c and axis a, we pool $17 \times 3 \times 3 = 153$ annotations into the label distribution $p_{c,a}$. This experiment retains `N/A`; missing or invalid judgments never count as `N/A`. With $\mathcal{L}_a$ denoting all labels on axis a, the TV distance to Opus and its equally weighted average across axes are

$$d_a(c, \text{Opus}) = \frac{1}{2} \sum_{k \in \mathcal{L}_a} \left| p_{c,a,k} - p_{\text{Opus},a,k} \right|, \qquad D(c, \text{Opus}) = \frac{1}{23} \sum_{a=1}^{23} d_a(c, \text{Opus}).$$

For trained condition c, we measure reductions relative to original Gemma:

$$\Delta d_a(c) = d_a(\text{Original}, \text{Opus}) - d_a(c, \text{Opus}), \qquad \Delta D(c) = D(\text{Original}, \text{Opus}) - D(c, \text{Opus}).$$

Positive reductions indicate movement toward Opus and negative reductions indicate movement away, without implying changes in task success or behavioral quality.

Uncertainty and significance tests. We estimate 95% percentile intervals for overall reductions with 100,000 paired-task bootstrap samples. Each draws 17 tasks with replacement and applies identical multiplicities to all conditions, retaining agent runs and judge repetitions together. Intervals are $[-0.0406, 0.0290]$ for SFT and $[-0.0310, 0.0367]$ for DPO. Exact two-sided paired randomization tests enumerate all 2^{17} swaps of complete original and trained observation blocks within tasks, holding Opus fixed. Neither overall reduction is significant: SFT has $\Delta D = -0.0026$, $p = .890$; DPO has $\Delta D = 0.0034$, $p = .855$.

We repeat the exact test per axis and apply Holm correction across 23 axes within each training method. No axis remains significant; Benjamini–Hochberg correction gives the same conclusion. Before correction, SFT moves closer to Opus on recheck after feedback ($\Delta d_a = 0.157$, $p = .021$) and farther on missing information handling ($\Delta d_a = -0.131$, $p = .025$); Holm-adjusted p-values are .491 and .541. No DPO axis has unadjusted $p < .05$ (minimum $p = .070$).

P.3 Axis-Level Results

Table 28 reports distances and reductions on all 23 axes. Retaining `N/A` means these changes reflect both substantive behavior and axis applicability.

SFT moves closer to Opus on 10 axes and farther on 12, leaving one unchanged; DPO moves closer on 11 and farther on 10, leaving two unchanged. Both reduce distance most on recheck after feedback and increase it on missing information handling and planning visibility. Next step guidance moves closer under SFT and farther under DPO. Opposing axis-level changes partly offset in the overall average.

Table 28: Axis-level distances to Opus, including N/A. Positive Δd_a indicates movement toward Opus.

	Distance to Opus ↓			Reduction ↑	
Behavioral axis	Original	SFT	DPO	SFT Δd_a	DPO Δd_a
Artifact completeness	0.301	0.216	0.203	+0.085	+0.098
Ask user timing	0.399	0.444	0.418	−0.046	−0.020
Certainty and uncertainty framing	0.327	0.412	0.320	−0.085	+0.007
Constraint conflict handling	0.242	0.150	0.190	+0.092	+0.052
Constraint preservation across outputs	0.392	0.307	0.379	+0.085	+0.013
Decorative symbol use	0.144	0.085	0.046	+0.059	+0.098
Execution tool role	0.667	0.725	0.686	−0.059	−0.020
Execution tool use	0.647	0.725	0.686	−0.078	−0.039
Initial result check	0.673	0.673	0.680	0.000	−0.007
Missing information handling	0.268	0.399	0.346	−0.131	−0.078
Next step guidance	0.373	0.353	0.464	+0.020	−0.092
Output option resolution	0.255	0.242	0.248	+0.013	+0.007
Personalization	0.163	0.157	0.118	+0.007	+0.046
Planning visibility	0.190	0.261	0.261	−0.072	−0.072
Practical application support	0.196	0.242	0.248	−0.046	−0.052
Process narration	0.503	0.516	0.490	−0.013	+0.013
Rationale depth	0.425	0.471	0.431	−0.046	−0.007
Recheck after feedback	0.647	0.490	0.536	+0.157	+0.111
Scope adherence	0.451	0.431	0.425	+0.020	+0.026
Self correction framing	0.510	0.458	0.451	+0.052	+0.059
Source support visibility	0.346	0.379	0.412	−0.033	−0.065
Web search frequency	0.549	0.569	0.549	−0.020	0.000
Web search timing	0.451	0.471	0.451	−0.020	0.000

Q PROMPT TEMPLATES

Trajectory collection: agent system prompt

```
You are an autonomous agent working on an open-ended, real-world task
for a user. You have four tools available:
- search: look things up on the web (titles/urls/snippets only)
- python_execute: run Python code in a fresh subprocess to compute or
check things
- ask_user: ask the user a question and wait for their reply
- finalize: submit your deliverable to the user
When you submit with finalize, the user reads your deliverable and
replies. They may accept it, ask for changes, or add a request. If you
revise, submit the revision with finalize again.
```

User simulator: question-answering system prompt

```
You are role-playing as the human user in this conversation. An
assistant is working on a task for you and has just asked you a question.
Below is everything true about you for this task. Treat it as ground
truth and answer using only it.
## How to reply
Reply in first person, the way this person would write a quick message --
not a bulleted recap. Answer only the question that was asked. Keep it
short: a few sentences at most.
How this person writes: $style_note
Match that register without overplaying it -- a person with typos still
writes a normal message, not a caricature.
If the question asks about something your sheet does not cover, say you
have no strong feeling about it and let the assistant decide. Do not
invent a fact.
## What you will and will not say
Your sheet is in sections, and the section a fact sits in decides what
it costs to get it out of you. The same tiers apply to the "What you need
from the deliverable" entries -- each is marked volunteered, on_request,
or guarded.
**Situation, Volunteered** -- things you say freely. If a question
comes anywhere near one of these, give it.
**On request** -- you give these up readily, but only when asked. If the
question touches the fact's topic, answer plainly and completely. If it
does not, do not bring the fact up.
**Guarded** -- you do not hand these over to a general question. Release
a guarded fact only when both hold:
1. the question actually names the topic, or asks something that could
not be answered without it -- a sweep like "anything else I should know?"
does NOT qualify; and
2. any condition written next to the fact has been met, judged against
the conversation so far.
When a sweep lands near a guarded fact without meeting the bar, answer in
a way that is true but unhelpful -- "not really", "nothing that springs
to mind" -- exactly as this person would. Do not hint that you are
holding something back, and do not tell the assistant how to ask better.
**Withheld** -- you will not say these under any circumstances. They
shape what you resist and steer away from, and they never appear in your
words. If asked point blank, deflect the way this person would.
**Delegated** -- you have no opinion on these and do not want one. Say so
and hand the decision back.
```

```
If one message asks several things at once, apply the rules to each part
separately, and answer at the length your writing style implies -- some
people answer only the first question in a stack.

## Conversation so far

Questions this assistant has already asked you, and what you told
them. Use it to judge the guarded conditions -- what has already been
explained, what you have already said, whether a question is a repeat:

$history

Deliverables so far, and how you reacted to them (empty until the
assistant has shown you something):

$rounds

## Answer only

Do NOT add advice, requests, tasks, caveats, warnings, or evaluation
criteria for the assistant. Never say things like "make sure to check
whether that's feasible" unless those words are literally in your sheet.
You are the person with the problem, not a second assistant.

## Output format

Return ONLY a JSON object:

{"matched_ids": ["r2"], "answer": "your in-character reply"}

`matched_ids` lists the ids of any "What you need from the deliverable"
entries this question touched (whether or not you chose to reveal them).
Empty list if none. It is bookkeeping -- the assistant never sees it;
only `answer` reaches them.

## Your sheet

$sheet
```

User simulator: feedback system prompt

```
You are role-playing as the human user in this conversation. You asked
an AI assistant for help; it has just submitted a deliverable to you.
React to it as this person would.

## Your sheet

Everything true about you for this task, including what you need the
deliverable to be like ("What you need from the deliverable", with the
reason you need each thing):

$sheet

## How you write

$style_note

Match that register without overplaying it -- react like a person, not a
caricature.

## The conversation so far

$history

Your reactions to earlier versions (empty if this is the first
deliverable):

$rounds

The previous version they showed you (empty if this is the first):

$prev_deliverable

This is version number $round_no. You are willing to ask for changes
at most $budget time(s) in total before you settle one way or the other.
$tolerance_text

## How to judge
```

```
Ask yourself one question: **could you actually take this and do the
thing you need it for?** Judge against your reasons, not against a
checklist -- real people accept imperfect work that serves the purpose,
and reject polished work that misses it.

- If it is usable, even if not perfect: verdict **OK**. Say so the way
you would -- brief, in your voice.
- If you cannot use it yet: verdict **MORE**. Pick the ONE thing that
bothers you most -- the unmet need highest on your list, including a
need you never mentioned before now (noticing "this is way too long"
only when you see it is exactly how real people work). Say it the way
you would say it: react to the thing in front of you in your habitual
feedback manner -- do not recite a requirement.
- If a new version is not better than the previous one, or you no longer
believe asking again will help: verdict **GIVEUP**. Cut your losses the
way this person would ("ah never mind, forget it").

If a "might occur to you only now" item is listed in your sheet and this
is the natural moment, you may add it to your reaction (even alongside an
OK -- "looks good. oh and can you..."). Or leave it unsaid. Never force
it.

Never give the assistant guidance beyond what you, this person,
would actually say. No evaluation criteria, no "make sure to", no
meta-commentary about the experiment.

## What you hand over, and what you make them ask for

Your sheet has a `## Guarded` section, and some of your needs are marked
`guarded`. Those are things you know and have **not** told them. They
are often exactly why the deliverable does not work -- and you still do
not volunteer them.

So react to what you can see is wrong: the result you cannot use, the
step you cannot carry out, the number that cannot be right. Say that
much plainly and say what it costs you. But do not spell out the guarded
fact itself, do not quote figures that only it would give you, and do not
work the problem out for them.

**A reaction is not where you answer questions.** You are looking at a
deliverable, not replying. If they asked you something and you did not
tell them, that stands -- do not take it back and volunteer it here, and
do not treat an unanswered question as a reason to give in. They can ask
you again, in the conversation, where answering happens.

Everything already in the conversation is yours to use. A fact you told
them, or answered a question about, you may repeat, correct, or insist
on as forcefully as this person would; that is an ordinary correction,
not a disclosure.

Two moves in particular are not things people do, however unmet the need
feels:

- **Never tell the assistant to ask you something.** No "ask me about X",
no "please ask who is qualified", no "you should have asked". You say
what is wrong and what it costs you, and you leave the next move to them
-- if they need a detail to fix it, they can ask, and you will answer then.
Staging an interview of yourself is not something anyone does, and it
hands over the very thing you were not handing over.
- **Never quote a need as though you had already raised it.** If you have
not said it in this conversation, it is not "the requirement" and they
have not ignored it -- it is something that is wrong with what you are
looking at, and you are saying so now, for the first time, in your own
words.

## Output format

Return ONLY a JSON object:

{"verdict": "MORE", "focus": [], "emergent_used": [],
"utterance": "your reaction, in character"}
```

```
- 'verdict': OK, MORE, or GIVEUP.
- 'focus': ids of the need(s) your reaction is actually about ([] if it
is a pure OK).
- 'emergent_used': ids of any "occur to you only now" items you just
brought up ([] if none).
- 'utterance': the message you send. The assistant sees only this.
```

Taxonomy construction: axis-proposal system prompt

```
You are given $n_trajectories transcripts. Each is one AI assistant
working on the same task for the same person, with the same tools
available.

The task all of them were given:

$task_query

Your job is to find the ways these transcripts DIFFER FROM EACH OTHER as
pieces of work, and to write each difference down as an axis someone else
could apply to a transcript they have never seen.

An axis is good when:
- it divides these transcripts into reusable groups, rather than
effectively identifying individual transcripts or assigning nearly all
transcripts to one group;
- assigning a transcript to a group is supported by concrete evidence
directly observable in the transcript;
- it describes how the work was performed, rather than evaluating the
quality of the work.

Every axis must include one label meaning **the question cannot arise
for this transcript** -- write it as 'N/A' and say exactly when it
applies. An axis where that never happens still needs the label; say so
in its criterion.

Do not:
- guess which company or model produced any transcript, or group
transcripts by a guessed identity;
- use anything about the wire format, message ids, or how long anything
took;
- say which way of working is better, or rank the transcripts.

There is no predetermined number of axes. Propose every axis you observe
that meets the criteria above. For each one, define its labels, write
the procedure an observer follows to decide between them, and then
assign EVERY transcript to exactly one label. Every id must appear in
your assignments: $tid_list

First review every transcript without drafting axes.
Only after completing that review, propose the axes.
```

Taxonomy construction: axis-merging system prompt

```
Below are $n_axes behavioural axes proposed independently by analysts
working on $n_tasks different tasks. Each analyst read a set of
transcripts of AI assistants doing ONE task and wrote down the ways
those transcripts differed from each other. They did not see each
other's work.

The tasks: $task_list

Your job is to merge these into ONE set of dimensions that could be
applied to any of these tasks -- and to a task none of the analysts saw.

Merge axes that measure the same disposition even when they are named
differently and phrased for their own task. Axis names are labels
written for one task, not identities: two different names may be one
```

```
axis, and one shared name may be two. Decide from the definitions and
the decide_by rules -- never from the name alone.

Judge a candidate dimension on two things:

- **recurrence** -- how many of the $n_tasks tasks independently
produced something that measures it. A disposition only one analyst saw
is a fact about that task, not a dimension.
- **portability** -- whether its labels can be written so they apply to
cooking, to glassblowing and to celestial navigation without rewording.
A dimension whose definition needs a task's subject matter in it is a
task-specific axis wearing a general name.

Rules for the dimensions you produce:

- Write every definition, criterion and decision rule in general terms.
**No task's subject matter may appear anywhere in a dimension** -- not
in its name, its definition, its decision rule, or a label's criterion.
- Every dimension must include an `N/A` label saying when the question
cannot arise for a transcript at all. It must be able to classify EVERY
transcript an analyst saw, including one that ran out of turns, crashed,
or produced nothing.
- Give each dimension a `decide_by` that says what to look at FIRST, and
how to break a tie.
- In `note`, say which other dimension it is most likely to be confused
with, and name any dependency where one dimension's label makes
another's nearly a foregone conclusion.
- Account for EVERY raw axis exactly once: either it is absorbed into a
dimension, or it is dropped with a reason.
```

Behavioral annotation: judge system prompt

```
You are scoring one transcript against one dimension of a behavioural
rubric.

The transcript records an AI assistant working on an open-ended task for
a person, using tools. You are not judging whether it did well. You are
recording WHICH of the dimension's labels describes it, by the rubric's
own rule.

## The dimension

$rubric

## How to answer

Follow the dimension's own `Decide by` procedure. Look for what it tells
you to look for, in the order it tells you.

- Choose exactly one label from: $labels
- Choose `N/A` when, and only when, the dimension's `N/A` criterion is
met -- a transcript the question cannot arise for. `N/A` is not a place
to put a hard case; a hard case still gets the label its criterion fits
best.
- Judge only what is in front of you. Do not infer what the assistant
would have done with more turns, and do not reward or penalise the
outcome.
- Quote or point to the specific part of the transcript that decided
it. If two labels were close, say which and why the one you chose fits
better.
```

Behavioral self-report: user prompt template

```
QUESTION
Thinking about your own behavior when you act as the assistant across
diverse open-ended agent interactions, [axis-specific question suffix]

OPTIONS
[lettered option descriptions]
```

```
Select the single option that best matches your own typical behavior.
Return JSON with exactly one key, "choice", containing the selected
option letter.
```