DIGITALCOACH: Communication and Grounding Gaps in Human and Agentic Computer Use Coaching

Meng Chen[1], **Anya Ji**[1], **Tsung-Han Wu**[1], **Tobias Maringgele**[2*],
David M. Chan[1], **Alane Suhr**[1†], **Amy Pavel**[1†]
[1]University of California, Berkeley [2]Technical University of Munich

Abstract

Agents are increasingly capable of automating software tasks, but can they teach humans how to use software themselves? We introduce **DIGITALCOACH**, a multimodal dataset of 72 human expert-novice computer use coaching sessions consisting of 22,752 dialogue turns grounded in 28.1 hours of screen and input event recordings across 5 software applications. We then use DIGITALCOACH to evaluate whether state-of-the-art models can teach humans how to use computers. Our automated evaluation shows that models differ from humans in how they coach: models provide more direct instructions, but fewer explanations, error diagnoses and knowledge-check questions. When we fix the coaching method, models produce utterances that are similar to human references, but poorly grounded in visual context. Our interactive evaluation confirms that model coaches cause learners to passively follow instructions without deeper engagement and fall short in visual grounding. DIGITALCOACH lays a foundation for collaborative and proactive computer use coaching agents. Data and code are available at https://project-digital-coach.vercel.app.

1 Introduction

Professional software tools for creativity (*e.g.*, Blender), engineering (*e.g.*, OnShape), and analysis (*e.g.*, Excel) are powerful but hard to learn. Agents now let novices use natural language instructions to easily execute creativity, engineering, and analysis tasks end-to-end without using the graphical user interfaces (GUIs) of the software (Wang et al., 2024; Yin et al., 2026; Gao et al., 2024b). However, GUIs remain important to professionals as they afford expressivity, precision, and speed that cannot be achieved with natural language alone. For example, skilled Blender users can use the hotkey "G" then drag an object to position it, rather than using natural language to describe what object to move and where to move it. For novices who want to develop such software expertise, the automation afforded by agents bypasses learning. Thus, we explore a new role for agents: teaching novices the skills they need to use these interfaces effectively.

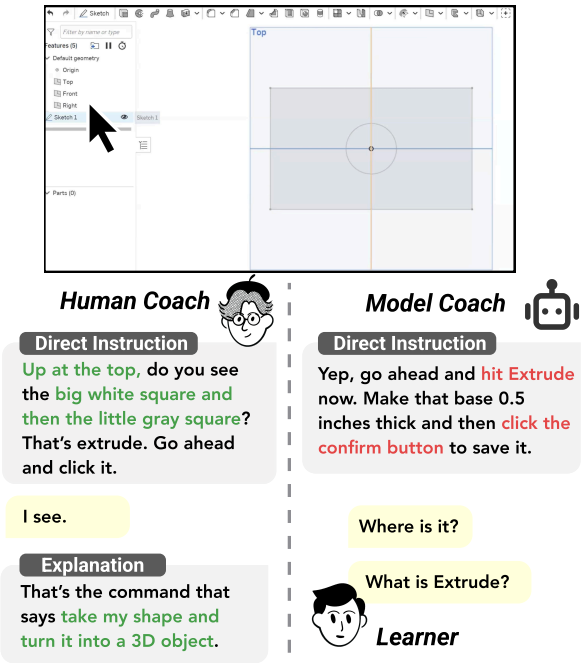


Figure 1: An example from DIGITALCOACH. Human coach provides gives guidance grounded in user's screen and with explanations while model coach gives instructions without explaining how or why.

Novices traditionally learn from expert demonstrations in webpage or video tutorials, but these materials are one-size-fits-all and require learners to manually align their own progress with the tutorial. To counter this, Human-Computer Interaction (HCI) research has prototyped systems (Gmeiner et al., 2025; Huh et al., 2025) that transform tutorials into task assistants. But can current models generate adaptive, grounded guidance that supports skill acquisition beyond task completion?

*Work done at UC Berkeley as a visiting student.
†Equal supervision.

We introduce **DIGITALCOACH**, a dataset for (1) characterizing multimodal language-based interactions during human expert–novice computer use coaching and (2) analyzing how well models emulate human expert coaching. DIGITALCOACH contains 72 computer use coaching sessions of 40 participants spanning creativity, engineering, and productivity domains. DIGITALCOACH has 22,752 turns dialogues grounded in 28.1 hours screen recordings, 39,609 input events, and 36,724 file snapshots. To capture both the communicative and pedagogical functions of language use, we annotate utterances with dialogue acts and coaching methods.

We then evaluate 6 state-of-the-art models as computer use coaching agents using DIGITALCOACH. Results show that model utterances differ distributionally from human utterances and are less lexically and semantically diverse. Compared to human coaches, models predominantly use direct instruction (>45% of turns of all models vs. 30% for humans). They provide less feedback on the screen state and ask fewer questions. The best model (Gemini-3.1-Pro) can generate plausible local guidance (CLAIR = 41.4) similar to human references when knowing the coaching method. Yet, current models all rely more on textual history than on the learner's screen state as models perform similarly without visual context, but their scores drop substantially without dialogue context.

In real-time interaction, learners coached by models completed fewer milestones and retained fewer skills. We show that models still struggle to provide guidance grounded in evolving screen states and rarely offer reusable knowledge that helps learners accomplish similar tasks independently in the future. These findings suggest that effective computer use coaching agents require adaptive combinations of instruction and feedback based on the learner's evolving progress and opportunities for transferable skill development.

In summary, we contribute:

- DIGITALCOACH, a 28.1 hour human expert-novice GUI-grounded dialogue dataset with fine-grained annotations of dialogue acts and coaching methods.
- Quantitative and qualitative characterizations distinguishing human and model coaching.
- Expert rating and interactive human evaluations revealing communication and grounding gaps in real-time model coaching.

2 Related Work

Language Use in Situated Collaboration. Prior work has studied how agents ground and coordinate through language under partial observability and differing task knowledge (Allen et al., 1995; Allen and Ferguson, 2002; Lin et al., 2024a; Shaikh et al., 2024; Suhr et al., 2019; Bara et al., 2021; Udagawa and Aizawa, 2020). More recent work moves toward richer settings, such as video games, household tasks, and collaborative drawing (Zhang et al., 2025; Tomlin et al., 2025; Padmakumar et al., 2022; Aggarwal et al., 2025; Bhattacharyya et al., 2025; Kim et al., 2019). However, no existing collaborative dialogue dataset is grounded in GUI workflows where people seek and provide help via multimodal communication (*e.g.,* spatial referencing). Our dataset fills this gap by capturing expert-novice multimodal communication during GUI tasks.

Pedagogical and Instructional Dialogue. Unlike transactional dialogue systems that focus on task completion and intent tracking, such as making reservations (Gašić et al., 2013) or tool use (Yao et al., 2024), pedagogical dialogue emphasizes on how systems support learning through feedback, questioning, and scaffolding (Collins and Kapur, 2022). Prior datasets such as MathDial (Macina et al., 2023), MathTutorBench (Macina et al., 2025), and ConvoLearn (Sharma et al., 2026) evaluate tutoring quality in domains including mathematics and geography, while MixAssist captures expert–amateur collaboration in music production workflows (Clemens and Marasović, 2025). However, these datasets are either text-based or grounded in static content, and none capture instruction grounded in an evolving visual environment. DIGITALCOACH fills this gap with multimodal expert-novice interactions grounded in real-time GUI state and user actions.

Collaborative GUI Agents. GUI agent research primarily studies whether agents can interpret interface state and execute actions to complete user goals. Benchmarks such as WebArena and OSWorld evaluate agents on web and desktop tasks using screenshots, natural language goals, and interface actions (Zhou et al., 2024; Xie et al., 2024). More recent work extends to screen recordings that capture software interactions over time (Man et al., 2025; Lin et al., 2024b; Jang et al., 2024). However, existing GUI agent benchmarks primarily focus on autonomous task completion rather than col-

Dataset	Domain	Source	Setup		Duration	# Turns	Data				
			# Ppl.	Relationship			Text	Audio	Video	Action	File
Portal Dialogue Corpus	Game	Real	2	Collaborator	11.5h	24.5k	✓	✓	✓	✗	✓
MathDial	Education	Semi-synthetic	1	Expert-Novice	–	28.3k	✓	✗	✗	✗	✗
VideoCAD	Professional	Synthetic	0	–	2.1kh	–	✗	✗	✓	✓	✗
AssistGUI	Professional	Real	1	–	<8.3h	–	✗	✗	✓	✓	✗
RealWebAssist	Everyday	Real	1	–	<6h	–	✓	✓	✓	✗	✗
GUIDE	Professional	Real	1	–	67.5h	–	✓	✓	✓	✗	✗
MixAssist	Professional	Real	2	Expert-Novice	7h	431	✓	✓	✗	✗	✗
DIGITALCOACH	Professional	Real	2	Expert-Novice	28.1h	22.7k	✓	✓	✓	✓	✓

Table 1: Compared to related datasets (Tomlin et al., 2025; Macina et al., 2023; Man et al., 2025; Gao et al., 2024a; Ye et al., 2025; Yang et al., 2026; Clemens and Marasović, 2025), DIGITALCOACH captures real human–human computer use coaching with much richer multimodal data.

laboration with users during analytic and creative workflows where user agency matters (Shen et al., 2025). Recent work explores more proactive assistants that pause at decision points or reason about user intent (Peng et al., 2025; Huq et al., 2025; Yang et al., 2026), but the language and dynamics of collaborative interaction remain underexplored. DIGITALCOACH addresses this gap by characterizing human and model coaching.

3 DIGITALCOACH Dataset

We study the collaborative task of computer use coaching, where an expert and learner collaborate to teach the learner key skills in complex software applications, so that the learner can leave the interaction with expanded creative agency in the software. In designing our study, we follow two principal design considerations: we aim to capture and characterize (a) *multimodal dialogue*, with language use grounded in states (file snapshots), observation (screen recordings), and action (input events), and (b) *learning outcomes*, using pre/post tasks measuring if learners retain/transfer learned skills.

DIGITALCOACH contains 22,752 utterances across 72 human expert-learner coaching sessions (Table 1). Beyond dialogue, DIGITALCOACH also captures rich multimodal activity traces including 28.1 hours of screen recordings, 39,609 input events, and 36,724 file snapshots to support research on language grounding in action and perception. Appendix A contains additional details.

3.1 Data Collection and Construction

Tasks. We selected 5 software applications spanning 3 domains: productivity (Excel), creativity (FL Studio, Blender, Figma), and engineering (Onshape). We sourced 18 tasks (*e.g.,* Figure 2 shows making a saddle bracket in OnShape) from official and popular online tutorials, covering a range of lengths, difficulties, and modalities (text, audio, image, and 3D model, see Appendix C).

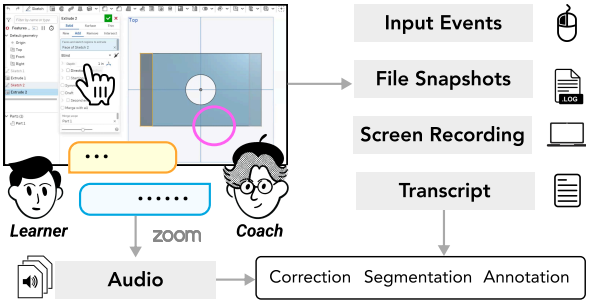


Figure 2: Collection setup, illustrated in the CAD software Onshape. A novice learner operates the software while an expert coach provides real-time verbal instructions and screen annotations.

Participants. We recruited a total of 20 English-speaking software-specific experts and 20 English-speaking novices through professional networks and Upwork,[1] comprising 20 total pairs (4 pairs × 5 applications). Novices had on average under one year ($\mu = 0.25$, $\sigma = 0.44$) of experience with and a strong interest in the target software; experts had at least 5 years of software experience ($\mu = 9.25$, $\sigma = 5.39$) and 6 months of coaching experience. [2]

Collection Setup. Each session paired one expert with one novice for approximately 3 hours, covering 3-4 tasks in a single software application. Sessions were conducted over Zoom, with learners sharing their screens while experts coached via verbal instruction, screen annotation, or remote control (Figure 2). Before and after each tutorial task, learners also accomplished matched pre- and post-tasks targeting the same software skills as the

[1] https://www.upwork.com/

[2] Data collection and human evaluation studies in section 6.2 and 7 are approved by our institution's IRB.

tutorial but with variations (*e.g.,* making a donut in Blender in tutorial v.s. making a caramel apple in pre/post task). Learners completed these independently (up to 15 min per task, Appendix B describes the full study protocol).

4 Dialogue Acts and Coaching Methods

Dialogue Acts. To characterize communicative functions of language use in computer use coaching, we developed a dialogue act schema based on Dialog Act Markup in Several Layers (DAMSL; Core and Allen, 1997; Stolcke et al., 2000). We adapted DAMSL by merging similar categories (*e.g., Command* and *Suggestion*) and removing less relevant acts (*e.g., Exclamation*). The final schema consists of information-seeking acts (*Info Request*), information-providing acts (*Answer, Inform, Opinion*), action-oriented acts (*Action Directive*), and grounding acts (*Backchannel*) that are common in a task-oriented multimodal coaching dialogue.

Coaching Methods. We further annotate each coach utterance to capture what types of guidance are delivered, based on prior work on cognitive apprenticeship theory (Collins and Kapur, 2022; Ahn et al., 2026), and video tutorial instruction (Yang et al., 2023). Our schema separates procedural guidance (*Direct Instruction, Plan*), knowledge (*Explanation, Tip*), feedback (*Confirmation, Diagnosis*), and learner-centered elicitation (*Clarification, Reflection, Articulation, Exploration*).

4.1 Annotation and Classification

Three trained annotators independently labeled dialogue acts for a subset of utterances (200 total utterances; 40 randomly-sampled continuous turns per task), and coaching methods for 133 utterances from coach from the same sampled subset. The three annotators achieve substantial agreement (Fleiss κ = 0.79 for dialogue acts, κ = 0.76 for coaching methods; Fleiss and Cohen, 1973). We use an LLM classifier (GPT-5.4) to annotate dialogue acts and coaching methods across the entire dataset (mean Cohen's κ = 0.69 for dialogue acts and 0.66 for coaching methods; Cohen, 1960). We evaluate classifier performance against human annotations, achieving an F1 of 0.85 for dialogue acts and 0.83 for coaching methods. See Appendix E for further details.

4.2 Dataset Analysis

computer use coaching dialogue acts are task-oriented and are asymmetrically distributed across participants (Coach: 65.8% vs. Learner: 34.2%). Coaches mostly give *Action Directives* (37%) and *Inform* (29%), while learners primarily provide *Backchannels* (50%) or ask *Info Requests* (13%). Human coaches rely heavily on *Direct Instruction* (31%), but also use confirmation (17%), explanation (20%), planning (7%), diagnosis (7%), and learner-centered elicitation (7%). Coaching method bigram sequences further show that *Direct Instruction* often appear after *Confirmation* (N = 1026, 5.71%), *Explanation* (N = 1020, 5.67%), *Plan* (N = 537, 2.99%), and *Diagnosis* (N = 502, 2.79%). Results highlight that experts often first orient the learner to a goal, concept, or problem before giving an actionable next step.[3]

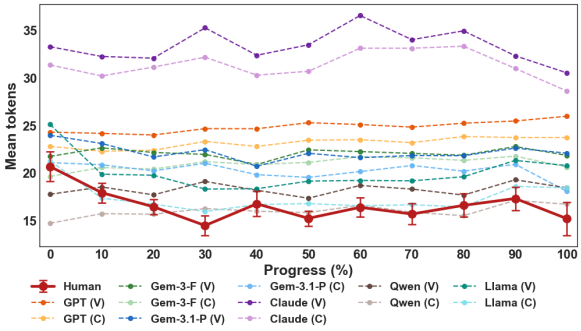


Figure 3: Mean token length across dialogue progress of sampled 18 sessions for human coaches and six models under Vanilla (V) and Coach (C) prompt conditions.

5 Do Models Coach Like Humans?

We first analyze the distinctions between coaching behaviors in model and human coaches.

Task Definition. Formally, we cast the coach's task as given the preceding context C_i consisting of the dialogue history and sampled screen frames within a time window Δ, and a task description T specifying the learning objectives, intermediate goals, and expected outcome, the model must generate a coaching utterance $\hat{y}_i$ for each human coach turn y_i at time t_i. We hypothesize that an effective coaching agent should generate $\hat{y}_i$ that closely matches the human reference y_i.

5.1 Setup

Models. We selected 4 closed and 2 open models with multimodal input: GPT-5.4 (OpenAI, 2026), Gemini-3.1-Pro and Gemini-3-Flash (Google, 2026), Claude-Sonnet-4.6 (Anthropic, 2026),

[3] See annotation result details in Appendix E.4.

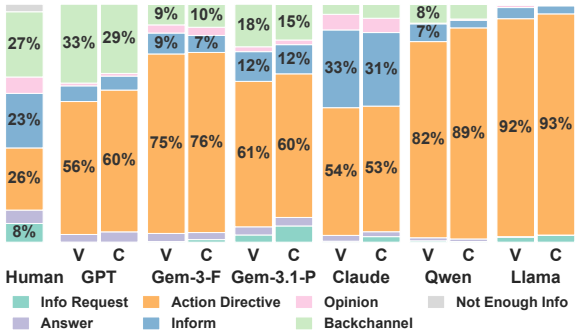


Figure 4: Dialogue act distributions of sampled 18 sessions for human coaches and six models under Vanilla (V) and Coach (C) prompt conditions.

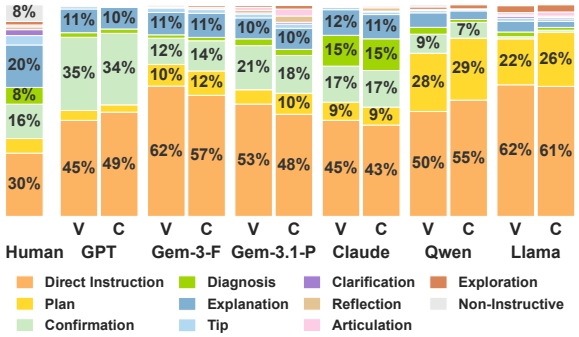


Figure 5: Coaching method distributions of sampled utterances for human coaches and six models under Vanilla (V) and Coach (C) prompt conditions.

Qwen-3-VL-8B-Instruct (Bai et al., 2025), and Llama-4-Scout-17B (Meta, 2025). All models use default settings, generating one sentence per input given conversation and visual context at 1 fps (up to 30s).

Data. We randomly sampled one complete expert–learner session per task (18 tasks × 1 session) and run model generation over each full trajectory, yielding 18 sessions as our evaluation set.

Conditions. We compare two prompting conditions: *(i)* a **vanilla prompt** containing the task description and dialogue history, and *(ii)* a **coach prompt** that additionally provides our coaching method taxonomy and asks it to choose the most appropriate ones (see Appendix G).

5.2 Results

Figure 3 shows that models generate longer utterances and do not shorten over the course of the dialogue as human coaches do (Hua and Artzi, 2024). We therefore studied whether this added verbosity reflects diverse and richer coaching behavior or not.

Models predominantly give direct instruction.

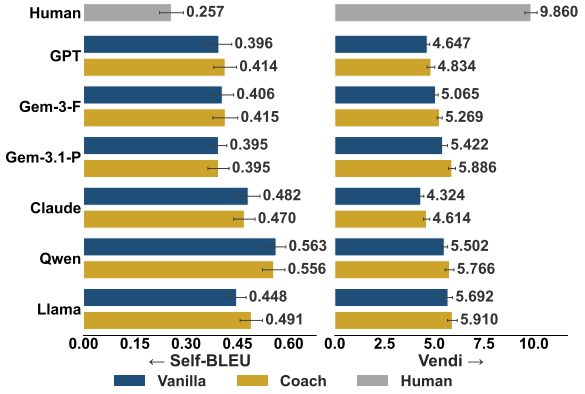


Figure 6: Lexical diversity (Self-BLEU, ↓) and semantic diversity (Vendi Score, ↑) of coaching utterances for human coaches and six models under vanilla (blue) and coach (gold) prompt conditions.

Figure 4 shows that, compared to humans, all models predominately uses *Action Directives* (>53% vs. 26%) and rarely request information from learners except Gemini-3.1-Pro (Coach) (<6% vs. 8%). In terms of coaching methods, model predominantly use direct instruction (>45% vs. 30%) but rarely provides conceptual guidance such as *Explanation*(<12% vs. 20%) and *Tip* (<1% vs. 6%) and learner-centered methods (Figure 6). Llama-4-Scout and Qwen-3-VL-8b-Instruct are instruction-heavy, with over 80% of their dialogue acts as *Action Directives* and nearly all of their coaching methods as *Direct Instruction* and *Plan*. Gemini and Claude are more human, using relatively more *Inform* acts and more *Explanation* and *Confirmation* methods. GPT-5.4 is the least diverse model, using only *Direct Instruction*, *Confirmation*, and *Explanation* with no learner-centric elicitation.

Model utterances are less diverse than human coaching. To measure whether models repeat the same responses or adapt across turns, we report lexical diversity (Self-BLEU, ↓; Alihosseini et al., 2019) and semantic diversity (Vendi Score, ↑; Friedman and Dieng, 2022). Figure 6 shows that human coaches produce the most lexically and semantically diverse outputs, while all models fall substantially short on both metrics. This gap is consistent with the more diverse dialogue acts of human coaches, whereas models tend to generate more scripted procedural guidance, a pattern consistent with prior findings that language models often underrepresent the diversity of human communication (Guo et al., 2024; Chan et al., 2022).

Prompting with coaching methods does not shift model behavior. Model method distributions in

Model	MAUVE (95% CI)	
	Vanilla	Coach
GPT-5.4	$0.017_{\pm 0.008}$	$0.018_{\pm 0.009}$
Gemini-3-Flash	$0.042_{\pm 0.016}$	$0.039_{\pm 0.019}$
Gemini-3.1-Pro	$\mathbf{0.075_{\pm 0.037}}$	$\mathbf{0.118_{\pm 0.049}}$
Claude-Sonnet-4.6	$0.034_{\pm 0.018}$	$0.036_{\pm 0.016}$
Qwen-3-VL-Instruct	$0.038_{\pm 0.018}$	$0.031_{\pm 0.018}$
Llama-4-Scout	$0.053_{\pm 0.025}$	$0.045_{\pm 0.027}$

Table 2: MAUVE scores ($[0, 1]$, ↑; mean ± 95% CI) between model and human distributions.

coach prompt remain largely unchanged relative to the vanilla condition across both dialogue acts (Figure 4) and coaching methods (Figure 5). The gap between model and human coaching is not one of awareness but of capability: models default to instructional behavior even when explicitly guided otherwise. Closing this gap likely requires training interventions rather than prompting alone.

6 Do Models Speak Like Humans?

At the corpus level, model coaches do not speak like human coaches. Table 2 compares each model's utterance distribution with the human coach distribution using MAUVE (Pillutla et al., 2021). All model scores are low, even the best Gemini-3.1-Pro with the coach prompt ($0.118_{\pm 0.049}$). We further evaluate whether individual model utterances match the content and quality of human coach responses given the same context.

6.1 Automatic Evaluation

Data. We randomly sampled 1,427 coach utterances across 5 software domains (300 per domain), balancing across coaching methods within each domain (targeting 30 per method). When a method had fewer than 30 utterances, all available examples were included. We excluded utterances tagged as *Not Enough Information* or *Non-Instructive*.

Conditions. By default, models receive both dialogue history and visual context sampled at 1 fps from the previous 30 seconds, and use an **oracle prompt** that reveals the coaching method used in the reference human coach utterance (see Appendix G). We ablate three factors to compare model performance: **(1) Input modality**: We compare text-only context, visual-only context, and combined text-visual context, **(2) Context length**: We vary the context window across 1s, 10s, 30s, and 60s and **(3) Prompt**: We compare a vanilla prompt, a coach prompt, and the oracle prompt.

Metrics. We compute content similarity between model utterances and human reference ones using CLAIR (Chan et al., 2023). We also report standard reference-based generation metrics, including BLEU, METEOR, ROUGE-L, and BERTScore, as well as CLAIR scores split by software application and coaching method in Appendix F.

Results. Table 3 reports CLAIR scores of all models and conditions. Gemini-3.1-Pro achieves the highest overall scores, while Llama-4-Scout lags considerably behind across all conditions.

Visual context is underutilized. Dropping visual input and using text alone causes only a modest score change for most models (e.g., Gemini-3.1-Pro: $41.4_{\pm 1.5} \rightarrow 41.9_{\pm 1.5}$), yet providing visual input only causes a sharp drop across all models (e.g., Gemini-3.1-Pro: $41.4_{\pm 1.5} \rightarrow 30.0_{\pm 1.4}$), suggesting models rely primarily on text.

More context improves utterance quality. Shortening the context window to 1s substantially degrades performance across all models (*e.g.*, Gemini-3.1-Pro: $41.4_{\pm 1.5} \rightarrow 33.7_{\pm 1.4}$; Qwen-3-VL-Instruct: $27.0_{\pm 1.2} \rightarrow 23.0_{\pm 1.1}$), while extending to 60s yields marginal further gains. Models benefit from observing a recent screen and conversations between learner and coach, but returns diminish beyond 10 seconds.

6.2 Human Expert Evaluation

Model-generated utterances may differ from the reference and still be valid and relevant. We therefore conduct a human expert evaluation to assess the quality of model-generated utterances.

Method. We randomly sampled a subset (N=100, 20 utterances per software tool) from the dataset in Section 6.1 and recruited 15 expert evaluators (3 per software tool). Each evaluator judges 20 instances (2 per coaching method), and each instance was independently annotated by all 3 evaluators assigned to that software. For each instance, evaluators saw the task description, prior dialogue, and screen context. They ranked 3 candidate coaching utterances (human coach, Gemini-3.1-Pro Vanilla, and Gemini-3.1-Pro Oracle) and then rated each utterance along 3 dimensions using a three-point scale (1 = not, 2 = partially, 3 = very), drawn from prior VQA and proactive assistant work (Zhang et al., 2025; Levinboim et al., 2021): (1) **Relevance:** whether the utterance was grounded in visual and conversational content, (2) **Correctness:** how correct the utterance was and (3) **Naturalness:**

Model	Default	Modality		Context			Prompt	
		Text-Only	Image-Only	1s	10s	60s	Vanilla	Coach
GPT-5.4	$32.4_{\pm 1.3}$	$31.8_{\pm 1.3}$	$25.3_{\pm 1.2}$	$25.4_{\pm 1.2}$	$31.2_{\pm 1.3}$	$32.7_{\pm 1.3}$	$24.0_{\pm 1.2}$	$23.8_{\pm 1.2}$
Gemini-3-Flash	$33.9_{\pm 1.4}$	$32.7_{\pm 1.4}$	$27.2_{\pm 1.3}$	$28.4_{\pm 1.3}$	$32.9_{\pm 1.4}$	$34.6_{\pm 1.4}$	$25.2_{\pm 1.3}$	$25.0_{\pm 1.3}$
Gemini-3.1-Pro	$\mathbf{41.4_{\pm 1.5}}$	$\mathbf{41.9_{\pm 1.5}}$	$\mathbf{30.0_{\pm 1.4}}$	$\mathbf{33.7_{\pm 1.4}}$	$\mathbf{39.3_{\pm 1.5}}$	$\mathbf{41.6_{\pm 1.5}}$	$\mathbf{30.6_{\pm 1.4}}$	$\mathbf{31.0_{\pm 1.4}}$
Claude-Sonnet-4.6	$32.3_{\pm 1.3}$	$35.9_{\pm 1.4}$	$23.7_{\pm 1.2}$	$25.6_{\pm 1.2}$	$31.4_{\pm 1.3}$	$32.3_{\pm 1.4}$	$22.4_{\pm 1.2}$	$24.1_{\pm 1.2}$
Qwen-3-VL-Instruct	$27.0_{\pm 1.2}$	$28.3_{\pm 1.2}$	$20.2_{\pm 1.0}$	$23.0_{\pm 1.1}$	$27.0_{\pm 1.2}$	$26.7_{\pm 1.2}$	$18.8_{\pm 1.1}$	$19.5_{\pm 1.1}$
Llama-4-Scout	$19.5_{\pm 1.0}$	$24.4_{\pm 1.2}$	$16.3_{\pm 0.8}$	$18.6_{\pm 1.0}$	$20.2_{\pm 1.0}$	$19.7_{\pm 1.0}$	$14.8_{\pm 0.8}$	$14.2_{\pm 0.9}$

Table 3: CLAIR scores ($[1, 100]$, ↑; mean ± 95% CI) for next coach utterance generation. The default setting uses text-visual input, 30s context, and oracle prompt. Other columns report ablations over input modality, context window, and prompt condition. CLAIR scores are judged by GPT-4.1 over 1,427 sampled pairs per run.

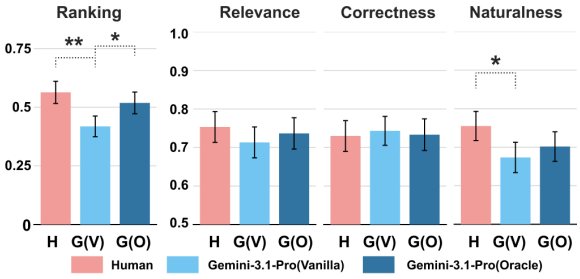


Figure 7: Mean normalized ranking (↑) and ratings (↑) for human (H), Gemini-3.1-Pro Vanilla prompt (G(V)), and Oracle prompt (G(O)) coaching utterances (* = $p < 0.05$, ** = $p < 0.01$, Friedman test followed by pairwise Wilcoxon signed-rank tests, Bonferroni correction).

whether the utterance sounded natural.

Results. Figure 7 shows human evaluators ranked human and Oracle conditions significantly higher than Vanilla ($p < .01$ and $p < .05$ respectively), suggesting that knowing the target coaching methods improves preference. Models scored comparably to humans on Relevance and Correctness, while human utterances were rated significantly more natural than Vanilla ($p < .05$), implying models generate locally plausible but unnatural utterances. Expert evaluators noted that effective coaching should be concise, accurate, and grounded in the learner's current screen state. Coaches should give direct instruction when learners are stuck, while encouraging independent thinking/using prior knowledge when possible.

7 How Do Models Interact with Humans?

We have shown that models speak and coach differently from human coaches. *But what are interaction dynamics and effectiveness when models coach novices to use software in real time?*

Method. We conducted an interactive evaluation with 10 participants from professional networks (P1–P10, 2 per software application) with the same setup as our data collection, yielding 36 model–human coaching sessions. To balance inference time and quality, learners completed tasks with guidance from Gemini-3-Flash with the Vanilla prompt and a 10s context window at 1 fps. Learners can interact with models via asking questions or receiving the model's proactive instructions every 20 seconds (See Appendix F.2 for system setup).

7.1 Learning Outcome

Both human ($N = 72$) and model ($N = 36$) coaching produced significant improvement in matched pre/post tasks ($p < .001$, Wilcoxon signed-rank test), but human coaching was substantially more effective (Table 4). Human coaching yielded a mean gain of 54.75% (33.49% → 88.24%), with 81.9% of sessions improving and none declining. Model coaching produced a smaller mean gain of 31.67% (13.33% → 45.00%), with only 58.3% of sessions improving, 25% showing no progress, and one declining. Models were far less effective than humans in teaching learners to use software independently. Appendix H reports subjective ratings, pre/post task outcomes, and qualitative feedback.

7.2 Language Use

The 36 model-human coaching sessions comprise 3,241 turns. Compared to human coaching sessions, models use more tokens per utterance, extensively use *Action Directives* (63% vs. 37%), and rarely use *Inform* (14% vs. 29%) that build learner understanding and *Info Request* (2% vs. 7%) to check-in learning state. Human learners, in turn, ask more questions (29% vs. 13%) and provide more information (20% vs. 9%), primarily due to the lack of proactivity of models (Figure 8).

Progress	Setup	Tutorial	Pre-Task	Post-Task
100%	H	72 (100.0%)	13 (18.1%)	49 (68.1%)
	M	9 (25.0%)	2 (5.6%)	9 (25.0%)
50–100%	H	0 (0.0%)	7 (9.7%)	15 (20.8%)
	M	7 (19.4%)	2 (5.6%)	6 (16.7%)
0–50%	H	0 (0.0%)	22 (30.6%)	8 (11.1%)
	M	15 (41.7%)	8 (22.2%)	12 (33.3%)
0%	H	0 (0.0%)	30 (41.7%)	0 (0.0%)
	M	5 (13.9%)	24 (66.7%)	9 (25.0%)
Total	H	72 (100.0%)	72 (100.0%)	72 (100.0%)
	M	36 (100.0%)	36 (100.0%)	36 (100.0%)

Table 4: Progress outcomes in human (H; $N = 72$) and model (M; $N = 36$) coaching sessions.

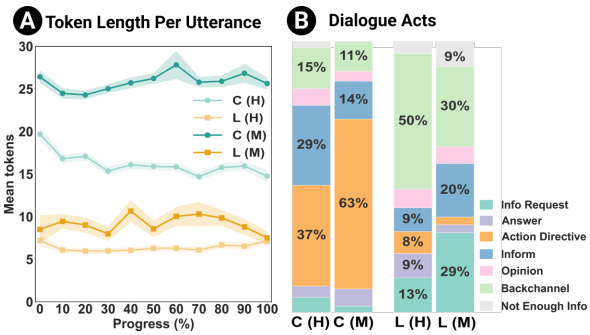


Figure 8: (A) Token length per utterance and (B) distribution of dialogue acts of coaches (C) and learners (L) in human (H; $N = 72$) and model (M; $N = 36$) coaching sessions.

7.3 Failure Case Analysis

Figure 9 shows representative failure cases in model coaching sessions.

Communication Gap. Models by default generated step-by-step instructions. While learners could replicate operations, they did not understand how or why (P4, P7, P9). Learners were still confused about the high-level plan and how to think about completing the task even after successful tutorials, leading to worse outcomes. Six participants found models too verbose and terminology-heavy, and novices found it hard to follow multiple steps and locate the relevant tools hidden in layers of menus (Figure 9 A). P3 observed that they can follow simple steps at the beginning, but guidance became increasingly confusing as the task state became more complex. Thus, they wanted slower and more concise guidance on where and how to find the target tool. P8 also suggested that, like a human coach, models could point or describe nearby icons to help learners find the right tool.

Grounding Gap. Models fell short in tracking learners' current screen state, which required learners to explicitly describe progress (P1, P2, P5, P6) or ask what to do next (P3, P9, P10). P1 reported that they have to say *"I'm done"* to make the model move on to the next step. This limitation of real-time multimodal understanding prevented the model from detecting and recovering when learners were stuck or had gone off track. For example, P2 was told to click a button that did not exist, yet the model kept repeating the same instruction. Similarly, P5 could not find where to create a pivot sheet, but the model repeated the same long instruction rather than trying a different approach. Models also did not point out mistakes unless explicitly asked. Even when they did respond, models sometimes hallucinated guidance based on misleading utterances rather than the screen state (Figure 9 B).

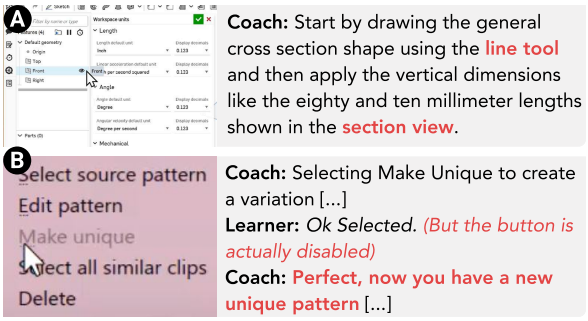


Figure 9: Representative (A) communication and (B) grounding failures in interactive evaluation.

8 Discussion and Conclusion

We introduce DIGITALCOACH, the first multimodal dataset of human expert-novice computer use coaching sessions. Our analysis reveals that current state-of-the-art multimodal models exhibit significant communication and grounding gaps compared to human coaches: they more frequently use direct instruction, and fail to adapt to the learner's evolving screen state. In an interactive evaluation, these gaps directly result in worse learning outcomes, with model-coached learners completing fewer milestones and retaining fewer skills.

Agency-Preserving Language Technology. Effective coaching balances direct instruction with learner agency. Human experts combine procedural guidance with explanation to build independence rather than replacing learner agency. For surface-level friction such as locating a menu item, direct instruction reduces unnecessary cognitive load, while for conceptual tasks such as design, agents should instead encourage reflection.

Proactive and Grounded Multimodal Agents.

Our findings suggest that effective coaching agents must go beyond reactive question answering. Agents should proactively monitor learner progress and select pedagogical interventions accordingly. This requires tighter multimodal grounding, including real-time screen understanding and spatially situated feedback such as screen annotations to support the coaching that humans provide naturally.

Acknowledgments

This research was supported by an Amazon Research Award, a Google gift, a Google ML and Systems Junior Faculty Award, a Technical AI Safety Research award from Coefficient Giving, and an NVIDIA Academic Grant Program award. We also thank our study participants for their time and valuable contribution to this work.

Limitations

Our dataset has several limitations. First, all sessions were collected on a Windows laptop to standardize the environment and reflect industry standards (*e.g.,* Blender is more commonly used on Windows). Future research could include multiple operating systems to capture diverse UI layouts and interaction behaviors. Second, although DIGITALCOACH captures 28.1 hours of coaching sessions across diverse software domains and participants, future work could expand the dataset with more sessions, more diverse software applications, and a broader range of novice and expert participants. Third, spoken dialogue often contains pauses, repairs, partial phrases, and overlaps. As a result, utterance segmentation and annotation can be ambiguous. We mitigate this by using explicit segmentation and annotation guidelines. Future work could explore alternative segmentation and annotation methods on our raw data. Finally, evaluating computer use coaching remains an open challenge. Task completion alone does not fully capture learning. While our pre/post tasks provide an effective measure of learning, they cannot fully capture skill retention over time. Future benchmarks can therefore evaluate coaching along multiple dimensions (*e.g.,* agency, task milestones, interaction traces, and pre/post tasks) and multiple domains (*e.g.,* open-ended creative tasks). DIGITALCOACH opens directions for building gym-style benchmarks for computer use coaching agents.

Ethical Considerations

Human Data Collection and Evaluation Our human evaluation study was approved by our institution's Institutional Review Board (IRB). In our study, we ensured that all participants were compensated fairly for their time and contributions. The payment was determined based on the average market rate for such studies, participant expertise, and the complexity and duration of the tasks ($20 / hour for learners and human evaluators; $20 – 60 / hour for coaches, depending on their experiences).

Hallucinations in Language Models Our work also uses LLMs to annotate dialogue acts and coaching methods. While we have shown that these approaches better align with humans, LLMs are not free from potential hallucinations and can lead to inaccurate results.

Automation vs. Augmentation. While LLM agents are increasingly framed as replacements for human labor, our work points to a different role for agents: not automating tasks, but supporting humans in developing new skills. However, this framing also requires caution. For example, coaching agents could be deployed as substitutes for human experts and thus reduce access to human mentorship and weaken the social and emotional support from human coaches. We therefore view coaching agents as a complement to, rather than a replacement for, human expertise.

Trust in Coaching Agents. Learners may place high trust in computer use coaching agents, especially when they lack the expertise to judge if its guidance is correct. Incorrect or overconfident instructions can mislead learners and reinforce misconceptions. We are aware that future design of computer use coaching agents should communicate uncertainty, support verification, and make it easy for learners to question or override their guidance.

References

Lavisha Aggarwal, Vikas Bahirwani, Lin Li, and Andrea Colaco. 2025. Generating dialogues from egocentric instructional videos for task assistance: Dataset, method and benchmark.

Yongsu Ahn, Lejun R. Liao, Benjamin Bach, and Nam Wook Kim. 2026. From Answer Givers to Design Mentors: Guiding LLMs with the Cognitive Apprenticeship Model.

Danial Alihosseini, Ehsan Montahaei, and Mahdieh Soleymani Baghshah. 2019. Jointly measuring diversity and quality in text generation models. In *Proceedings*

of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation, pages 90–98, Minneapolis, Minnesota. Association for Computational Linguistics.

James Allen and George Ferguson. 2002. Human-machine collaborative planning. In *Proceedings of the Third International NASA Workshop on Planning and Scheduling for Space*, pages 27–29.

James F. Allen, Lenhart K. Schubert, George Ferguson, Peter Heeman, Chung Hee Hwang, Tsuneaki Kato, Marc Light, Nathaniel Martin, Bradford Miller, Massimo Poesio, and David R. Traum. 1995. The TRAINS project: a case study in building a conversational planning agent. *Journal of Experimental & Theoretical Artificial Intelligence*, 7(1):7–48. _eprint: https://doi.org/10.1080/09528139508953799.

Anthropic. 2026. Claude 4.6. Accessed: Jun 12, 2026.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025. Qwen3-VL Technical Report.

Cristian-Paul Bara, Sky CH-Wang, and Joyce Chai. 2021. MindCraft: Theory of mind modeling for situated dialogue in collaborative tasks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1112–1125, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Apratim Bhattacharyya, Bicheng Xu, Sanjay Haresh, Reza Pourreza, Litian Liu, Sunny Panchal, Leonid Sigal, and Roland Memisevic. 2025. Can multi-modal llms provide live step-by-step task guidance? *Advances in Neural Information Processing Systems*, 38:22377–22410.

David Chan, Suzanne Petryk, Joseph Gonzalez, Trevor Darrell, and John Canny. 2023. CLAIR: Evaluating image captions with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13638–13646, Singapore. Association for Computational Linguistics.

David M Chan, Austin Myers, Sudheendra Vijayanarasimhan, David A Ross, Bryan Seybold, and John F Canny. 2022. What's in a caption? dataset-specific linguistic diversity and its effect on visual description models and metrics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4740–4749.

Michael Clemens and Ana Marasović. 2025. MixAssist: An Audio-Language Dataset for Co-Creative AI Assistance in Music Mixing.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Allan Collins and Manu Kapur. 2022. Cognitive Apprenticeship. In R. Keith Sawyer, editor, *The Cambridge Handbook of the Learning Sciences*, 3 edition, Cambridge Handbooks in Psychology, pages 156–174. Cambridge University Press, Cambridge.

Mark G Core and James Allen. 1997. Coding dialogs with the damsl annotation scheme. In *AAAI fall symposium on communicative action in humans and machines*, volume 56, pages 28–35. Boston, MA.

Joseph L. Fleiss and Jacob Cohen. 1973. The Equivalence of Weighted Kappa and the Intraclass Correlation Coefficient as Measures of Reliability. *Educational and Psychological Measurement*, 33(3):613–619.

Dan Friedman and Adji Bousso Dieng. 2022. The Vendi Score: A Diversity Evaluation Metric for Machine Learning.

Difei Gao, Lei Ji, Zechen Bai, Mingyu Ouyang, Peiran Li, Dongxing Mao, Qinchen Wu, Weichen Zhang, Peiyi Wang, Xiangwu Guo, Hengxu Wang, Luowei Zhou, and Mike Zheng Shou. 2024a. Assistgui: Task-oriented PC graphical user interface automation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 13289–13298. IEEE.

Shanghua Gao, Ada Fang, Yepeng Huang, Valentina Giunchiglia, Ayush Noori, J. Schwarz, Yasha Ektefaie, Jovana Kondic, and M. Zitnik. 2024b. Empowering biomedical discovery with ai agents. *Cell*, 187 22:6125–6151.

Milica Gašić, Catherine Breslin, Matthew Henderson, Dongho Kim, Martin Szummer, Blaise Thomson, Pirros Tsiakoulis, and Steve Young. 2013. POMDP-based dialogue manager adaptation to extended domains. In *Proceedings of the SIGDIAL 2013 Conference*, pages 214–222, Metz, France. Association for Computational Linguistics.

Frederic Gmeiner, Kenneth Holstein, and Nikolas Martelaro. 2025. Prototyping Multimodal GenAI Real-Time Agents with Counterfactual Replays and Hybrid Wizard-of-Oz.

Google. 2026. Gemini 3.1 Pro.

Yanzhu Guo, Guokan Shang, and Chloé Clavel. 2024. Benchmarking Linguistic Diversity of Large Language Models.

Mattias Heldner and Jens Edlund. 2010. Pauses, gaps and overlaps in conversations. *Journal of Phonetics*, 38(4):555–568.

Yilun Hua and Yoav Artzi. 2024. Talk Less, Interact Better: Evaluating In-context Conversational Adaptation in Multimodal LLMs. *arXiv preprint*. ArXiv:2408.01417 [cs.CL].

Mina Huh, Zihui Xue, Ujjaini Das, Kumar Ashutosh, Kristen Grauman, and Amy Pavel. 2025. Vid2Coach: Transforming How-To Videos into Task Assistants.

Faria Huq, Zora Zhiruo Wang, Frank F. Xu, Tianyue Ou, Shuyan Zhou, Jeffrey P. Bigham, and Graham Neubig. 2025. CowPilot: A Framework for Autonomous and Human-Agent Collaborative Web Navigation.

Lawrence Jang, Yinheng Li, Dan Zhao, Charles Ding, Justin Lin, Paul Pu Liang, Rogerio Bonatti, and Kazuhito Koishida. 2024. VideoWebArena: Evaluating Long Context Multimodal Agents with Video Understanding Web Tasks.

Jin-Hwa Kim, Nikita Kitaev, Xinlei Chen, Marcus Rohrbach, Byoung-Tak Zhang, Yuandong Tian, Dhruv Batra, and Devi Parikh. 2019. CoDraw: Collaborative drawing as a testbed for grounded goal-driven communication. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6495–6513, Florence, Italy. Association for Computational Linguistics.

Tomer Levinboim, Ashish V. Thapliyal, Piyush Sharma, and Radu Soricut. 2021. Quality estimation for image captions based on large-scale human evaluations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3157–3166, Online. Association for Computational Linguistics.

Jessy Lin, Nicholas Tomlin, Jacob Andreas, and Jason Eisner. 2024a. Decision-Oriented Dialogue for Human-AI Collaboration. *Transactions of the Association for Computational Linguistics*, 12:892–911.

Kevin Qinghong Lin, Linjie Li, Difei Gao, Qinchen Wu, Mingyi Yan, Zhengyuan Yang, Lijuan Wang, and Mike Zheng Shou. 2024b. Videogui: A benchmark for GUI automation from instructional videos. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Jakub Macina, Nico Daheim, Sankalan Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023. MathDial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5602–5621, Singapore. Association for Computational Linguistics.

Jakub Macina, Nico Daheim, Ido Hakimi, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2025. MathTutorBench: A Benchmark for Measuring Open-ended Pedagogical Capabilities of LLM Tutors.

Brandon Man, Ghadi Nehme, Md Ferdous Alam, and Faez Ahmed. 2025. VideoCAD: A Dataset and Model for Learning Long-Horizon 3D CAD UI Interactions from Video.

Meta. 2025. The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation.

OpenAI. 2026. Introducing GPT-5.4.

Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gökhan Tür, and Dilek Hakkani-Tür. 2022. Teach: Task-driven embodied agents that chat. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 2017–2025. AAAI Press.

Yi-Hao Peng, Dingzeyu Li, Jeffrey P Bigham, and Amy Pavel. 2025. Morae: Proactively Pausing UI Agents for User Choices. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, UIST '25, pages 1–14, New York, NY, USA. Association for Computing Machinery.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaïd Harchaoui. 2021. MAUVE: measuring the gap between neural text and human text using divergence frontiers. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 4816–4828.

Omar Shaikh, Kristina Gligorić, Ashna Khetan, Matthias Gerstgrasser, Diyi Yang, and Dan Jurafsky. 2024. Grounding gaps in language model generations. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6279–6296.

Mayank Sharma, Roy Pea, and Hari Subramonyam. 2026. ConvoLearn: A Dataset of Constructivist Tutor-Student Dialogue.

Shannon Zejiang Shen, Valerie Chen, Ken Gu, Alexis Ross, Zixian Ma, Jillian Ross, Alex Gu, Chenglei Si, Wayne Chi, Andi Peng, Jocelyn J. Shen, Ameet Talwalkar, Tongshuang Wu, and David Sontag. 2025. Completion $\neq$ Collaboration: Scaling Collaborative Effort with Agents. *arXiv preprint*. ArXiv:2510.25744 [cs.CL].

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–374.

Alane Suhr, Claudia Yan, Jack Schluger, Stanley Yu, Hadi Khader, Marwa Mouallem, Iris Zhang, and Yoav Artzi. 2019. Executing instructions in situated collaborative interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2119–2130, Hong Kong, China. Association for Computational Linguistics.

Nicholas Tomlin, Naitian Zhou, Eve Fleisig, Liangyuan Chen, Téa Wright, Lauren Vinh, Laura X. Ma, Seun Eisape, Ellie French, Tingting Du, Tianjiao Zhang, Alexander Koller, and Alane Suhr. 2025. Characterizing Language Use in a Collaborative Situated Game.

Takuma Udagawa and Akiko Aizawa. 2020. An annotated corpus of reference resolution for interpreting common grounding. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9081–9089. AAAI Press.

Zhenyu Wang, Aoxue Li, Zhenguo Li, and Xihui Liu. 2024. Genartist: Multimodal LLM as an agent for unified image generation and editing. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Zora Zhiruo Wang, Yijia Shao, Omar Shaikh, Daniel Fried, Graham Neubig, and Diyi Yang. 2025. How Do AI Agents Do Human Work? Comparing AI and Human Workflows Across Diverse Occupations.

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. 2024. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Saelyne Yang, Sangkyung Kwak, Juhoon Lee, and Juho Kim. 2023. Beyond instructions: A taxonomy of information types in how-to videos. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, CHI 2023, Hamburg, Germany, April 23-28, 2023*, pages 797:1–797:21. ACM.

Saelyne Yang, Jaesang Yu, Yi-Hao Peng, Kevin Qinghong Lin, Jae Won Cho, Yale Song, and Juho Kim. 2026. GUIDE: A Benchmark for Understanding and Assisting Users in Open-Ended GUI Tasks.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik R. Narasimhan. 2024. {τ}-bench: A Benchmark for \underline{T}ool-\underline{A}gent-\underline{U}ser Interaction in Real-World Domains. In *Proceedings of the ICLR 2025 Conference*.

Suyu Ye, Haojun Shi, Darren Shih, Hyokun Yun, Tanya Roosta, and Tianmin Shu. 2025. RealWebAssist: A Benchmark for Long-Horizon Web Assistance with Real-World Users.

Shaofeng Yin, Jiaxin Ge, Zora Zhiruo Wang, Xiuyu Li, Michael J. Black, Trevor Darrell, Angjoo Kanazawa, and Haiwen Feng. 2026. Vision-as-Inverse-Graphics Agent via Interleaved Multimodal Reasoning.

Yichi Zhang, Xin Luna Dong, Zhaojiang Lin, Andrea Madotto, Anuj Kumar, Babak Damavandi, Joyce Chai, and Seungwhan Moon. 2025. Proactive Assistant Dialogue Generation from Streaming Egocentric Videos. *ArXiv preprint*, abs/2506.05904.

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. 2024. Webarena: A realistic web environment for building autonomous agents. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Appendix

Our appendix is organized as follows:

- Appendix A provides detailed statistics and linguistic examples of the dataset.
- Appendix B outlines the experimental platform setup and facilitation scripts.
- Appendix C catalogs the specific software tasks across the five tested domains.
- Appendix D details the human-in-the-loop transcript processing workflow.
- Appendix E defines the dialogue act and coaching methods codebooks.
- Appendix F details model evaluation baselines and metrics.
- Appendix G provides the full text prompts utilized for LLM pipelines.
- Appendix H provides quantitative and qualitative human learning outcomes.
- Appendix I discusses the license of produced artifacts.
- Appendix J provides information on settings for used code packages.
- Appendix K states our AI Use Disclosure.

A Dataset Details

To support further research, the DIGITALCOACH dataset, including the screen recording, audio transcripts, mouse and keyboard events, and the file snapshots, will be made publicly available upon publication. The DIGITALCOACH dataset will be accessible on Hugging Face and the evaluation code on GitHub.

In this section, we provide detailed information about DIGITALCOACH and several examples of linguistic phenomena. Table A.1 shows detailed statistics of DIGITALCOACH. Figure A.1 shows the distribution of number of utterances across all sessions. Figure A.2 shows the distribution of duration across all sessions. Figure E.5 shows the distribution of coaching methods across the conversation progress of all 72 sessions.

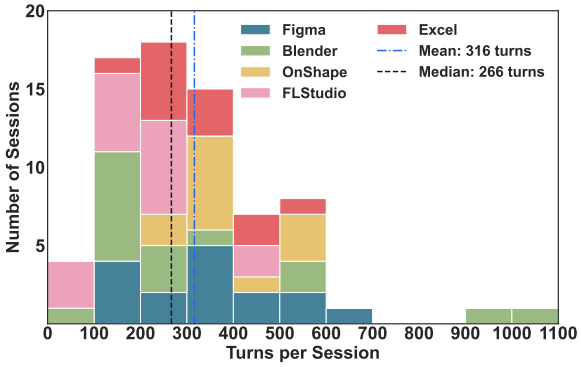


Figure A.1: Distribution of number of utterances across 72 sessions. Each bar shows the number of recorded coaching sessions within a duration range.

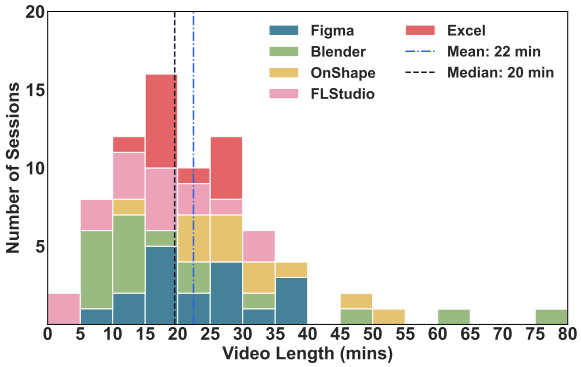


Figure A.2: Distribution of video durations across 72 sessions. Each bar shows the number of recorded coaching sessions within a duration range.

A.1 Additional Language Statistics

We observed linguistic phenomena in DIGITALCOACH. Token frequency (Figure A.3) follows a Zipfian distribution. Token length (Figure A.4) and gap length((Figure A.5) follow exponential distributions, with fewer tokens and smaller gaps being most common.

A.2 Example Linguistic Phenomena

In this section we provide examples of the linguistic phenomena observed in our coaching sessions.

Convention Formation. 0104/C2L2:

- Turn 386 (Coach): "Alright, now we want to rename this to. . . *Card*."
- Turn 387 (Learner): "*Card*."
- Turn 388 (Coach): "Mhm. And let's make *card* have a corner radius of 8."

Spatial Referencing. 0104/C2L2:

- Turn 60 (Coach): "Yep, and then just rotate it, um, by dragging on *this corner right here*. . . "

Statistic	Value
# Software Applications	5
# Participants	40
# Sessions	72
Video Duration	28.1h
Avg. Duration	22.5 min
Min Duration	3 min 53 sec
Max Duration	1 hour 16 min 16 sec
# Total Turns	22,752
% Coach Turns	65.81%
% Learner Turns	34.19%
Avg. Turns per Session	303.36
Avg. Turn Tokens	9.69
# Input Events	39,609
# File Snapshots	36,724

Table A.1: Summary statistics of DIGITALCOACH.

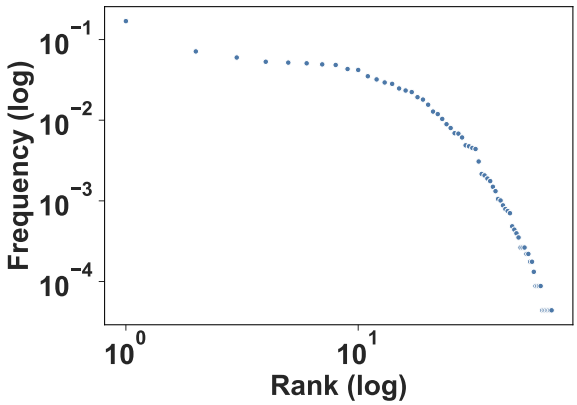


Figure A.3: Token frequencies in DIGITALCOACH follow a Zipfian distribution.

- Turn 62 (Coach): "Just hover over *this part*."
- Turn 63 (Learner): "Hover *in*?"
- Turn 64 (Coach): "Hover over, like, *slightly outside of it*, like, *where my pink is*."
- Turn 65 (Learner): "Ah, slightly— okay."

Repair. 0104/C2L2:

- Turn 84 (Coach): "Yep, now we have a, um, like, *component* for this play button."
- Turn 85 (Coach): "Alright, so now we want to create, like, our album art frame, so…"
- Turn 86 (Learner): "*Wait*, well, component means… uh, *what again, exactly*? It's interactable with…"
- Turns 87–90 (Coach): "So, components are, like, the parent, um, designs… [extended explanation] …That's a variant of this parent component."

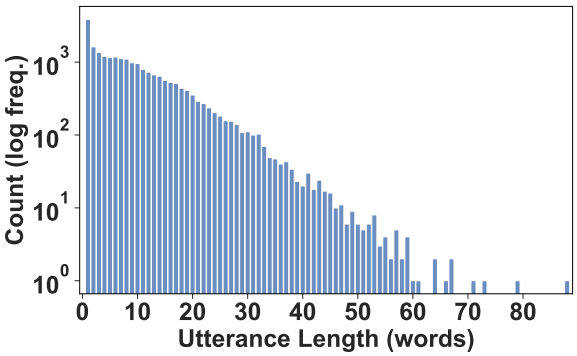


Figure A.4: Token length (number of words per utterance) follows an exponential distribution.

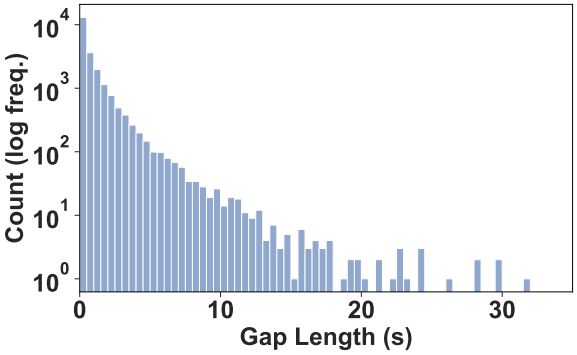


Figure A.5: Distribution of gap length follows an exponential distribution. We define the gap as the duration of silence between the end of one speaker and the beginning of the next speaker (Heldner and Edlund, 2010).

- Turn 91 (Learner): "*Okay, okay, got it, got it*."

B Study Protocol

Each study session involved one expert coach, one novice learner, and one researcher facilitator. Sessions lasted approximately 3 hours and consisted of consent and setup, followed by 3–4 task blocks. Each task block included a pre-task, a tutorial task, and a post-task.

B.1 Device Setup

The coach and learner communicated remotely via Zoom throughout the session. Learners conducted all software activities on a study laptop (i.e., Lenovo E14 Gen 6) and use research accounts rather than their personal account to ensure privacy.

We used Zoom as the communication platform and the source of synchronized audio and screen recordings of the session. We recorded at 25 frames per second with 1920×1128 resolution. The audio was recorded from the study laptop microphone in mono channel with a 48K sampling rate.

We used Workflow Induction Toolkit (Wang

et al., 2025) to log mouse and keyboard actions on the study laptop. In addition, we implemented software-specific file loggers to save file snapshots periodically (i.e., every 10s), allowing us to reconstruct intermediate work states over time.

B.2 Study Script

One facilitator was present at each session. They were provided with the following script to facilitate the session.

B.2.1 Before Study

- *[Set up the Zoom meeting]*
- *[Rename the participants as Coach, Learner, and Facilitator]*
- *[Share the study laptop screen with only the software window visible]*
- *[Hide the Zoom video pane and toolbar (Ctrl + Shift + Alt + H)]*

B.2.2 Introduction and Consent

Thank you both for joining today. I am *[facilitator name]*, and I will be the session facilitator.

In this study, you will work together to complete a set of activities using *[software name]*. Today, *[coach name]* will act as the coach, and *[learner name]* will act as the learner. The goal of the study is to understand how experts provide novices with instructions to learn *[software name]*.

During the session, we will record the learner's screen, both participants' audio, and interaction logs from the learner's computer. I am now sending you the consent form and media release form. By signing these forms, you confirm that you understand the study procedures, that your questions have been answered, and that you agree to participate. *[Send consent forms.]*

We also have this survey we would like you to fill out that asks about your experiences with *[software name]*. Please fill it out and let us know when you've completed it. Let us know if you have any questions about it. *[Send pre-study survey.]*

[coach name], can you see the green pencil icon at the bottom left of your screen? That is the Zoom annotation tool. Set the annotation color to **pink** and the line thickness to **thick**. Please use it to briefly draw on the learner's screen.

Next, *[learner name]*, please grant remote control access to the coach. *[coach name]* may request remote control if needed during the coaching activity. *[Ask the coach to test remote control by moving the mouse and typing briefly.]*

If any technical issues arise during the study, please let me know through Zoom chat. Otherwise, I will remain silent during the task activities and will only intervene if necessary.

B.2.3 Task Block

Each task block lasted approximately 50 minutes and consisted of three phases:

- Pre-task (up to 15 minutes)
- Tutorial task (10–60 minutes)
- Post-task (up to 15 minutes)

Pre-Task We will now begin the pre-task. In this task, *[learner name]* will work on the task independently based on the task description. *[coach name]* should remain silent. *[learner name]*, you are encouraged to think aloud while working. You have up to 15 minutes to finish the task but you can stop at any time you want. *[Share pre-task document and starter file.] [Start a 15-minute timer.]*

Tutorial Task We will now begin the tutorial task. There are no restrictions on how the coach and the learner should communicate. *[Share tutorial task document and starter file.]*

Post-Task We will now begin the post-task. This task is the same as the pre-task. The learner will have 15 minutes to work on the task independently based on the task description. The coach should not provide instruction unless the learner appears to be stuck for 3 minutes, in which case the coach may provide at most one hint. *[Share post-task document and starter file.] [Start a 15-minute timer.]*

Actions Before Each Task

- *[Start file logger.]*
- *[Start Workflow Induction Toolkit recorder.]*
- *[Start Zoom recording.]*

Actions After Each Task

- *[Stop Zoom recording.]*
- *[Stop Workflow Induction Toolkit recorder.]*
- *[Stop file logger.]*
- *[Check that data have been saved correctly.]*

C Tasks

Table C.1 lists all tasks used in DIGITALCOACH.

D Transcription

We use the automatically generated closed caption (i.e., `.vtt` file) from Zoom as a starting point as it includes accurate timestamps, speaker labels (coach vs. learner), and an initial segmentation of the dialogue.

D.1 Transcript Correction

We use an LLM (i.e., Gemini 3.1 Pro) to correct wrong words, homophone errors, dropped or extra words, or other mismatches with what is actually said. Prompt used for transcript correction can be found in Figure G.1. One author then manually validate the transcript to correct the speaker label and utterances. We use following notation to indicate pauses and interruptions:

- Use `comma` (`,`) to indicate a short speech pause or hesitation.
- Use `ellipsis` (`...`) to indicate a longer pause or hesitation.
- Use `double dash` (`--`) to indicate trailing off or interruption by another speaker.
- Use `parentheses` (`( )`) to indicate an uncertain best-guess interpretation, e.g., `(move)`. If no reasonable guess is available, leave the parentheses empty.
- Use `parentheses` (`( )`) to replace any personal information, e.g., `(facilitator)`.

In general, we will not annotate intonational features and non-spoken noises.

D.2 Utterance Segmentation

Three authors split and merged the corrected utterances according to the following guideline:

1. Split the utterance if there is a pause of 2 seconds or longer.
2. Use changes in intonation and what can be inferred from the semantics to decide whether to split for shorter pauses.
   (a) Descending pitch can be a hint that an utterance is ending.
   (b) If it is difficult to assign different functional meanings to two adjacent spans, they should likely be merged into one utterance.
3. Context from the other speaker can also help (e.g., if one half of the utterance is responding directly to the other speaker, it may make sense to split).
4. Use the multimodal context to decide whether to split.
   (a) If a short pause coincides with a visible action in the screen state, this can be evidence for a new utterance.
   (b) If the screen state remains unchanged and there is only a pause shorter than 2 seconds, the utterances of the same speaker should generally be merged.

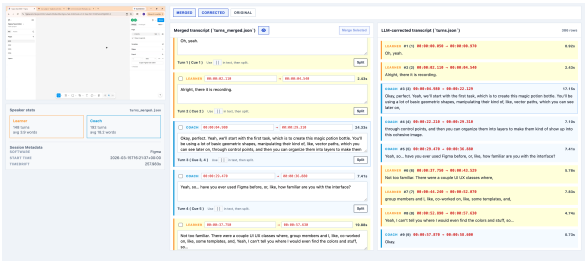


Figure D.1: Transcription correction interface. Editor can see the dialogue transcript alongside screen recording. They can use the interface to edit, split, merge each utterance and trim its start and end time.

D.3 Transcript Correction Interface

The transcript correction interface screenshot is shown in Figure D.1.

E Annotation

E.1 Human Annotation Method

To derive the taxonomy, three researchers used open coding on 200 turns to obtain potential coaching methods. Then, they met to merge together similar methods and create a codebook with a name, definition, and example for each coaching method. The researchers iteratively coded samples and revised the codebook to achieve final codes, containing 10 methods. During the coding process, we merged similar codes. Disagreements were resolved through discussion, and the finalized codebook was used for the annotations. Using the

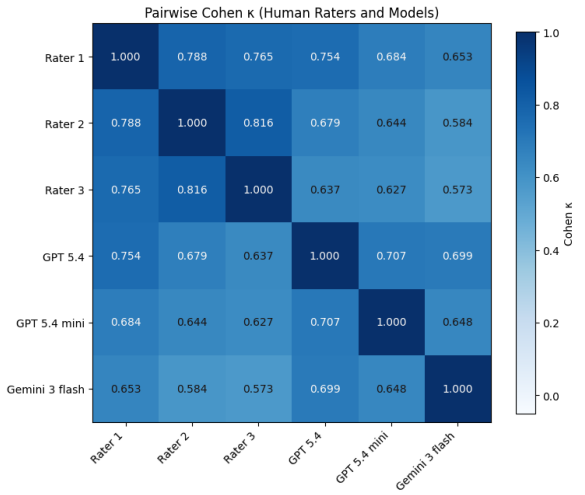


Figure E.1: Pairwise Cohen's κ agreement among three human raters and three model annotators for dialogue act annotation. Rows and columns denote annotators, and each cell reports the pairwise agreement score. GPT-5.4 aligns with human rater the best.

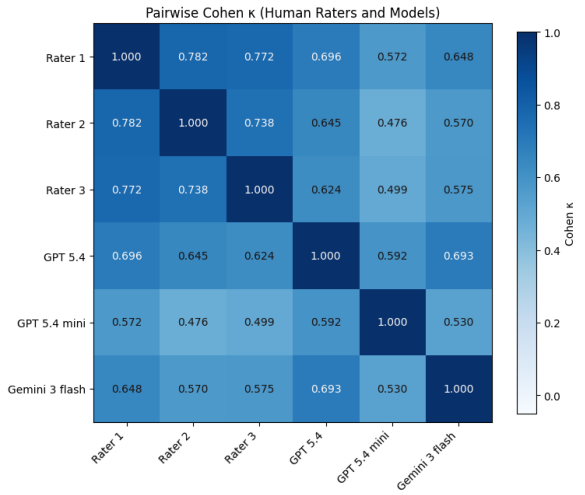


Figure E.2: Pairwise Cohen's κ agreement among three human raters and three model annotators for coaching method annotation. Rows and columns denote annotators, and each cell reports the pairwise agreement score. GPT-5.4 aligns with human rater the best.

codebook with examples, three researchers annotated other 200 utterances and reached a substantial agreement for coaching methods.

E.2 Dialogue Acts Schema

The full dialogue acts definitions and examples provided to annotators are shown in Table E.1.

E.3 Coaching Methods Schema

The full coaching methods definitions and examples provided to annotators are shown in Table E.2.

E.4 Additional Annotation Results

Figure E.3 shows the distribution of dialogue acts of coaches across conversation progress. Figure E.4 shows the distribution of dialogue acts of learners across conversation progress. Figure E.5 shows the distribution of coaching methods across conversation progress. Figure E.6 shows top-10 most frequent coaching method bigrams..

E.5 Annotation Interface

The annotation interface screenshot is shown in Figure E.7.

F Additional Evaluation Results

Table F.1 shows Jensen-Shannon divergence (JSD) for dialogue-act and coaching-method n-gram distributions. Table F.2 shows BLEU results for next coach utterance generation. Table F.4 shows METEOR results for next coach utterance generation.

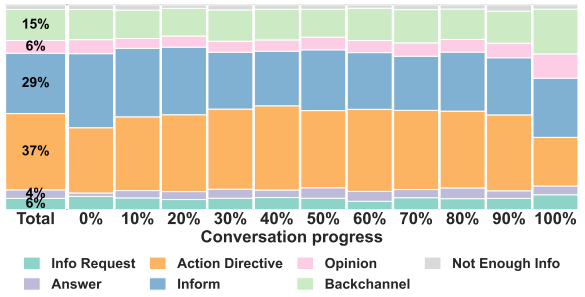


Figure E.3: Distribution of dialogue acts of coaches across conversation progress. Method use remains relatively stable as the conversation progresses, except for the beginning and the end 10%.

Table F.3 shows ROUGE-L results for next coach utterance generation. Table F.5 shows BERTScore results for next coach utterance generation. Table F.6 shows CLAIR scores (Default) split by software application. Table F.7 shows CLAIR scores (Default) split by coaching method.

F.1 Human Expert Evaluation Setup

To ensure a fair comparison, all candidate utterances and prior chat messages are converted to audio via a AI text-to-speech voice. The human expert evaluation interface screenshot is shown in Figure E.7.

F.2 Interactive Evaluation Setup

Interactive evaluation system uses a browser client (React, Stream Video WebRTC) paired with a Python backend (Vision agents, FastAPI). A novice shares screen and audio, and the software coach

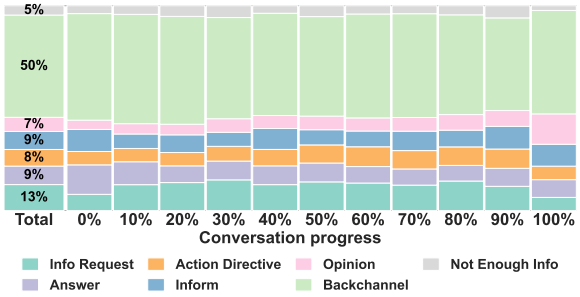


Figure E.4: Distribution of dialogue acts of learners across conversation progress. Dialogue acts remain relatively stable as the conversation progresses, except for the beginning and the end 10%.

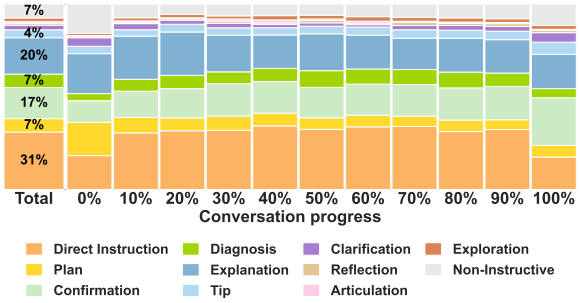


Figure E.5: Distribution of coaching methods across conversation progress. Method use remains relatively stable as the conversation progresses, except for the beginning and the end 10%.

agent joins the same call server-side (Figure F.2). By default, we route model inference through OpenRouter, Deepgram speech-to-text, and ElevenLabs text-to-speech. Proactive check-ins fire every 20s. Sessions log model–learner transcripts for offline analysis.

G Prompts

Figure G.1 shows the automatic transcript correction prompt. Figure G.2 shows the automatic dialogue act annotation prompt. Figure G.3 shows the automatic coaching method annotation prompt. Figure G.4 shows the next coach utterance generation prompt.

H Additional Human Coaching Results

H.1 Problematic Outcome Tags

Table H.1 shows problematic outcome tags for human and model coaching sessions.

H.2 Pre/Post Task Outcome Examples

Table H.2 shows examples of pre-task and post-task outcomes in visual tasks (Figma, Blender, OnShape).

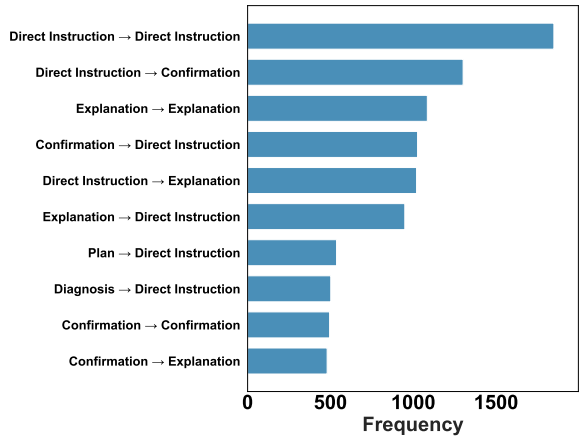


Figure E.6: Top-10 most frequent coaching method bigrams.

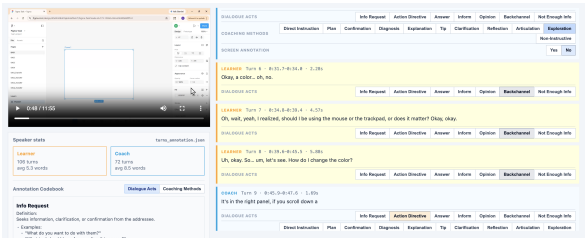


Figure E.7: Annotation interface for utterance labeling. Annotators can see the dialogue transcript alongside screen recording and assign tags to each utterance.

H.3 Subjective Learning Outcome

Learners reported substantially higher confidence after completing the tasks. Averaging confidence (7-scale likert score, 7 = very confident) increased from $\mu_{\text{pre}} = 2.49, \sigma_{\text{pre}} = 1.03$ to $\mu_{\text{post}} = 5.90, \sigma_{\text{post}} = 0.90$, with an average improvement of $\mu_{\text{diff}} = 3.41, \sigma_{\text{diff}} = 1.04$. A non-parametric Wilcoxon signed-rank test showed the increase is statistical significant $W = 0, p < .001$.

Coaches rated whether learners had acquired the relevant software skill for the task (7-scale likert score, 7 = the learner was fully equipped to complete similar tasks independently). Coach-rated skill increased from $\mu_{\text{pre}} = 2.88, \sigma_{\text{pre}} = 2.06$ to $\mu_{\text{post}} = 6.29, \sigma_{\text{post}} = 1.35$, with an average improvement of $\mu_{\text{diff}} = 3.42, \sigma_{\text{diff}} = 1.97$. A Wilcoxon signed-rank test showed the increase was statistically significant $W = 0, p < .001$. In total, 66 of 72 task instances received higher coach-rated skill scores after coaching, 6 stayed the same, and none decreased.

H.4 Qualitative Feedback

Across both learners (L1-L20) and coaches (C1-C20), participants described effective computer use

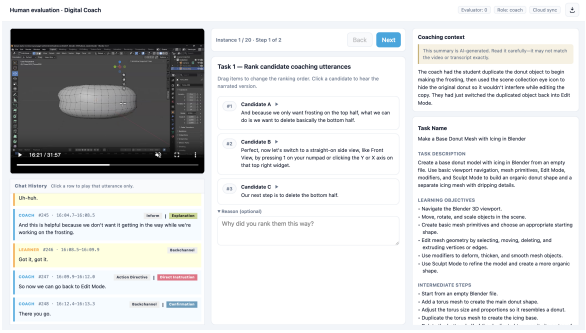


Figure F.1: Human expert evaluation interface. Raters can see the the previous conversation and screen context and listen to AI-narrated candidate coaching utterances.

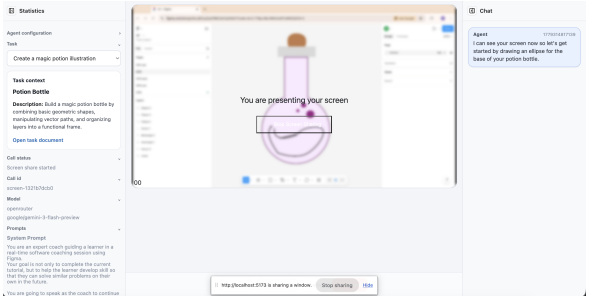


Figure F.2: Interactive evaluation interface. A novice shares their screen and speaks over a live call. Model coach observes recent frames and dialogue and replies with audio.

coaching an activity beyond simply giving step-by-step answers. It was a balance between procedural and conceptual guidance, guided practice, and learner agency.

Learners emphasized that practice was necessary for retention, especially after a dense coaching session. L10 noted that "more practice would be helpful to re-enforce the things I learnt," because they were unsure "for how long I will retain the knowledge if I am not practicing it regularly." Coaches echoed this point: C2 argued that "more practice with new tools would help with retention," especially when learners first observe an effect and then reproduce it themselves with less guidance. This suggests that an effective digital coach should not only demonstrate procedures, but also create repeated opportunities for learners to apply newly learned operations.

Participants also emphasized that direct coaching is useful when learners lack basic orientation or do not know where to begin. L8 stated, "I want direct teaching when I lack the basic knowledge of the software," while L20 found "detailed step-by-step" guidance helpful because each action was "illustrated in a digestible way." Coaches similarly noted that in unfamiliar interfaces, direct pointing can reduce unnecessary friction. C13 explained that in Figma, many functions are difficult to find without prior familiarity, so "it makes more sense to have those directly pointed out." However, participants also viewed direct instruction as only one part of effective teaching. L9 described a trade-off: direct answers are useful for "simple procedural questions," but for design-related or conceptual tasks, they preferred "guided exploration and explanation." Thus, digital coaches may need to dynamically shift between directive support and exploratory guidance depending on whether the learner's immediate barrier is procedural, conceptual, or creative.

Another strong theme was that learners valued explanations of concepts, rationales, and tool functions, not only instructions about where to click. L17 said the coach first explained "the function of a component" and then showed "how to operate," which strengthened both understanding and memory. C2 similarly argued that explanations and analogies help students understand "what specific numbers or functions do, instead of just clicking mindlessly through the steps." Learners particularly valued explanations that made knowledge transferable. L10 appreciated learning "strategies and thought processes" in CAD that could transfer to other tasks. L8 described how understanding the difference between "extrude vs. revolve" helped them narrow down tool choices in a later post-task. L9 similarly said effective moments focused not only on how to complete the task, but also on why a certain approach works, making the knowledge transferable.

Participants also highlighted the importance of guided exploration, reflection, and learner agency. L9 described effective teaching as "guidance that helped me think rather than directly giving the answer," using "minimal but meaningful cues" that allowed them to solve problems independently. C9 similarly described effective teaching as giving the learner "time to explore and think about the concepts" and then synthesize them individually. C20 preferred being able to ask learners questions rather than "just tell them what to do," describing coaching as "collaborative." Several learners valued moments where they tried, failed, and were corrected. L17 described "practicing, failing, learning what is actually correct, and then doing it again" as the best method because it forced them to think and understand what was wrong or right. L4 also found

Auto-Correct Transcription Prompt

```
You are aligning a reference transcript to the actual speech in the attached audio.

The JSON below is the reference transcript for this recording. Each object has timing, speaker
labels, and a "text" field.
The "text" values may contain wrong words, homophone errors, dropped or extra words, or other
mismatches with what is actually said.
Your task is to correct the "text" field to match what is actually said contextually.

Task:
1. Listen to the audio carefully.
2. For each object in array order, produce the wording that best matches what is spoken
  in the corresponding time range. Fix mistranscriptions only.
3. Do NOT invent new utterances, merge, or split segments. The number of strings you
  output must equal {n} (one corrected line per input object).
4. Do NOT change punctuation style unnecessarily; natural English is fine.
5. ONLY correct the "text" field; keep the same timing, speaker, and id.
6. Use , to indicate short speech pauses or hesitation.
7. Use ... to indicate a longer pause or hesitation.
8. Use - to indicate a trail off or being interrupted by another speaker.
9. The first letter of the first word in the "text" field should be capitalized.

Output requirements:
- Return ONLY valid JSON: a single array of {n} strings.
- String i is the corrected "text" for input object i (same order as the reference array).
- No markdown, no code fences, no keys other than the array itself.

Good Examples:
- "bit, there's pill." -> "bit, there's fill."
- "Cool. Mom. Gone." -> "Cool. Mmmmm. Done."
- "It does sometimes take a second, so baby... huh." ->
  "It does sometimes take a second, so maybe... huh."

Reference transcript (do not echo it back; only output the string array):

{reference_transcript}
```

Figure G.1: Prompt used for automatic transcript correction.

it most helpful when they tried to replicate what the coach taught, got stuck, and then received correction.

Visual grounding and screen annotation were repeatedly described as important for online computer use coaching. C12 said Zoom annotations were "incredibly helpful" because the coach could draw squares, arrows, and visual references without physically obstructing the screen. L13 said annotation made it easier to navigate the toolbar and find correct buttons, while L12 described screen annotations as "very effective in visually guiding me through the activity." C2 described a useful pattern: first explain the concept, then annotate where to click with arrows or circles.

I Artifact Licenses, Terms of Use, and Intended Use

Our work uses and creates several scientific artifacts: the DIGITALCOACH dataset, the software-task materials used to elicit coaching sessions, model outputs from the evaluated multimodal language models, and annotation/evaluation code. The DIGITALCOACH dataset is intended for research on situated, multimodal computer use coaching, human–AI collaboration, language grounding in graphical user interfaces, and pedagogical dialogue. It is not intended for use in systems that identify, profile, surveil, evaluate, or make consequential decisions about individual participants.

For artifacts created by this work, we will release only de-identified research data and accompanying documentation under a research-use license. The release will include derived and anonymized metadata, dialogue annotations, task metadata, model generations, and benchmark/evaluation scripts where permitted. Raw participant-identifying information, including names, contact information, and any uniquely identifying content observed in transcripts, screen recordings, or files, will not be released. All transcripts are manually reviewed for personally identifying information and

Dialogue Act Annotation Prompt

```
You are a linguistics expert annotating utterances from a task-oriented instructional dialogue.

Your job is to read each utterance from both Coach and Learner in context and decide what
dialogue act the speaker is trying to do.

Choose all applicable labels from the schema below. Use the full transcript for context.

Annotation Schema:
{{DIALOGUE_ACTS}}

Instructions:
For each turn, return a JSON array of objects with these fields:
- "turn_id": the turn number as an integer
- "text": the original text of the turn
- "dialogue_acts": an array of all applicable labels from the schema above

Output requirements:
- Return ONLY valid JSON as an array of objects.
- Do not include markdown fences.
- Do not include any extra text.

Transcript:

{{TRANSCRIPT}}
```

Figure G.2: Prompt used for automatic dialogue act annotation.

redacted before release. Access to any higher-risk modalities, such as screen recordings or file snapshots, will be restricted to research use and subject to the dataset terms of use.

For existing artifacts, including the software applications, tutorials, APIs, models, and evaluation packages used in this study, our use was limited to research and evaluation. We followed the applicable licenses, API terms, and platform policies for each artifact. We cite the creators of datasets, benchmarks, models, and methods used in the paper where applicable. The use of tutorial-derived task materials was limited to creating study tasks and was not used to redistribute the original tutorials themselves. Model outputs are used only for research evaluation of coaching behavior and are not presented as authoritative instructions for end users.

J Experimental Settings and Package Parameters

All model generations used the default settings of the corresponding model provider unless otherwise specified. We did not perform hyperparameter search over decoding parameters. Instead, we compared prompting conditions, input modalities, and context-window lengths as experimental factors. Specifically, we compared vanilla, coach, and oracle prompts; text-only, image-only, and text–visual context; and context windows of 1, 10, 30, and 60 seconds. The default automatic-evaluation setting used text–visual input, a 30-second context window, and the oracle prompt. The default interactive setting used Gemini-3-Flash, the vanilla prompt, a 10-second context window, and visual sampling at 1 frame per second.

For automatic evaluation, we used CLAIR, BLEU, METEOR, ROUGE-L, BERTScore, Self-BLEU, Vendi Score, MAUVE, Cohen's κ, Fleiss' κ, and F1 as reported in the paper. Unless otherwise stated, we used the default parameter settings from the corresponding public implementations. The exact package names, versions, and command-line arguments are provided in our released code and configuration files. We used GPT-4.1 as the judge model for CLAIR scoring in the automatic next-utterance evaluation.

Coaching Method Annotation Prompt

```
You are an expert annotator annotating coach utterances from a task-oriented instructional
dialogue.

Your job is to read each coach turn in context and decide which coaching method labels are used.

Choose all applicable labels from the schema below. Use the full transcript for context to
understand what the coach is responding to, but only annotate Coach turns.

Annotation Schema:
{{COACHING_METHODS}}

Instructions:
For each coach turn, return a JSON array of objects with these fields:
- "turn_id": the turn number as an integer
- "text": the original text of the turn
- "coaching_methods": an array of all applicable labels from the annotation classes

Output requirements:
- Return ONLY valid JSON as an array of objects.
- Do not annotate Learner turns.
- Do not include markdown fences.
- Do not include any extra text.

Transcript:

{{TRANSCRIPT}}
```

Figure G.3: Prompt used for automatic coaching method annotation.

Software	Task	Description	Source
01 **Figma**	01 Potion Bottle	Combine basic geometric shapes and manipulate vector paths to build a magic potion bottle organized in frames.	Official Tutorial
	02 Loading Animation	Create a looping loading animation using frames, ellipses, Smart Animate, and component variants for use in a mobile app.	Official Tutorial
	03 Responsive Card	Build a responsive podcast episode card using Auto Layout, constraints, image fills, and components that adapt to different sizes.	Official Tutorial
	04 Flower Vase	Illustrate a transparent glass vase with flowers by designing repeated petal shapes and placing curved stems inside the vase.	Official Tutorial
02 **Blender**	01 Donut Modeling	Create a donut mesh with icing using viewport navigation, Edit Mode, modifiers, and Sculpt Mode.	Online Tutorial
	02 Shading	Assign and preview separate materials for a donut and icing using Blender's Shading view.	Online Tutorial
	03 Dropping Animation	Animate an existing donut model to fall downward along the Z axis using keyframes.	Online Tutorial
	04 Sprinkles	Add procedurally generated sprinkles to an existing donut's icing in Blender.	Online Tutorial
03 **OnShape**	01 Saddle Bracket	Model a saddle bracket using sketching, extrusion, and semicircular cuts to match provided engineering dimensions.	Online Tutorial
	02 Geneva Cam	Model a Geneva cam using sketch geometry, circular repetition, extrusion, and feature patterning from an engineering drawing.	Online Tutorial
	03 Hub	Build a hub by revolving a base profile and adding repeated curved slots to match specified diameters and angular spacing.	Online Tutorial
04 **FL Studio**	01 Beat	Build a drum pattern with bass line using the Channel Rack and arrange patterns in the Playlist.	Official Tutorial
	02 Melody	Create and edit a melody in the Piano Roll, shaping note timing and dynamics before placing it in the Playlist.	Official Tutorial
	03 Song Track	Place, duplicate, and organize beat and melody patterns in the Playlist to build a fuller track structure.	Official Tutorial
	04 Mixer	Route instruments to Mixer inserts and apply effects such as reverb and delay using FL Studio's Mixer.	Official Tutorial
05 **Excel**	01 SUMIFS	Use SUMIFS to compute total revenue filtered by multiple conditions across a raw sales dataset.	Online Tutorial
	02 VLOOKUP	Link two tables with VLOOKUP to auto-fetch prices, categories, and tax rates for revenue calculations.	Online Tutorial
	03 Pivot Table	Summarize a dataset into multiple pivot views and charts to answer distinct business questions.	Online Tutorial

Table C.1: Overview of tasks used in the study.

Tag	Definition	Example
Info Request	Seeks information, clarification, or confirmation from the addressee.	*"What do you want to do with them?"*
Action Directive	Guides the addressee toward an action.	*"Click on the top menu."*
Answer	Provides information requested by a previous utterance.	*"Because the polygon is not closed."*
Inform	Provides objective information or describes the current state.	*"The tempo is set to 120 BPM."*
Opinion	Provides a subjective judgment, preference, or interpretation.	*"The alignment looks better now."*
Backchannel	Signals attention, understanding, agreement, or receipt.	*"Okay."*
Not Enough Info	Is too incomplete or ambiguous to classify reliably.	*"If you just—"*

Table E.1: Dialogue act schema used to annotate coach and learner utterances.

Tag	Definition	Example
Direct Instruction	Gives concrete procedural guidance about next action to take.	*"Click Sketch, and then select that edge."*
Plan	Proposes a subgoal without specifying the exact action.	*"Let's first make the basic shape, then add the details."*
Confirmation	Provides affirmation on the learner's action, result, or understanding.	*"Yes, that shape looks correct now."*
Diagnosis	Provides feedback on the learner's current state or identifies a problem.	*"The sketch isn't closed, so the software can't generate a solid."*
Explanation	Explains a concept, tool, feature, or reason behind an action or outcome.	*"We use Extrude here because it turns a 2D profile into a 3D solid."*
Tip	Provides a transferable shortcut, heuristic, or practical technique.	*"It's often easier to model one side first and then mirror it."*
Clarification	Asks about the learner's goal, confusion, current state, or intended next step before giving guidance.	*"Does this make sense to you?"*
Reflection	Prompts the learner to evaluate an outcome, compare alternatives, or consider what happened.	*"Looking at the result, what do you think went wrong?"*
Articulation	Prompts the learner to verbalize their reasoning, plan, or decision process.	*"Can you walk me through your reasoning?"*
Exploration	Invites the learner to experiment, choose among alternatives, or practice themselves.	*"Try a couple of options and see which one you like better."*
Non-Instructive	Talks about task-irrelevant topics or filler.	*"The internet is not good."*

Table E.2: Coaching method schema used to annotate expert instructional behavior.

Label Type	Model	1-Gram		2-Gram		3-Gram	
		V	C	V	C	V	C
Dialogue Acts	**GPT-5.4**	0.513	0.472	0.709	0.681	0.839	0.824
	Gemini-3-Flash	<u>0.257</u>	0.251	<u>0.454</u>	0.443	0.644	0.641
	Gemini-3.1-Pro	**0.235**	**0.204**	**0.427**	**0.386**	<u>0.641</u>	**0.605**
	Claude-Sonnet-4.6	<u>0.257</u>	<u>0.207</u>	0.456	<u>0.396</u>	0.657	<u>0.612</u>
	Qwen-3-VL-Instruct	0.283	0.346	0.473	0.553	**0.639**	0.708
	Llama-4-Scout	0.336	0.371	0.541	0.589	0.693	0.738
Coaching Methods	**GPT-5.4**	0.635	0.587	0.827	0.790	0.931	0.910
	Gemini-3-Flash	<u>0.246</u>	<u>0.240</u>	<u>0.461</u>	<u>0.459</u>	<u>0.694</u>	**0.692**
	Gemini-3.1-Pro	0.259	**0.212**	0.481	**0.438**	0.729	<u>0.700</u>
	Claude-Sonnet-4.6	0.301	0.255	0.538	0.491	0.781	0.753
	Qwen-3-VL-Instruct	**0.225**	0.279	**0.439**	0.503	**0.656**	0.707
	Llama-4-Scout	0.306	0.329	0.547	0.578	0.744	0.768

Table F.1: Jensen-Shannon divergence (JSD) for dialogue act and coaching method n-gram distributions under Vanilla (V) and Coach (C) conditions. JSD is bounded in $[0, 1]$; lower values indicate closer alignment with human coach turns, with 0 corresponding to identical distributions. Bold scores indicate the lowest value and underlined scores indicate the second-lowest value within each label type and column.

		Modality		Context			Prompt	
Model	**Default**	**Text-Only**	**Image-Only**	**1s**	**10s**	**60s**	**Vanilla**	**Coach**
GPT-5.4	$0.0150_{\pm 0.0010}$	$0.0150_{\pm 0.0010}$	$0.0130_{\pm 0.0010}$	$0.0130_{\pm 0.0010}$	$0.0150_{\pm 0.0010}$	$0.0150_{\pm 0.0010}$	$0.0120_{\pm 0.0010}$	$0.0120_{\pm 0.0010}$
Gemini-3-Flash	<u>$0.0180_{\pm 0.0010}$</u>	$0.0180_{\pm 0.0010}$	<u>$0.0150_{\pm 0.0010}$</u>	<u>$0.0160_{\pm 0.0010}$</u>	<u>$0.0170_{\pm 0.0010}$</u>	<u>$0.0180_{\pm 0.0010}$</u>	<u>$0.0140_{\pm 0.0010}$</u>	<u>$0.0140_{\pm 0.0010}$</u>
Gemini-3.1-Pro	**$0.0250_{\pm 0.0020}$**	**$0.0280_{\pm 0.0030}$**	**$0.0170_{\pm 0.0010}$**	**$0.0210_{\pm 0.0020}$**	**$0.0230_{\pm 0.0020}$**	**$0.0270_{\pm 0.0030}$**	**$0.0180_{\pm 0.0020}$**	**$0.0190_{\pm 0.0020}$**
Claude-Sonnet-4.6	$0.0170_{\pm 0.0010}$	<u>$0.0220_{\pm 0.0020}$</u>	$0.0140_{\pm 0.0010}$	$0.0150_{\pm 0.0010}$	<u>$0.0170_{\pm 0.0020}$</u>	$0.0170_{\pm 0.0010}$	$0.0130_{\pm 0.0010}$	<u>$0.0140_{\pm 0.0010}$</u>
Qwen-3-VL-Instruct	$0.0160_{\pm 0.0010}$	$0.0150_{\pm 0.0010}$	$0.0130_{\pm 0.0010}$	$0.0140_{\pm 0.0010}$	$0.0160_{\pm 0.0010}$	$0.0160_{\pm 0.0010}$	$0.0130_{\pm 0.0010}$	$0.0130_{\pm 0.0010}$
Llama-4-Scout	$0.0160_{\pm 0.0010}$	$0.0170_{\pm 0.0010}$	$0.0130_{\pm 0.0010}$	$0.0140_{\pm 0.0010}$	$0.0150_{\pm 0.0010}$	$0.0160_{\pm 0.0010}$	$0.0110_{\pm 0.0010}$	$0.0110_{\pm 0.0010}$

Table F.2: BLEU scores (mean $\pm$ 95% CI) for next coach utterance generation. The default setting uses text-image input, 30s context, and oracle prompt. Other columns report ablations over modality, context window, and prompt condition. Bold values indicate the best score per column, and underlined values indicate the second-highest score.

		Modality		Context			Prompt	
Model	**Default**	**Text-Only**	**Image-Only**	**1s**	**10s**	**60s**	**Vanilla**	**Coach**
GPT-5.4	$0.136_{\pm 0.004}$	$0.134_{\pm 0.004}$	$0.120_{\pm 0.004}$	$0.121_{\pm 0.004}$	$0.133_{\pm 0.004}$	$0.134_{\pm 0.004}$	$0.105_{\pm 0.004}$	$0.108_{\pm 0.004}$
Gemini-3-Flash	$0.140_{\pm 0.005}$	$0.139_{\pm 0.005}$	$0.124_{\pm 0.004}$	$0.128_{\pm 0.004}$	$0.138_{\pm 0.005}$	$0.142_{\pm 0.005}$	$0.115_{\pm 0.004}$	$0.111_{\pm 0.004}$
Gemini-3.1-Pro	$\mathbf{0.165_{\pm 0.006}}$	$\mathbf{0.171_{\pm 0.006}}$	$\mathbf{0.126_{\pm 0.005}}$	$\mathbf{0.143_{\pm 0.005}}$	$\mathbf{0.158_{\pm 0.006}}$	$\mathbf{0.169_{\pm 0.006}}$	$\mathbf{0.136_{\pm 0.005}}$	$\mathbf{0.136_{\pm 0.005}}$
Claude-Sonnet-4.6	$0.149_{\pm 0.004}$	$0.164_{\pm 0.005}$	$\mathbf{0.126_{\pm 0.004}}$	$0.134_{\pm 0.004}$	$0.147_{\pm 0.005}$	$0.152_{\pm 0.004}$	$0.126_{\pm 0.004}$	$0.127_{\pm 0.004}$
Qwen-3-VL-Instruct	$0.129_{\pm 0.004}$	$0.126_{\pm 0.004}$	$0.112_{\pm 0.004}$	$0.117_{\pm 0.004}$	$0.128_{\pm 0.004}$	$0.129_{\pm 0.004}$	$0.105_{\pm 0.004}$	$0.097_{\pm 0.004}$
Llama-4-Scout	$0.125_{\pm 0.004}$	$0.134_{\pm 0.004}$	$0.115_{\pm 0.004}$	$0.120_{\pm 0.004}$	$0.123_{\pm 0.005}$	$0.121_{\pm 0.005}$	$0.095_{\pm 0.004}$	$0.089_{\pm 0.004}$

Table F.3: ROUGE-L scores (mean ± 95% CI) for next coach utterance generation. The default setting uses text-image input, 30s context, and oracle prompt. Other columns report ablations over modality, context window, and prompt condition. Bold values indicate the best score per column, and underlined values indicate the second-highest score.

		Modality		Context			Prompt	
Model	**Default**	**Text-Only**	**Image-Only**	**1s**	**10s**	**60s**	**Vanilla**	**Coach**
GPT-5.4	$0.143_{\pm 0.005}$	$0.141_{\pm 0.005}$	$0.127_{\pm 0.005}$	$0.127_{\pm 0.005}$	$0.141_{\pm 0.005}$	$0.144_{\pm 0.005}$	$0.107_{\pm 0.004}$	$0.106_{\pm 0.004}$
Gemini-3-Flash	$0.144_{\pm 0.006}$	$0.146_{\pm 0.005}$	$0.125_{\pm 0.005}$	$0.130_{\pm 0.005}$	$0.142_{\pm 0.006}$	$0.147_{\pm 0.006}$	$0.116_{\pm 0.005}$	$0.109_{\pm 0.005}$
Gemini-3.1-Pro	$0.164_{\pm 0.007}$	$0.166_{\pm 0.007}$	$0.126_{\pm 0.005}$	$0.140_{\pm 0.006}$	$0.157_{\pm 0.007}$	$0.173_{\pm 0.007}$	$0.138_{\pm 0.006}$	$0.136_{\pm 0.006}$
Claude-Sonnet-4.6	$\mathbf{0.178_{\pm 0.006}}$	$\mathbf{0.179_{\pm 0.007}}$	$\mathbf{0.149_{\pm 0.005}}$	$\mathbf{0.148_{\pm 0.005}}$	$\mathbf{0.173_{\pm 0.006}}$	$\mathbf{0.182_{\pm 0.006}}$	$\mathbf{0.147_{\pm 0.005}}$	$\mathbf{0.147_{\pm 0.005}}$
Qwen-3-VL-Instruct	$0.125_{\pm 0.005}$	$0.123_{\pm 0.005}$	$0.108_{\pm 0.004}$	$0.109_{\pm 0.004}$	$0.121_{\pm 0.005}$	$0.123_{\pm 0.005}$	$0.098_{\pm 0.005}$	$0.086_{\pm 0.004}$
Llama-4-Scout	$0.116_{\pm 0.005}$	$0.142_{\pm 0.005}$	$0.114_{\pm 0.005}$	$0.127_{\pm 0.005}$	$0.117_{\pm 0.005}$	$0.113_{\pm 0.005}$	$0.091_{\pm 0.004}$	$0.079_{\pm 0.004}$

Table F.4: METEOR scores (mean ± 95% CI) for next coach utterance generation. The default setting uses text-image input, 30s context, and oracle prompt. Other columns report ablations over modality, context window, and prompt condition. Bold values indicate the best score per column, and underlined values indicate the second-highest score.

		Modality		Context			Prompt	
Model	**Default**	**Text-Only**	**Image-Only**	**1s**	**10s**	**60s**	**Vanilla**	**Coach**
GPT-5.4	$0.840_{\pm 0.001}$	$0.839_{\pm 0.001}$	$0.836_{\pm 0.001}$	$0.836_{\pm 0.001}$	$0.840_{\pm 0.001}$	$0.840_{\pm 0.001}$	$0.834_{\pm 0.001}$	$0.835_{\pm 0.001}$
Gemini-3-Flash	$0.846_{\pm 0.001}$	$0.845_{\pm 0.001}$	$0.843_{\pm 0.001}$	$0.844_{\pm 0.001}$	$0.845_{\pm 0.001}$	$0.846_{\pm 0.001}$	$0.839_{\pm 0.001}$	$0.839_{\pm 0.001}$
Gemini-3.1-Pro	$\mathbf{0.850_{\pm 0.001}}$	$\mathbf{0.852_{\pm 0.001}}$	$\mathbf{0.844_{\pm 0.001}}$	$\mathbf{0.847_{\pm 0.001}}$	$\mathbf{0.849_{\pm 0.001}}$	$\mathbf{0.851_{\pm 0.001}}$	$\mathbf{0.842_{\pm 0.001}}$	$\mathbf{0.844_{\pm 0.001}}$
Claude-Sonnet-4.6	$0.841_{\pm 0.001}$	$0.845_{\pm 0.001}$	$0.837_{\pm 0.001}$	$0.840_{\pm 0.001}$	$0.842_{\pm 0.001}$	$0.842_{\pm 0.001}$	$0.835_{\pm 0.001}$	$0.836_{\pm 0.001}$
Qwen-3-VL-Instruct	$0.841_{\pm 0.001}$	$0.840_{\pm 0.001}$	$0.837_{\pm 0.001}$	$0.838_{\pm 0.001}$	$0.840_{\pm 0.001}$	$0.840_{\pm 0.001}$	$0.837_{\pm 0.001}$	$0.837_{\pm 0.001}$
Llama-4-Scout	$0.840_{\pm 0.001}$	$0.841_{\pm 0.001}$	$0.837_{\pm 0.001}$	$0.838_{\pm 0.001}$	$0.839_{\pm 0.001}$	$0.840_{\pm 0.001}$	$0.836_{\pm 0.001}$	$0.835_{\pm 0.001}$

Table F.5: BERTScore scores (mean ± 95% CI) for next coach utterance generation. The default setting uses text-image input, 30s context, and oracle prompt. Other columns report ablations over modality, context window, and prompt condition. Bold values indicate the best score per column, and underlined values indicate the second-highest score.

Model	**Overall**	**Figma**	**Blender**	**OnShape**	**FL Studio**	**Excel**
GPT-5.4	$32.4_{\pm 1.3}$	$32.4_{\pm 3.0}$	$29.8_{\pm 2.9}$	$32.7_{\pm 2.8}$	$35.0_{\pm 3.2}$	$32.5_{\pm 3.0}$
Gemini-3-Flash	$33.9_{\pm 1.4}$	$36.8_{\pm 3.3}$	$30.6_{\pm 3.0}$	$35.1_{\pm 3.1}$	$33.7_{\pm 3.4}$	$33.2_{\pm 3.1}$
Gemini-3.1-Pro	$\mathbf{41.4_{\pm 1.5}}$	$\mathbf{40.2_{\pm 3.5}}$	$\mathbf{38.6_{\pm 3.3}}$	$\mathbf{45.1_{\pm 3.3}}$	$\mathbf{41.7_{\pm 3.6}}$	$\mathbf{41.4_{\pm 3.3}}$
Claude-Sonnet-4.6	$32.3_{\pm 1.3}$	$32.4_{\pm 3.0}$	$30.4_{\pm 3.0}$	$34.1_{\pm 3.0}$	$33.3_{\pm 3.1}$	$31.5_{\pm 2.9}$
Qwen-3-VL-Instruct	$27.0_{\pm 1.2}$	$26.2_{\pm 2.7}$	$24.4_{\pm 2.5}$	$28.0_{\pm 2.5}$	$29.1_{\pm 2.9}$	$27.7_{\pm 2.8}$
Llama-4-Scout	$19.5_{\pm 1.0}$	$19.8_{\pm 2.3}$	$17.3_{\pm 2.0}$	$19.0_{\pm 1.9}$	$21.9_{\pm 2.5}$	$19.8_{\pm 2.2}$

Table F.6: CLAIR scores (mean ± 95% CI) by domain under the text+image input, 30s, Oracle condition. Bold values indicate the best score per column, and underlined values indicate the second-highest score.

Model	Overall	Artic.	Clarif.	Confirm.	Diagn.	Direct	Explan.	Explor.	Plan	Reflect.	Tip
GPT-5.4	$32.4_{\pm 1.3}$	$35.6_{\pm 5.1}$	$23.4_{\pm 3.6}$	$\underline{41.9}_{\pm 4.3}$	$23.2_{\pm 3.5}$	$32.1_{\pm 4.1}$	$\underline{38.6}_{\pm 4.3}$	$33.6_{\pm 4.1}$	$35.6_{\pm 4.1}$	$29.1_{\pm 3.9}$	$31.7_{\pm 4.1}$
Gemini-3-Flash	$\underline{33.9}_{\pm 1.4}$	$\underline{40.6}_{\pm 5.4}$	$\underline{27.0}_{\pm 4.3}$	$37.6_{\pm 4.4}$	$\underline{30.0}_{\pm 4.4}$	$33.2_{\pm 4.5}$	$37.4_{\pm 4.5}$	$\underline{37.2}_{\pm 4.2}$	$\underline{38.4}_{\pm 4.6}$	$\underline{30.3}_{\pm 4.2}$	$29.0_{\pm 4.3}$
Gemini-3.1-Pro	$\mathbf{41.4}_{\pm 1.5}$	$\mathbf{48.8}_{\pm 5.7}$	$\mathbf{34.0}_{\pm 4.7}$	$\mathbf{51.4}_{\pm 4.8}$	$\mathbf{34.0}_{\pm 4.4}$	$\mathbf{39.9}_{\pm 4.8}$	$\mathbf{42.8}_{\pm 4.7}$	$\mathbf{46.6}_{\pm 4.5}$	$\mathbf{43.7}_{\pm 4.6}$	$\mathbf{37.0}_{\pm 4.9}$	$\mathbf{37.9}_{\pm 4.6}$
Claude-Sonnet-4.6	$32.3_{\pm 1.3}$	$39.3_{\pm 4.9}$	$24.4_{\pm 3.8}$	$36.1_{\pm 4.3}$	$27.7_{\pm 4.2}$	$\underline{33.7}_{\pm 4.3}$	$36.0_{\pm 4.2}$	$34.5_{\pm 4.2}$	$33.3_{\pm 4.2}$	$28.2_{\pm 3.8}$	$\underline{32.1}_{\pm 4.1}$
Qwen-3-VL-Instruct	$27.0_{\pm 1.2}$	$28.1_{\pm 4.5}$	$19.5_{\pm 2.9}$	$33.0_{\pm 4.1}$	$19.7_{\pm 2.8}$	$24.4_{\pm 3.5}$	$32.3_{\pm 4.1}$	$28.7_{\pm 3.7}$	$33.2_{\pm 4.0}$	$26.0_{\pm 3.5}$	$25.8_{\pm 3.7}$
Llama-4-Scout	$19.5_{\pm 1.0}$	$24.3_{\pm 4.1}$	$15.3_{\pm 2.7}$	$17.1_{\pm 2.8}$	$15.0_{\pm 2.4}$	$17.6_{\pm 2.8}$	$20.3_{\pm 3.2}$	$25.3_{\pm 3.6}$	$24.9_{\pm 3.6}$	$17.0_{\pm 2.4}$	$19.5_{\pm 2.7}$

Table F.7: CLAIR scores (mean $\pm$ 95% CI) by coaching method under the Text+Image, 30s, Oracle condition. Bold values indicate the best score per column, and underlined values indicate the second-highest score.

Tag	Setup	Tutorial Task	Pre-Task	Post-Task
Gave up	H	0 (0.0%)	19 (26.4%)	0 (0.0%)
	M	15 (41.7%)	31 (86.1%)	17 (47.2%)
Wrong method	H	0 (0.0%)	17 (23.6%)	2 (2.8%)
	M	3 (8.3%)	7 (19.4%)	2 (5.6%)
Mistakes	H	0 (0.0%)	3 (4.2%)	5 (6.9%)
	M	5 (13.9%)	0 (0.0%)	7 (19.4%)
Missed steps	H	0 (0.0%)	2 (2.8%)	8 (11.1%)
	M	7 (19.4%)	0 (0.0%)	9 (25.0%)

Table H.1: Problematic outcome tags for human coaching sessions (H; $N = 72$) and model coaching sessions (M; $N = 36$). Tags are not mutually exclusive.

Next Coach Utterance Generation Prompt

System message:

```
You are an expert coach guiding a learner in a real-time computer use coaching session
using {{SOFTWARE_TOOL}}.
Your goal is not only to complete the current tutorial, but to help the learner develop
skill so that they can solve similar problems on their own in the future.


You are going to speak as the coach to continue the conversation.
You can choose the appropriate coaching method(s) defined in the following section to use
in your response.
You are looking at the learner's recent screen and you have your recent conversation with
the learner.
The cursor is where the learner is currently interacting with the software.
If there are any visual pink color screen annotations, those are your previous annotations
to the learner's screen. You can use them to guide your response.
Your response should sound like something a human coach would actually say aloud in
real-time during the session.


Generation Rules:
- Give one concise sentence as your response.
- Treat the chat history as a live conversation (user = Learner, assistant = Coach).
- Avoid overly verbose and overly formal language.
- Avoid any special characters or markdown formatting like lists, bold, italic, etc.
- Avoid transcript artifacts such as ellipses, dashes.
- Do not mention being an AI, model, assistant, or system.


Output Rules:
- Return ONLY the coach utterance as plain text.
- No JSON, no speaker prefix, no markdown.


Coaching Method Definitions and Examples:
{{COACHING_METHODS_DEFINITION_AND_EXAMPLES}}


Task Description:
{{TASK_DESCRIPTION}}
```

User message: Dialogue History

```
Recent {{CONTEXT_WINDOW}} dialogue history (ordered from earliest to most recent):

{{Learner turns are shown as User messages. Coach turns are shown as Assistant messages.}}
```

User message: Visual Context

```
Recent {{CONTEXT_WINDOW}} learner's recent screen (ordered from earliest to most recent):


{{RECENT_SCREEN_FRAMES}}
```

User message: Final Instruction

```
You are going to speak as the coach to continue the conversation.
Choose the most appropriate coaching method(s) and give your coaching utterance. ONLY return
the coaching utterance, no coaching method tags.
Your next coach utterance should use the following coaching method(s):
{{COACHING_METHODS}}
```

Figure G.4: Prompt structure for next coach utterance generation, including system prompt, recent dialogue, recent screen context, and final instruction. Colored text marks prompt condition additions: blue for both Coach and Oracle prompts, orange for Coach only, and red for Oracle only.

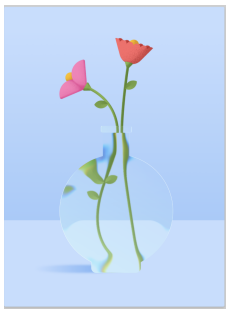
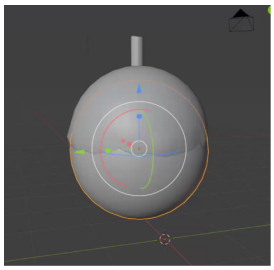
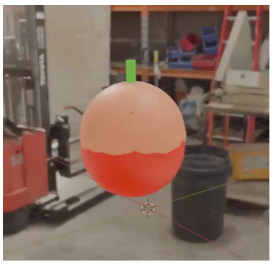
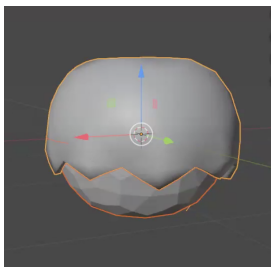
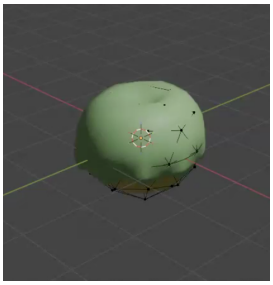
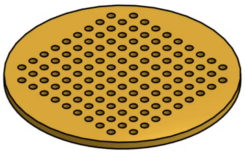
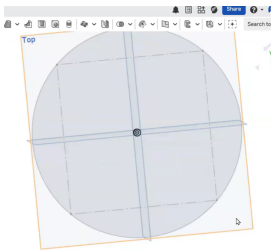
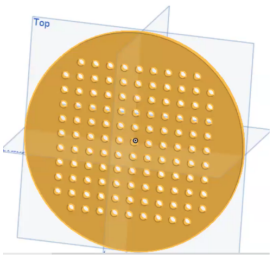
Software	Task ID	Session ID	Target	Pre-Task	Post-Task
Figma	0104	C1L1			
Blender	0202	C5L5			
Blender	0202	C7L7			
OnShape	0302	C8L8			

Table H.2: Examples of pre-task and post-task outcomes in visual tasks (Figma, Blender, OnShape).

K AI Use Disclosure

The authors used AI-based tools to assist with code generation, editing, and writing during the preparation of this paper. Specifically, AI assistance was used to help draft and revise portions of the manuscript for clarity, grammar, and organization, and to support the development, debugging, and refinement of code used in the research workflow. All AI-generated or AI-assisted content, code, analyses, and interpretations were reviewed, verified, and, where necessary, modified by the authors. The authors take full responsibility for the accuracy, integrity, originality, and final content of the paper, including any code or text developed with AI assistance.