MINDCOMPLETION: LEARNING LATENT HUMAN INTENT FROM BEHAVIORAL CONTINUATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models are increasingly deployed as proactive writing assistants, suggesting continuations as the writer drafts. Effective writing assistance demands more than fluency: the assistant must anticipate what the writer intends to say next, a behavioral judgment that standard language model training is not designed to capture. To address this, we propose **process-continuation learning**, a paradigm that turns naturally occurring expert writing traces into behavioral rewards for reinforcement learning, operationalized through a structured user agent without per-instance annotation. To stabilize optimization, we schedule a curriculum reward progressing from local validity to semantic alignment, and construct context-variant trajectories covering prefix-only and history-conditioned settings. Across three co-writing benchmark families, the resulting 8B policy consistently improves suggestion acceptance and is competitive with strong closed-source assistants. Human evaluation confirms writer-facing gains: it wins 62.4% of pairwise comparisons against the base model and more than doubles real acceptance rate in a user study. The gains further transfer to unseen academic and textbook domains, demonstrating that naturally occurring continuations provide scalable behavioral supervision for models that must anticipate users' next-step intent.

1 INTRODUCTION

Human-AI collaborative writing has attracted growing interest for augmenting the drafting process: an assistant observes the evolving document and proactively proposes continuations, while the writer decides whether to accept, revise, or reject each suggestion Lee et al. (2022); Coenen et al. (2021); Fang et al. (2026); Jackson et al. (2025). The quality of this collaboration hinges on the assistant's

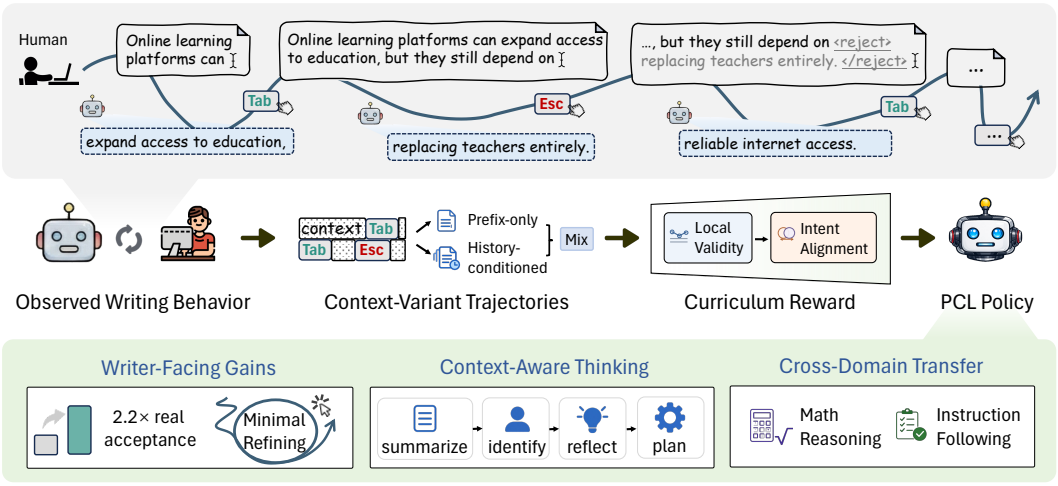


Figure 1: **Overview of process-continuation learning (PCL).** PCL turns observed writing behavior into behavioral supervision through context-variant trajectories and a curriculum from local validity to intent alignment, yielding improved co-writing, context-aware thinking, and cross-domain transfer.

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

ability to anticipate the writer's *next intended move*—not merely to produce fluent text, but to generate a continuation that aligns with what the writer would naturally write next.

Yet existing training objectives are poorly matched to this requirement. Standard language-model and instruction-tuning objectives optimize token-level likelihood Ouyang et al. (2022); Touvron et al. (2023), treating an observed continuation as the target surface form even though many different sentences may realize the same writer intent. Preference learning moves supervision closer to user judgment Rafailov et al. (2023); Ethayarajh et al. (2023); Hong et al. (2024); Jung et al. (2025), but typically relies on explicitly collected comparisons or feedback that are expensive to obtain at scale in open-ended writing domains. Meanwhile, naturally occurring human writing already records a sequence of behavioral decisions: every sentence in a well-crafted document reveals what its author chose to express next in a particular context.

We therefore propose **process-continuation learning (PCL)**, a paradigm that uses these implicit authorial decisions as behavioral supervision. Rather than fitting the observed next sentence as a unique token-level target, we treat it as a behavioral reference: a candidate continuation is rewarded when it is locally valid and compatible with the author's intended next step, even when its surface form differs from the reference. We operationalize this sentence-level judgment with a structured *user agent*, enabling reinforcement learning over naturally occurring traces without per-instance annotation.

Two design choices are critical for instantiating this framework in practice. First, early-stage policies struggle to satisfy both reward channels at once—semantic alignment provides little useful gradient before local continuation behavior has been established; we therefore introduce a **curriculum reward** that begins with local validity and then switches to semantic alignment, allowing the policy to develop continuation capability in stages. Second, context may be prefix-only or history-conditioned; we construct **context-variant trajectories** and call policies trained on the former, the latter, or both **PCL-Prefix**, **PCL-History**, and **PCL-Mix**, respectively.

We evaluate across three complementary benchmark families—MindCopilot Fang et al. (2026), CoAuthor Lee et al. (2022), and ScholaWrite Le et al. (2026)—spanning five continuation settings, using hierarchical acceptance rate (HAR) for suggestion acceptance and accepted-only knowledge-weighted edit distance (KED) Fang et al. (2026) for post-acceptance revision effort. PCL-Mix reaches 36.10 Macro HAR—an equal-weight average over the five settings—outperforming the matched mixed-trajectory SFT baseline by 7.69 points and remaining competitive with strong closed-source assistants. Human evaluation confirms that these offline gains are writer-facing: PCL-Prefix wins 62.4% of pairwise comparisons against the base model, while PCL-History more than doubles real acceptance rate in a controlled user study. On external continuation sets drawn from 100 academic papers and 100 textbooks, both trained variants outperform the base model across all five evaluation dimensions. Reward ablations show that local validity provides the strongest early learning signal and that switching to semantic alignment raises the final acceptance ceiling; trajectory ablations further show that mixed training improves all five evaluation settings over prefix-only training.

Beyond interactive co-writing, process-continuation training yields two additional findings. First, **context-aware thinking** emerges in model reasoning traces as the policy learns to summarize context, identify discourse roles, reflect on feedback, and plan the next continuation. Second, writing-trained policies show modest gains on mathematical reasoning and instruction following, while applying the same paradigm to mathematical solution traces produces larger improvements. Together, these results suggest that sentence-level behavioral optimization can support capabilities that require anticipating sequential intent.

Our contributions are as follows:

- We propose **process-continuation learning**, which treats naturally occurring expert continuations as behavioral evidence rather than unique imitation targets, providing scalable sentence-level supervision without explicit preference annotation.
- We instantiate the paradigm with a **behavior-aligned curriculum reward** and **context-variant trajectories**; controlled ablations show that staging local validity before semantic alignment raises the performance ceiling and that mixing prefix-only with history-conditioned trajectories improves performance across evaluation settings.
- Across three benchmark families and human and out-of-distribution evaluations, we demonstrate writer-facing gains, emergent context-aware thinking, and broader cross-domain applicability.

2 Related Work

Human-AI Collaborative Writing. A growing body of work has studied how writers interact with AI generation systems Ashkinaze et al. (2025); Zhang et al. (2025); He et al. (2025); Yang et al. (2022). Early studies found that accepting, modifying, or rejecting suggestions reflects nuanced authorial intent beyond surface-level quality Clark et al. (2018); Storch (2019), and subsequent efforts provided structured interaction corpora at scale Lee et al. (2022); Coenen et al. (2021), enabling analysis of how suggestion granularity, genre, and user expertise shape acceptance behavior. More recently, research has moved toward proactive suggestion strategies Shen & Wu (2023); Devatine & Abraham (2024); Le et al. (2026) and formal characterizations of interactive writing as a decision process with hierarchical acceptance criteria Fang et al. (2026). Despite this progress, these behavioral signals have largely served as evaluation targets rather than training objectives; our work bridges this gap by formulating behavioral traces as structured reward signals for policy optimization.

Feedback-Based Optimization and User Simulation. RLHF Chaudhari et al. (2025); Sharma et al. (2025); Bai et al. (2022); Kaufmann et al. (2023); Lee et al. (2023) relies on explicit preference annotations to align language models with human judgment, and has become the dominant paradigm for post-training alignment. To reduce annotation cost, a parallel line of work turns to user simulation, employing prompted LLMs or rule-based proxies to generate synthetic feedback at scale Le et al. (2026); Zhou et al. (2025); Qian et al. (2025). However, such simulators are typically constructed without grounding in real behavioral data, making their fidelity to authentic user responses difficult to validate Lin & Tomlin (2025a;b). In contrast, our reward mechanism derives its evaluation criteria from naturally-occurring writing interactions and is validated against real human judgments, ensuring that the optimization signal reflects authentic authorial behavior.

3 Process-Continuation Learning

3.1 Task Formulation

We formalize interactive writing as a process-continuation problem, where the assistant predicts the writer's next sentence given the current document context. To simulate the moment a writer pauses, we set breakpoints in naturally occurring documents: the prefix up to each breakpoint defines the writing state s_t, and the original author's next sentence serves as a behavioral reference a_t^* that reveals the writer's latent intent at that point. Formally, a policy π_θ generates a candidate snippet

$$a_t \sim \pi_\theta(\cdot \mid s_t), \tag{1}$$

where $a_t \in \mathcal{A}$ is a proposed continuation. Conditioned on both the writing state and the behavioral reference, the acceptance decision is modeled as

$$u_t \sim p_{\text{user}}(\cdot \mid s_t, a_t, a_t^*), \tag{2}$$

where $u_t \in \mathcal{U}$ encodes the accept/revise/reject outcome and p_{user} is an idealized model of the writer's acceptance decision. The state then evolves as

$$s_{t+1} = \mathcal{T}(s_t, a_t, u_t), \tag{3}$$

after which the assistant continues generation from the updated state.

Optimization objective. The user's acceptance decision provides the supervision signal for policy optimization. We learn a policy that maximizes the expected response-derived reward:

$$\max_\theta \ \mathbb{E}_{(s_t, a_t^*)\sim\mathcal{D},\, a_t\sim\pi_\theta(\cdot|s_t)} \left[R(s_t, a_t, a_t^*) \right], \tag{4}$$

where $R(s_t, a_t, a_t^*)$ measures how well the proposed snippet aligns with both the writing state and the behavioral reference.

3.2 User Agent

Since the real user p_{user} is unavailable in naturally occurring writing traces, we approximate it with a prompt-based user agent that produces structured judgment signals without per-instance annotation. Specifically, we instantiate the response space $\mathcal{U}$ with three binary judgment channels:

$$\mathcal{U} = \{y_{\text{loc}}, y_{\text{sem}}, y_{\text{hier}}\}, \tag{5}$$

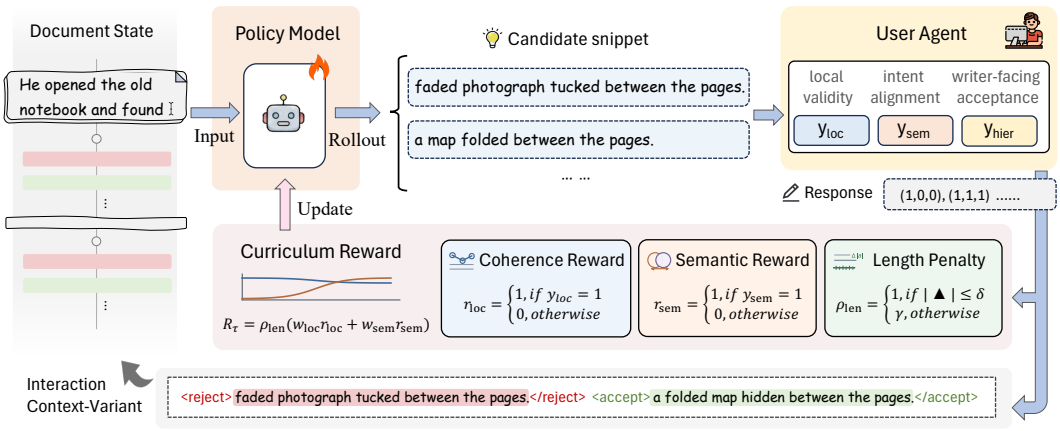


Figure 2: **Framework of process-continuation learning.** Given a document state, the policy rolls out candidate snippets evaluated by a structured user agent along three judgment channels (y_{loc}, y_{sem}, y_{hier}). The resulting signals are composed into a curriculum reward for policy optimization. Context-variant trajectories (bottom) augment the input with prior interaction history for multi-turn training.

where y_{loc} indicates whether the snippet is a locally valid continuation of the current writing state, y_{sem} indicates whether it is consistent with the writer's intended next step as evidenced by a_t^*, and y_{hier} indicates a hierarchical judgment that evaluates the snippet as a writer-facing suggestion, following the hierarchical acceptance criteria proposed in Fang et al. (2026).

3.3 Reward Design

While y_{hier}—the holistic-acceptance signal, used as the *HAR reward* when optimized directly in our ablation—is the natural optimization target, directly using it as reward is difficult: for early-stage policies, multiple criteria are violated simultaneously, making the signal sparse. We therefore construct a curriculum reward from the decomposed channels y_{loc} and y_{sem}:

$$r_{\text{loc}}(s_t, a_t) = \mathbb{I}[y_{\text{loc}} = 1], \tag{6}$$

$$r_{\text{sem}}(s_t, a_t, a_t^*) = \mathbb{I}[y_{\text{sem}} = 1], \tag{7}$$

The local-validity reward encourages snippets that are compatible with the current writing state, while the semantic reward encourages alignment with the writer's observed next-step intent.

Length penalty. Beyond content quality, a useful suggestion should also match the scope the writer expects. We therefore attenuate the reward for snippets whose length substantially deviates from the observed continuation, as such mismatches increase revision effort:

$$p_{\text{len}}(a_t, a_t^*) = \begin{cases} 1, & \text{if } \big||a_t| - |a_t^*|\big| \le \delta, \\ \gamma, & \text{otherwise}, \end{cases} \tag{8}$$

where $|\cdot|$ denotes snippet length, δ is a tolerance threshold, and $\gamma \in (0, 1)$ is an attenuation factor.

Curriculum reward. The final reward is

$$R(s_t, a_t, a_t^*; \tau) = p_{\text{len}}(a_t, a_t^*)\big[w_{\text{loc}}(\tau) r_{\text{loc}}(s_t, a_t) + w_{\text{sem}}(\tau) r_{\text{sem}}(s_t, a_t, a_t^*)\big], \tag{9}$$

where τ is the training step, and $w_{\text{loc}}(\tau)$ and $w_{\text{sem}}(\tau)$ control the curriculum schedule. Training begins with r_{loc} dominant so that the policy first learns locally coherent continuations; the weight on r_{sem} is then increased to shift the objective toward writer-intent alignment, while the length penalty remains active throughout.

3.4 Context-Variant Policy Training

The reward in Section 3.3 is defined over a generic writing state s_t, but the observable context varies across interaction modes: the policy may generate from a document prefix alone or additionally condition on prior suggestions and user-agent responses. Optimizing for only one regime risks brittleness, so we construct two state variants that share the same reward function but differ in the information included in s_t.

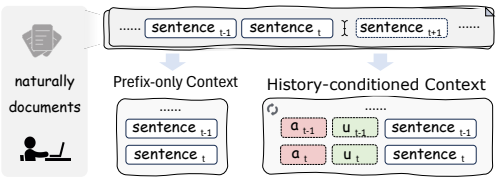


Figure 3: Context-variant trajectory construction from naturally occurring documents: prefix-only context (PCL-Prefix) and history-conditioned context interleaving prior suggestions a_t and user responses u_t (PCL-History).

Prefix-only context. The prefix-only variant corresponds to context-free continuation, where the state contains only the document prefix:

$$s_t = c_t, \tag{10}$$

where $c_t \in \mathcal{C}$ denotes the current writing context. Training pairs are sampled directly from naturally occurring writing traces:

$$(s_t, a_t^*) = (c_t, a_t^*) \sim \mathcal{D}^{\text{pfx}}. \tag{11}$$

History-conditioned context. The history-conditioned variant augments the document prefix with a short interaction history:

$$s_t = (c_t, h_t), \qquad h_t = \{(\tilde{a}_j, \tilde{u}_j)\}_{j=1}^{m}, \tag{12}$$

where $\tilde{a}_j$ denotes a previous assistant suggestion and $\tilde{u}_j$ the corresponding user-agent response. Since naturally occurring traces do not contain real multi-turn interactions, we synthesize h_t offline by sampling auxiliary snippets for the same writing state and collecting user-agent feedback. The document prefix c_t and behavioral reference a_t^* remain grounded in the original human-written trace, ensuring that the target continuation objective is unchanged.

We refer to the policy trained on prefix-only instances as **PCL-Prefix**, the policy trained on history-conditioned instances as **PCL-History**, and the policy jointly trained on both instance types as **PCL-Mix**.

4 Experiments

We evaluate process-continuation learning on interactive co-writing, examining suggestion acceptance against competitive baselines, the contribution of each reward component, and the effect of context-variant training on generalization.

4.1 Experimental Setup

Training data. We construct training data from high-quality long-form articles covering diverse domains and writing styles, from which we construct both prefix-only and history-conditioned training instances; details on collection, segmentation, and context construction are in the Appendix B.

Evaluation protocol. We evaluate on three complementary benchmark families. First, MindCopilot (Fang et al., 2026) measures proactive continuation under both single-turn document context and multi-turn interaction history. Second, we derive two continuation test sets from CoAuthor (Lee et al., 2022): CoAuthor-Human asks models to predict the next sentence written directly by a human, whereas CoAuthor-Interactive replays real suggestion-request states and uses the continuation ultimately realized in the document as reference. Third, we adapt ScholaWrite (Le et al., 2026) into ScholaWrite-Continuation by reconstructing consecutive text-production edits into writing bursts and extracting the first completed new sentence. Together, these benchmarks test in-domain interaction, transfer to authentic human–AI writing sessions, and generalization to long-horizon scholarly writing. All models receive the same observable context. We run each benchmark four times; in each run, every model generates one continuation per instance, and the reported scores pool the four stochastic outputs. We report offline hierarchical acceptance rate (HAR) and accepted-only knowledge-weighted edit distance (KED); benchmark construction and metric definitions are detailed in Appendix D.1 and Appendix D.2, respectively.

Table 1: Per-benchmark continuation results pooled across four stochastic runs. PREFIX denotes prefix-only training, while MIX denotes a 1:1 mixture of prefix-only and history-conditioned instances. API are reference-only and excluded from ranking; among non-API systems, **bold** and underline mark the best and second-best HAR.

Method	MindCopilot Single-turn		MindCopilot Multi-turn		CoAuthor Human		CoAuthor Interactive		ScholaWrite Continuation	
	HAR ↑	KED ↓	HAR ↑	KED ↓	HAR ↑	KED ↓	HAR ↑	KED ↓	HAR ↑	KED ↓
Closed API reference										
Gemini-3.1-Flash-Lite	32.23	10.31	21.80	9.70	42.09	7.44	42.48	7.31	31.49	11.36
GPT-5.4-Nano	29.44	9.93	21.50	10.01	41.00	7.03	45.64	6.88	30.64	10.82
Qwen3-8B base and training baselines										
Qwen3-8B (Base)	10.31	13.81	8.99	13.18	13.78	8.26	17.28	7.18	9.27	12.45
Embedding-Prefix	12.26	9.70	3.97	11.97	22.58	6.68	25.14	6.27	10.21	11.29
SFT-Prefix	**27.96**	9.26	14.57	9.65	34.91	7.27	38.28	7.00	22.13	10.71
SFT-Mix	27.49	9.07	13.98	9.53	39.37	7.26	39.95	6.94	21.28	11.40
Process-continuation learning (ours)										
PCL-Prefix	25.53	9.93	19.02	10.29	47.38	6.79	50.33	6.52	25.96	9.61
PCL-Mix	27.55	10.07	**19.96**	9.89	**49.74**	6.74	**52.62**	6.47	**30.64**	10.46

Baselines. We compare against two API systems—Gemini-3.1-Flash-Lite (Google DeepMind, 2025) and GPT-5.4-Nano (OpenAI, 2025)—as high-capability references. To isolate the contribution of the learning objective, we train embedding-reward and supervised fine-tuning (SFT) baselines on prefix-only trajectories. The embedding-reward baseline uses Qwen3-Embedding-0.6B (Qwen Team, 2025) to compute the similarity between the model continuation and the behavioral reference, using this score as the training reward. We additionally include SFT with mixed trajectories as a matched control for separating the effect of trajectory composition from that of process-continuation learning. All trainable models use Qwen3-8B (Yang et al., 2025) as the starting checkpoint, and all systems share the same generation prompt, sampling configuration, and evaluation judge.

Implementation. All models use the same prompt template and sampling configuration. Training follows the curriculum reward in Eq. equation 9 via GRPO Shao et al. (2024): stage one optimizes local validity until its training reward plateaus, after which training resumes from the selected checkpoint with the semantic-alignment objective; the length penalty remains active throughout. The transition criterion and resulting checkpoint are detailed in Appendix C, and the curriculum is ablated in Section 4.3.

4.2 MAIN RESULTS

Overall comparison. Tables 1 and 2 compare the shared local-validity-to-semantic-alignment curriculum across the focal trajectory families against the Qwen3-8B base model and all API, embedding-reward, and SFT baselines. Macro HAR weights the five benchmarks equally, whereas Weighted HAR pools all evaluation instances. The mixed-trajectory policy achieves 36.10 Macro HAR, improving over the matched SFT-Mix baseline (28.41) by 7.69 points and increasing Weighted HAR from 36.83 to 46.88. Relative to the Qwen3-8B base model, it improves Macro and Weighted HAR by 24.17 and 32.65 points, respectively.

Table 2: Aggregate HAR results (%).

Method	Macro ↑	Weighted ↑
Gemini-3.1-Flash-Lite	34.02	40.05
GPT-5.4-Nano	33.65	40.19
Qwen3-8B (Base)	11.93	14.23
Embedding-Prefix	14.83	21.26
SFT-Prefix	27.57	33.94
SFT-Mix	28.41	36.83
PCL-Prefix	33.64	44.64
PCL-Mix	**36.10**	**46.88**

Cross-benchmark transfer. The gains are concentrated in transfer settings: compared with the best SFT result on each dataset, PCL-Mix improves multi-turn MindCopilot by 5.39 points, CoAuthor-Human by 10.37 points, CoAuthor-Interactive by 12.67 points, and ScholaWrite by 8.51 points. On

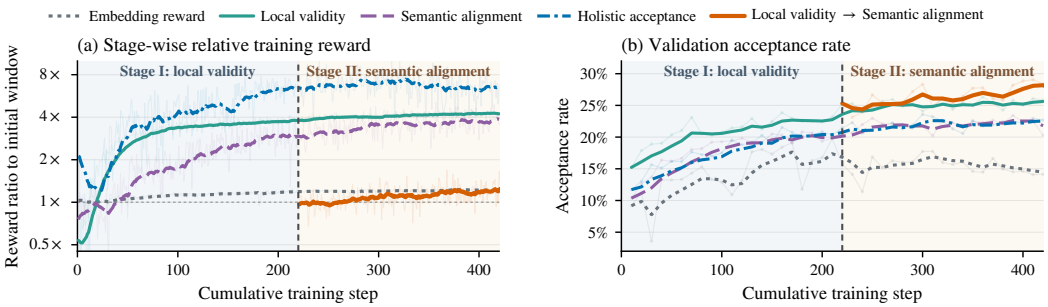


Figure 4: Reward-objective ablation with prefix-only trajectories. Panel (a) reports training reward normalized to each run's initial window; values are comparable only within an objective. Panel (b) reports validation acceptance rate. The dashed line marks the plateau-triggered transition from local validity to semantic alignment, after which the curriculum attains the highest final validation acceptance rate.

single-turn MindCopilot, SFT-Prefix remains marginally higher (27.96 vs. 27.55), indicating that the benefit of process-continuation learning is strongest under interaction and distribution shift rather than prefix-only imitation alone.

Comparison with API systems. PCL-Mix exceeds Gemini-3.1-Flash-Lite and GPT-5.4-Nano in aggregate, by 2.08 and 2.45 Macro-HAR points, respectively, with substantially larger gains on both CoAuthor sets. Both API systems remain stronger on the two MindCopilot settings. We therefore view the comparison as evidence that process-continuation learning makes a compact open model competitive with strong general-purpose assistants, particularly under interaction and distribution shift. Accepted-only KED is reported alongside HAR but is not compared in isolation because each model induces a different accepted subset.

Prefix-only and mixed trajectories. Under the shared local-validity-to-semantic-alignment curriculum, mixed-trajectory training obtains higher Macro HAR than prefix-only training (36.10 vs. 33.64) and improves all five benchmark-level HAR scores. Section 4.4 isolates this effect and examines how trajectory diversity interacts with curriculum learning.

4.3 Ablation on Reward Composition

We compare the proposed curriculum with four single-objective alternatives—embedding similarity, local validity, semantic alignment, and the holistic-acceptance reward—under the same optimization and evaluation setup (Figure 4). All curves cover 420 steps, and the curriculum switches objectives once the local-validity reward plateaus. Panel (a) normalizes each reward to its own initial window, so cross-objective quality should be judged by validation acceptance in Panel (b); Appendix E reports the corresponding raw rewards and the exact transition checkpoint.

Single-objective training exposes a reward–metric gap. The holistic-acceptance reward achieves the largest relative training-reward increase, but its validation curve remains near the semantic-only run and below local-validity training; the embedding objective remains weakest. Local validity improves acceptance most rapidly and leads over most checkpoints, indicating that a dense continuation-level signal is a stronger early learning target than directly optimizing higher-level alignment.

The curriculum raises the performance ceiling. After the switch, semantic-alignment supervision preserves the local-validity gains and lifts validation acceptance to approximately 28% by step 420, above the 25–26% local-validity plateau and all other single objectives. Because semantic alignment alone does not follow this trajectory, ordering matters: local validity establishes continuation competence before semantic alignment refines writer-intent matching.

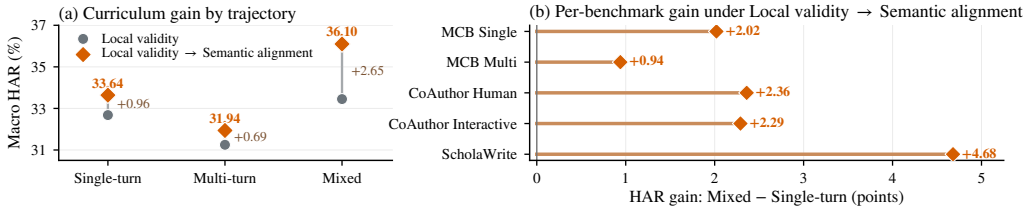


Figure 5: Effect of training-trajectory composition. *Left:* Macro HAR before and after introducing the semantic curriculum for single-turn, multi-turn, and mixed trajectories. *Right:* per-benchmark HAR gains of mixed over single-turn training under the local-validity-to-semantic-alignment curriculum.

4.4 Effect of Training-Trajectory Composition

To isolate trajectory composition from reward design, we hold the model, optimization recipe, evaluation protocol, and curriculum fixed, varying only whether training uses prefix-only, history-conditioned, or mixed trajectories.

Mixed trajectories amplify curriculum learning. Figure 5 (left) shows that the curriculum improves every trajectory family, with the largest Macro HAR gain under mixed training ($+2.65$ points). Its advantage over either trajectory family alone suggests that prefix-only and history-conditioned examples provide complementary supervision.

The benefit transfers across evaluation settings. Mixed training also improves over single-turn training across all five benchmarks (Figure 5, right), with the strongest gain on the long-horizon ScholaWrite transfer set. The consistent trend across static and interaction-conditioned evaluation indicates that jointly covering both context forms improves robustness.

5 Human Validation.

5.1 Human Preference Validation and User Study.

Three PhD annotators conduct blind pairwise comparisons on 600 held-out contexts (Fleiss' $\kappa = 0.65$); full protocol is in Appendix H. As shown in Table 3, PCL-Prefix achieves a 62.4% win rate against the base model, and is competitive with GPT-5.4-Nano (30.2% win vs. 22.3% loss), confirming that HAR gains correspond to tangible improvements in human-perceived quality.

User Study. While the pairwise study above validates suggestion quality in isolation, we further conduct a user study in which participants complete full writing sessions comparing Qwen3-8B and PCL-History; detailed protocol is in Appendix H. Table 4 shows that PCL-History more than doubles the real acceptance rate (13.18→28.42), with participants reporting higher satisfaction (2.5→3.8) and fluency (2.1→4.3).

Table 3: Human preference validation. Win/Tie/Loss from the perspective of PCL-Prefix.

Comparison	Win ↑	Tie	Loss ↓
vs. Qwen3-8B	**62.4**	18.7	18.9
vs. GPT-5.4-Nano	30.2	47.5	22.3

Table 4: User study results. Accept: acceptance rate (%); Satisfaction/Fluency: 1–5 Likert.

Model	Accept ↑	Satisfaction ↑	Fluency ↑
Qwen3-8B	13.18	2.5	2.1
+ PCL-History	**28.42**	**3.8**	**4.3**

5.2 Reliability of the LLM Judge.

We validate the LLM judge against human annotations and assess whether the reported conclusions remain stable under alternative judges. Appendix F reports the complete per-instance agreement statistics, inter-annotator agreement, confusion matrices, and cross-judge robustness results. Appendix Figure 16 provides qualitative cases in which the LLM judge and human annotators agree on semantic misalignment while conventional automatic metrics assign misleadingly high scores.

Table 5: External continuation scores (1–5) on arXiv and textbook breakpoints.

Dataset	Model	Local	Intent	Role	Style	Scope	Overall
arXiv	Qwen3-8B	4.042	3.122	3.519	4.262	4.085	3.806
	+PCL-Prefix	4.349	3.364	3.814	4.678	4.461	4.133
	+PCL-History	**4.499**	**3.471**	**3.905**	**4.754**	**4.601**	**4.246**
Textbook	Qwen3-8B	3.906	2.751	3.205	4.068	3.974	3.581
	+PCL-Prefix	**4.617**	3.181	3.714	4.910	4.753	4.235
	+PCL-History	4.607	**3.186**	**3.716**	**4.915**	**4.795**	**4.244**

6 Generalization and Discussion

External Continuation Evaluation. To test out-of-distribution transfer, we construct two external test sets from arXiv papers and textbook passages, where each instance provides a document prefix and the original next sentence as a behavioral reference; full dataset details are in Section G.1. Under the prefix-only setting, we evaluate the base model, PCL-Prefix, and PCL-History with an independent LLM judge (GPT-OSS-120B) along five dimensions: local coherence, intent alignment, discourse-role fit, style consistency, and scope control. Table 5 reports the full scores; Appendix Figure 9 visualizes the same comparison and shows that both PCL variants consistently outperform the base model, especially in intent alignment and discourse-role fit. The judge prompt is in Figure 18.

Emergent Context-Aware Thinking. Process-continuation learning requires the model to understand prior context, assess discourse position, and anticipate what follows. We observe a corresponding *context-aware thinking* pattern in PCL-History reasoning traces. Its complete characterization, quantitative analysis, human validation, and an annotated example are provided in Appendix G.2 and Figure 13.

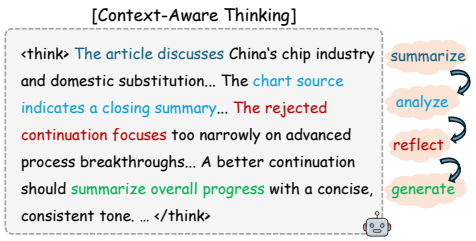


Figure 6: Emergent context-aware thinking. Full analysis and example are in Appendix G.2.

Cross-Domain Generalization. Table 6 tests whether process-continuation learning transfers beyond interactive writing, evaluating the trained policy on mathematical reasoning (Zhang & Team, 2025) and instruction following Zhou et al. (2023). Training on writing data alone yields modest gains on AIME2025 (67.29→67.92) and IFEval (85.03→86.32). Applying the same paradigm to mathematical solution traces produces substantially larger improvements, confirming that the training methodology—sentence-level behavioral optimization over expert traces—generalizes across domains requiring sequential reasoning (case studies in Figures 14 and 15).

Table 6: Cross-domain generalization on mathematical reasoning and instruction following.

Model	AIME2025		IFEval
	avg_pass@1	Pass@32	strict
Qwen3-8B	67.29	86.67	85.03
+ writing data	67.92 (+0.63)	90.00 (+3.33)	86.32 (+1.29)
+ math data	68.85 (+1.56)	93.33 (+6.66)	86.72 (+1.69)

7 Conclusion

We introduced process-continuation learning (PCL), which treats expert continuations as behavioral references rather than unique imitation targets. PCL combines a local-validity-to-semantic-alignment curriculum with prefix-only and history-conditioned trajectories. Across five settings from three benchmark families, PCL-Mix improves Macro HAR by 7.69 points over SFT-Mix and is competitive with closed-source assistants. Human evaluation shows a 62.4% pairwise win rate for PCL-Prefix over the base model; PCL-History raises real-session acceptance from 13.18% to 28.42%. Gains also transfer to unseen academic and textbook continuations, with emerging context-aware thinking and preliminary improvements in mathematics and instruction following. These results support expert continuations as scalable behavioral supervision for anticipating the next appropriate step in evolving contexts.

AI USE STATEMENT

Generative AI is an integral part of our methodology: language models are used for breakpoint segmentation, synthetic interaction-history construction, reward computation, and automatic evaluation, as detailed in the paper. We also used generative AI to assist with research design, literature search and summarization, code implementation and debugging, result interpretation, and technical writing. We did not use it to fabricate or alter experimental results, formulate or prove mathematical claims, or replace the reported human judgments. All AI-assisted content was reviewed and verified by the authors, who take full responsibility for the final work.

ETHICS STATEMENT

Our user study involved ten adult participants, who retained control over whether to accept, revise, or reject every suggestion; we report only aggregate, non-identifying results. The study uses publicly accessible or established research datasets, with openly licensed materials for the external textbook evaluation. Because writing assistants and automatic judges may introduce inaccurate or biased outputs, we validate the primary judge against human labels and view the system as an assistive tool whose outputs require human review.

REPRODUCIBILITY STATEMENT

Section 3 specifies the method, curriculum reward, and trajectory construction, while Section 4.1 describes the experimental protocol and baselines. The appendix documents data processing (Appendix B), training details and hyperparameters (Appendix C), benchmark construction and metrics (Appendices D.1–D.2), complete ablations and robustness checks (Appendix E), and all reward and evaluation prompts (Appendix I). We also report sample sizes, confidence intervals, and results over four stochastic evaluation runs where applicable.

REFERENCES

Art of Problem Solving. Art of problem solving. `https://artofproblemsolving.com/`, 2026. Accessed: 2026-05-13.

Joshua Ashkinaze, Julia Mendelsohn, Li Qiwei, Ceren Budak, and Eric Gilbert. How ai ideas affect the creativity, diversity, and evolution of human ideas: evidence from a large, dynamic experiment. In *Proceedings of the ACM collective intelligence conference*, pp. 198–213, 2025.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Shreyas Chaudhari, Pranjal Aggarwal, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, Karthik Narasimhan, Ameet Deshpande, and Bruno Castro da Silva. Rlhf deciphered: A critical analysis of reinforcement learning from human feedback for llms. *ACM Computing Surveys*, 58 (2):1–37, 2025.

Elizabeth Clark, Anne Spencer Ross, Chenhao Tan, Yangfeng Ji, and Noah A Smith. Creative writing with a machine in the loop: Case studies on slogans and stories. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces*, pp. 329–340, 2018.

Andy Coenen, Luke Davis, Daphne Ippolito, Emily Reif, and Ann Yuan. Wordcraft: a human-ai collaborative editor for story writing. *arXiv preprint arXiv:2107.07430*, 2021.

Nicolas Devatine and Louis Abraham. Assessing human editing effort on llm-generated texts via compression-based edit distance. *arXiv preprint arXiv:2412.17321*, 2024.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization, 2024. *URL https://arxiv. org/abs/2402.01306*, 14, 2023.

Youqing Fang, Yinhao Tang, Yanan Sun, Jiangning Liu, Ziyi Wang, Xun Zhao, Bin Liu, Weiming Zhang, Kuikun Liu, Wenwei Zhang, and Kai Chen. Mindcopilot: Towards formalizing and evaluating human-llm co-writing. In *Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*, 2026. arXiv:2605.23535.

Google DeepMind. Gemini flash lite. https://deepmind.google/models/gemini/flash-lite/, 2025. Accessed: 2025.

Jessica He, Stephanie Houde, and Justin D Weisz. Which contributions deserve credit? perceptions of attribution in human-ai co-creation. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pp. 1–18, 2025.

Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 11170–11189, 2024.

Jacob Jackson, Phillip Kravtsov, and Shomil Jain. Improving Cursor Tab with online RL. https://cursor.com/blog/tab-rl, September 2025. Cursor Blog. Accessed: 2026-05-18.

Seungjae Jung, Gunsoo Han, Daniel Wontae Nam, and Kyoung-Woon On. Binary classifier optimization for large language model alignment. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1858–1872, 2025.

Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. *arXiv preprint arXiv:2312.14925*, 2023.

Khanh Chi Le, Linghe Wang, Minhwa Lee, Ross Volkov, Luan Tuyen Chau, and Dongyeop Kang. ScholaWrite: A dataset of end-to-end scholarly writing process. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 34755–34788. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1606. URL https://aclanthology.org/2026.acl-long.1606/.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Ren Lu, Thomas Mesnard, Johan Ferret, Colton Bishop, Ethan Hall, Victor Carbune, and Abhinav Rastogi. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. 2023.

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

Jessy Lin and Nick Tomlin. User simulators bridge rl with real-world interaction. https://jessylin.com/2025/07/10/user-simulators-1/, July 2025a. Accessed: 2026-05-17.

Jessy Lin and Nick Tomlin. What does it take to build a human-like user simulator? https://jessylin.com/2025/09/25/user-simulators-2/, September 2025b. Accessed: 2026-05-17.

OpenAI. Introducing GPT-5.4 mini and nano. https://openai.com/index/introducing-gpt-5-4-mini-and-nano/, 2025. Accessed: 2025.

OpenAI. GPT-5.6 Luna. https://developers.openai.com/api/docs/models/gpt-5.6-luna, 2026. Accessed: 2026.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Cheng Qian, Zuxin Liu, Akshara Prabhakar, Jielin Qiu, Zhiwei Liu, Haolin Chen, Shirley Kokane, Heng Ji, Weiran Yao, Shelby Heinecke, et al. Userrl: Training interactive user-centric agent via reinforcement learning. *arXiv preprint arXiv:2509.19736*, 2025.

Qwen Team. Qwen3-embedding-0.6b. https://huggingface.co/Qwen/Qwen3-Embedding-0.6B, 2025. Hugging Face model.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Raghav Sharma, Manan Mehta, and Sai Tiger Raina. Rlhf: A comprehensive survey for cultural, multimodal and low latency alignment methods. In *International Conference on Computing and Communication Networks*, pp. 230–246. Springer, 2025.

Hua Shen and Tongshuang Wu. Parachute: Evaluating interactive human-lm co-writing systems. *arXiv preprint arXiv:2303.06333*, 2023.

Neomy Storch. Collaborative writing. *Language Teaching*, 52(1):40–59, 2019.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. arxiv 2023. *arXiv preprint arXiv:2302.13971*, 10, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Daijin Yang, Yanpeng Zhou, Zhiyuan Zhang, Toby Jia-Jun Li, and Ray Lc. Ai as an active writer: Interaction strategies with generated text in human-ai collaborative fiction writing. In *Joint proceedings of the ACM IUI workshops*, volume 10, pp. 1–11. CEUR-WS Team, 2022.

Shuning Zhang, Hui Wang, and Xin Yi. Exploring collaboration patterns and strategies in human-ai co-creation through the lens of agency: A scoping review of the top-tier hci literature. *Proceedings of the ACM on Human-Computer Interaction*, 9(7):1–43, 2025.

Yifan Zhang and Math-AI Team. American invitational mathematics examination (aime) 2025. `https://huggingface.co/datasets/math-ai/aime25`, 2025. Hugging Face dataset.

Zhihu. Zhihu. `https://www.zhihu.com`, 2026. Accessed: 2026-05-26.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

Yifei Zhou, Song Jiang, Yuandong Tian, Jason Weston, Sergey Levine, Sainbayar Sukhbaatar, and Xian Li. Sweet-rl: Training multi-turn llm agents on collaborative reasoning tasks. *arXiv preprint arXiv:2503.15478*, 2025.

A LIMITATIONS

Suggestion timing. The current framework generates a suggestion at every writing step, regardless of the policy's confidence or the informativeness of the current context. A more capable writing assistant should decide *when* to suggest, not only *what* to suggest: withholding low-confidence proposals preserves the writer's flow state, while reserving suggestions for moments of high contextual signal tends to yield higher-quality continuations and better acceptance rates. Extending process-continuation learning with a suggestion-timing module—one that conditions on contextual richness and estimated generation confidence before triggering a proposal—is a natural direction for future work.

Scope of writing operations. The present work focuses exclusively on proactive continuation suggestions—predicting the writer's next sentence given a document prefix. A complete writing assistant, however, must support a broader range of editorial operations, including rewriting, compression, expansion, and polishing of existing content. These operations differ fundamentally from continuation in that they act on already-written text rather than extending it, and may require different reward formulations and training objectives. Extending the process-continuation framework to cover this wider spectrum of writing assistance remains an open direction.

B DATA AND REPRODUCIBILITY

Source. Training data comes from two domains. The writing corpus is drawn from Zhihu Zhihu (2026), a long-form writing platform, collecting articles from 30 high-quality authors spanning over 20 topic categories, including technology, finance, education, science communication, psychology, history, and sports. Raw articles are first filtered by engagement (upvotes > 10), reducing the pool from 9,250 to 1,250 articles. A difficulty filter is then applied: we run GPT-OSS-120B inference on each candidate snippet state and retain only instances with an acceptance rate below 0.4, keeping examples that require substantive contextual understanding rather than shallow local completion. The math corpus is collected from the Art of Problem Solving platform Art of Problem Solving (2026), comprising worked solutions to competition mathematics problems. Table 7 summarizes both corpora.

Table 7: Training corpus statistics.

Corpus	Source	Items	Snippets
Writing	Zhihu	760 articles	45,120
Math	AoPS	1,288 problems	30,050

Breakpoint segmentation. Each article is first cleaned (removing duplicate content, image-linked text, and formatting noise) and converted to plain markdown. Breakpoints are then inserted in two stages: paragraph-level boundaries are identified at newline positions, and an LLM subsequently inserts finer-grained breakpoints within each paragraph to produce snippet-level writing traces (Figure 7). Each breakpoint defines a training instance: the document prefix up to the breakpoint serves as the writing state s_t, and the following snippet serves as the behavioral reference a_t^* (Figure 3).

Context-variant trajectory. As illustrated in Figure 3 and Figure 7, each breakpoint in the training corpus yields two distinct training instances that differ in what the policy observes as the writing state s_t. In the **prefix-only** setting (PCL-Prefix), s_t contains only the document prefix c_t up to the breakpoint—the model conditions solely on the human-written context and generates the next snippet independently. In the **history-conditioned** setting (PCL-History), $s_t = (c_t, h_t)$ additionally includes an interaction history $h_t = \{(\tilde{a}_j, \tilde{u}_j)\}_{j=1}^{m}$, where each pair records a prior assistant suggestion and the corresponding user-agent response. Since naturally occurring documents do not contain real multi-turn interactions, h_t is synthesized offline by sampling auxiliary continuations for the same writing state and collecting user-agent feedback; the behavioral reference a_t^* remains grounded in the original human-written trace throughout. Both settings share the same curriculum reward, ensuring that the two trajectory types are complementary rather than competing objectives.

Interaction history encoding. To encode the interaction history h_t in history-conditioned contexts, we use coarse-grained tags that record suggestion-level feedback. The `<accept>` tag marks content

retained by the user, <reject> marks a rejected continuation, and <add> marks the user's subsequent revision. Table 8 summarizes these tags, and the example below shows how they encode the last interaction record in Figure 7.

Encoding example.

<accept> soning </accept> <reject> under multi-token prediction can generate code with lower latency. </reject> <add> tasks**, multi-token prediction also brings strong out-of-distribution gains. </add>

Table 8: Interaction tag definitions.

Tag	Semantics
<accept>	The user retained the model's previous continuation snippet.
<reject>	The user rejected the model's previous continuation snippet.
<add>	Content newly added by the user.

Table 9: Hyperparameters for Training.

Hyperparameters	Setting
Global batch size	256
Rollouts per prompt	16
Max response length	32K
Learning rate	1e−6
KL loss	off

C Reward and Optimization Analysis

LLM backbone. During training, the local coherence reward judge (r_{loc}) and the semantic alignment reward judge (r_{sem}) are both implemented using GPT-5.6 Luna (OpenAI, 2026). We use the same model as the primary evaluation judge for computing HAR and KED, following the evaluation protocol of Fang et al. (2026). Because the reward and evaluation judges share a backbone, we validate its agreement with human judgments and test cross-judge robustness in Section F.

Optimization algorithm. We optimize the policy using Group Relative Policy Optimization (GRPO; Shao et al. 2024), a variant of policy gradient that estimates advantages within a group of rollouts sampled from the same prompt, eliminating the need for a separate value network. Concretely, for each writing context, we sample 16 candidate continuations from the current policy; their scalar rewards are normalized within the group to produce relative advantages, which are then used to compute the clipped policy-gradient objective. This group-wise normalization provides a low-variance baseline that is well-suited to our setting, where reward magnitude varies considerably across writing contexts.

Curriculum transition. We monitor the smoothed local-validity training reward and begin the semantic-alignment stage once this reward has plateaued. In our experiments, this criterion selected the checkpoint at step 220, from which training was resumed with the semantic-alignment objective; the length penalty remained active in both stages.

Embedding-reward baseline. The embedding-reward baseline uses Qwen3-Embedding-0.6B (Qwen Team, 2025) to encode the generated continuation and its behavioral reference. Their embedding similarity is used directly as the scalar reward under the same prefix-only GRPO setup as the other single-objective baselines.

Hyperparameters. All RL stages share the same optimizer configuration: learning rate 1e−6, and the KL penalty against the reference model is disabled. The remaining stage-specific hyperparameters are listed in Tab. 9. For the length penalty (Eq. equation 8), we set the tolerance threshold $\delta = 30$ tokens and the attenuation factor $\gamma = 0.5$.

Validation rejection reason analysis. To complement the reward and acceptance curves in Section 4.3, Figure 12 analyzes the HAR outcomes of the best Round-1 validation checkpoint, Local Validity→Semantic Alignment/Mix, on the 1,688-example single-turn validation set. The model accepts 465 completions (27.5%) and rejects 1,223 (72.5%). Among rejected completions,

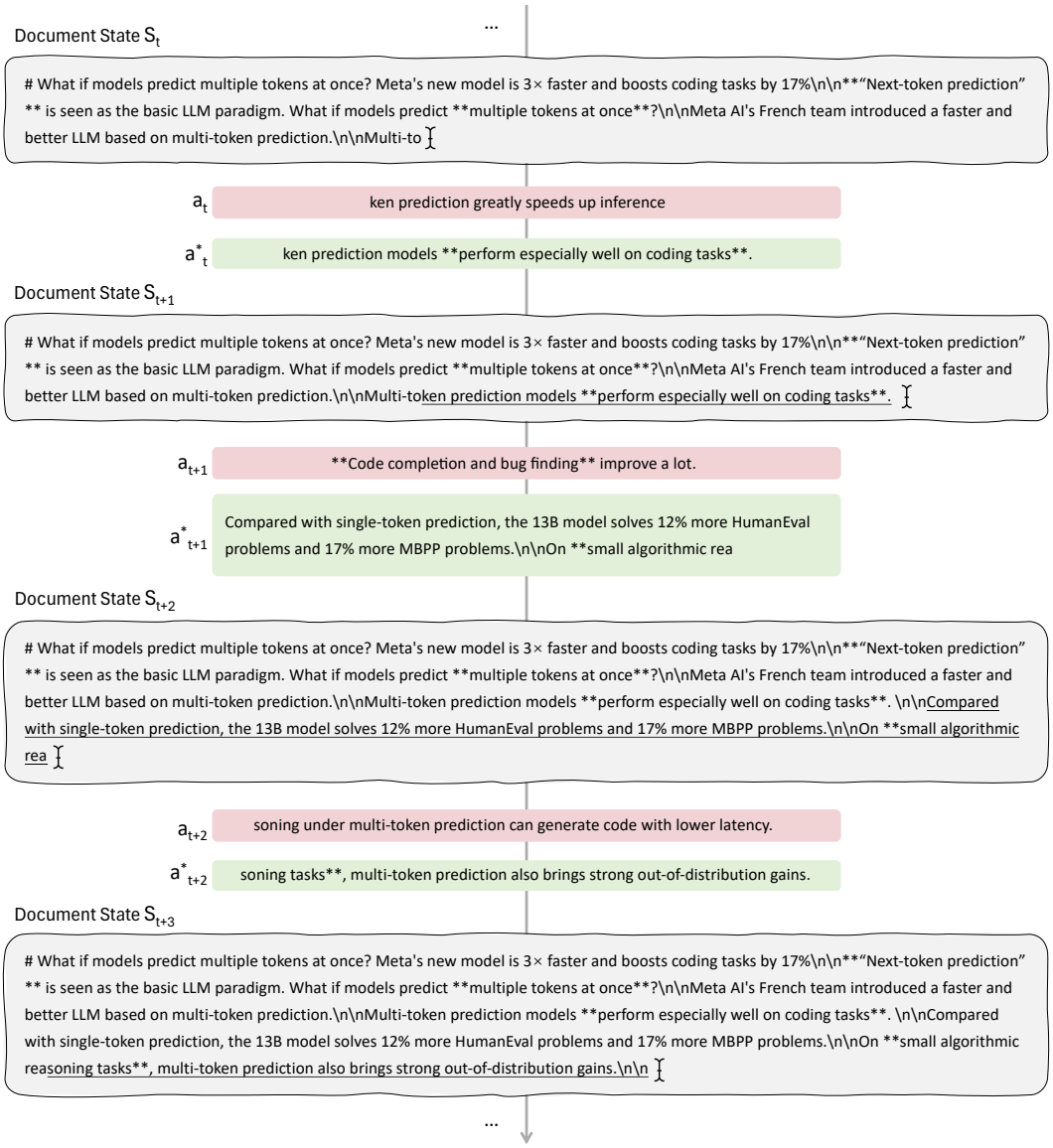


Figure 7: Training data construction. Raw articles are cleaned and segmented by inserting breakpoints (shown as gray cursor blocks). The prefix-only context (PCL-Prefix) exposes the model to the document prefix up to each breakpoint. The history-conditioned context (PCL-History) additionally incorporates the full interaction history at each breakpoint, including prior model suggestions and user feedback (shown as red and green blocks).

Key Entities is the largest failure category (41.5%), followed by *Format Mismatch* (18.3%), *Topic Divergence* (9.5%), *Depth Mismatch* (8.4%), and *Intent Mismatch* (7.2%). Together, the entity-, topic-, depth-, and intent-related categories account for 66.6% of all rejections, whereas *Start Repetition* accounts for only 0.5%. This distribution indicates that the strongest checkpoint has largely learned to produce genuine continuations, while preserving reference-specific entities and matching the intended semantic scope remain the principal bottlenecks.

D Evaluation Benchmarks and Metrics

D.1 Benchmark Construction

We evaluate all models under a unified continuation protocol: the model observes only the context available at the breakpoint, generates a single candidate continuation, and is never given the reference continuation. Table 10 summarizes the resulting evaluation sets.

Table 10: Evaluation datasets used under the unified continuation protocol. CoAuthor counts correspond to writer-held-out test sets; ScholaWrite uses a project-held-out test set.

Dataset	Source	Test instances	Context
MindCopilot-Single	MindCopilot	1,688	document prefix
MindCopilot-Multi	MindCopilot	1,688	prefix + interaction history
CoAuthor-Human	CoAuthor	13,633	prompt + document prefix
CoAuthor-Interactive	CoAuthor	7,469	prompt + request-time prefix
ScholaWrite-Continuation	ScholaWrite	235	manuscript prefix

MindCopilot. We retain the single-turn and multi-turn evaluation settings introduced by Fang et al. (2026). The single-turn setting conditions on the current document prefix, while the multi-turn setting additionally exposes prior suggestion–feedback interactions. Both sets contain 1,688 evaluation states and use the same generation and judging configuration as the external benchmarks below.

CoAuthor-Human. CoAuthor records keystrokes, cursor movements, model-suggestion requests, and subsequent edits from 1,445 English creative and argumentative writing sessions (Lee et al., 2022). We replay the Quill edit operations to reconstruct the document at each event while separately tracking characters inserted by the writer and by the original writing assistant. We retain states whose cursor is at the end of the current prose, whose prefix ends at a complete-sentence boundary, and whose next stable sentence remains in the final session text. The target in CoAuthor-Human must be composed entirely of writer-inserted characters; deleted continuations, incomplete sentences, ambiguous alignments, and duplicate state–target pairs are removed. We split by writer so that no writer appears in both development and test sets, yielding 5,444 development and 13,633 test instances.

CoAuthor-Interactive. This subset replays the document state at each real suggestion-request event. Starting from the exact request-time prefix, we align the state with the final session text and extract the first stable sentence realized after that point. The target is the final retained continuation rather than the original GPT-3 suggestion; it may therefore be written by the author, adapted from an accepted suggestion, or combine both. Original accept/reject labels are retained only as metadata and are never transferred to a newly generated candidate. Applying the same writer-disjoint split gives 1,594 development and 7,469 test instances. Consequently, this benchmark measures offline compatibility with the realized writing trajectory, not actual user reactions to the evaluated model.

ScholaWrite-Continuation. ScholaWrite contains 61,504 timestamped LaTeX editing actions from five computer-science manuscripts, each annotated with a writing intention (Le et al., 2026). We reconstruct the complete event stream and retain *Text Production* events that are pure insertions at a complete-sentence boundary. Consecutive snapshots from the same project, author, and insertion anchor are merged into a writing burst when they are separated by at most 120 seconds and do not modify the pre-existing prefix. From the final state of each burst, we extract the first complete newly written sentence, discarding bursts without a complete sentence, extreme multi-paragraph pastes, structural LaTeX targets, and duplicate prefix–target pairs. The official event-level test split preserves only 20.1% of eligible adjacent burst edges, so we reconstruct trajectories before splitting and use a project-held-out partition: one project for development and the remaining four for test. This produces 63 development and 235 test instances. The set includes both document-end continuations and transition sentences inserted before existing prose; in both cases the model observes only the prefix preceding the insertion point.

Interpretation and data safeguards. CoAuthor and ScholaWrite provide naturally occurring references but no user responses to the newly evaluated model outputs. We therefore describe HAR as an *offline* acceptance estimate and KED as a reference-conditioned editing-cost proxy. ScholaWrite is accessed under its official data-use terms; raw trajectories and reconstructed samples are not redistributed. Because the source manuscripts or CoAuthor texts may occur in model pretraining corpora, incomplete decontamination remains a limitation.

D.2 Evaluation Metrics

Hierarchical Acceptance Rate (HAR). HAR estimates how often a model's suggestion would satisfy the hierarchical acceptance criterion of Fang et al. (2026), conditioned on the observed writing state and reference continuation. A suggestion $\tilde{a}$ is accepted only if it passes the interaction-, surface-, and semantic-level criteria implemented by the evaluation judge. Formally, let $y_i \in \{0, 1\}$ denote the acceptance label for the i-th suggestion. HAR is then the proportion of accepted suggestions:

$$\mathrm{HAR} = \frac{1}{N} \sum_{i=1}^{N} y_i. \tag{13}$$

Knowledge-weighted Edit Distance (KED). KED estimates the post-acceptance revision required to transform an accepted suggestion into the reference continuation, weighted by the information density of the edited tokens (Fang et al., 2026). We compute KED only over the subset $\mathcal{A} = \{i : y_i = 1\}$ accepted by HAR. A lower value indicates that accepted suggestions are closer to the realized continuation, but KED must be interpreted jointly with HAR because $\mathcal{A}$ differs across models. Let d_i denote the edit distance for accepted suggestion i and w_i its knowledge weight. Then:

$$\mathrm{KED}_{\mathrm{accepted}} = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} w_i \cdot d_i. \tag{14}$$

Cross-benchmark aggregation. We first report HAR, the number of accepted instances, and accepted-only KED separately for each of the $B = 5$ evaluation sets. To summarize performance, Macro HAR gives each benchmark equal weight,

$$\mathrm{HAR}_{\mathrm{macro}} = \frac{1}{B} \sum_{b=1}^{B} \mathrm{HAR}_b, \tag{15}$$

whereas sample-weighted HAR pools all decisions,

$$\mathrm{HAR}_{\mathrm{weighted}} = \frac{\sum_{b=1}^{B} \sum_{i=1}^{N_b} y_{b,i}}{\sum_{b=1}^{B} N_b}. \tag{16}$$

Because the two CoAuthor sets account for most evaluation instances, the weighted score is dominated by CoAuthor; we therefore treat Macro HAR as the primary balanced summary and report both quantities. Macro KED averages the five accepted-only KED values, but remains auxiliary because each model induces a different accepted subset. We additionally report accepted counts to make this selection effect visible.

E Supplementary Experimental Results

This section follows the order of the main experiments: it first provides the original-scale reward dynamics for the reward-design ablation in Section 4.3, then reports the complete results for the displayed SFT and process-continuation variants and the focused trajectory-composition analysis underlying Section 4.4.

E.1 Original-Scale Training Reward

The main-text reward-dynamics figure compares all five reward configurations over the common 420-step training horizon. Figure 8 shows the corresponding training rewards on their original, unnormalized scale.

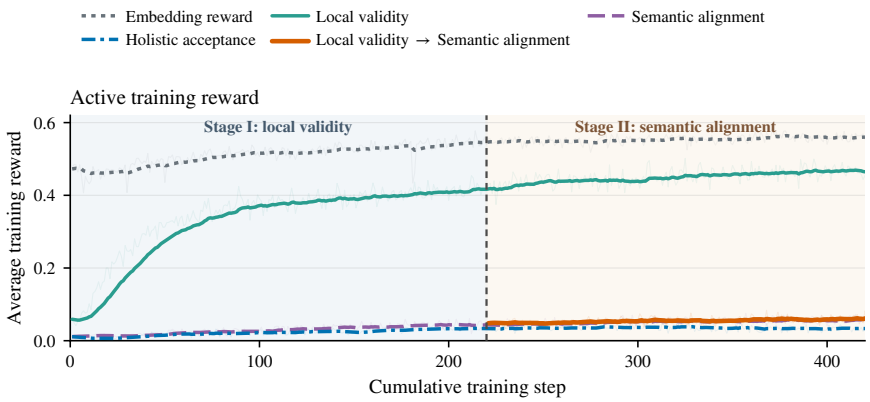


Figure 8: Training-reward dynamics under the five reward configurations over the common 420-step horizon. Thin lines show raw observations and thick lines show EMA-smoothed trends. Rewards are reported on their original linear scale, without the stage-wise normalization used in Figure 4. The vertical dashed line marks the plateau-triggered transition from local-validity training to semantic-alignment training at step 220.

E.2 Complete Training-Variant Results

Table 11 reports the Qwen3-8B base model and twelve trained variants evaluated under the unified protocol of Table 1: three SFT controls and nine process-continuation policies. The first two columns expose the training objective or reward schedule and the trajectory source, while the remaining columns report per-benchmark HAR with confidence intervals.

Table 11: Complete per-benchmark HAR results for the Qwen3-8B base model and twelve trained variants, comprising three SFT controls and nine process-continuation settings. HAR is reported as mean $\pm$ 95% confidence interval, computed as $1.96\sqrt{p(1-p)/N_{\text{eff}}}$ with $N_{\text{eff}} = 4N$ pooled across four evaluation runs per benchmark. Mixed trajectories use an equal mixture of single-turn and multi-turn training instances. **Bold** marks the best HAR and underline the second-best HAR among the displayed variants.

HAR (%) ↑, mean ± 95% confidence interval						
Objective / Reward	Trajectory	MindCopilot Single-turn	MindCopilot Multi-turn	CoAuthor Human	CoAuthor Interactive	ScholaWrite Continuation
Qwen3-8B	Base	10.31 ± 0.73	8.99 ± 0.68	13.78 ± 0.29	17.28 ± 0.43	9.27 ± 1.85
SFT	Single-turn	**27.96** ± 1.07	14.57 ± 0.84	34.91 ± 0.40	38.28 ± 0.55	22.13 ± 2.65
SFT	Multi-turn	22.04 ± 0.99	13.27 ± 0.81	31.81 ± 0.39	33.32 ± 0.53	20.00 ± 2.56
SFT	Mixed (1:1)	27.49 ± 1.06	13.98 ± 0.83	39.37 ± 0.41	39.95 ± 0.55	21.28 ± 2.62
Embedding	Single-turn	12.26 ± 0.78	3.97 ± 0.47	22.58 ± 0.35	25.14 ± 0.49	10.21 ± 1.94
Local validity	Single-turn	22.81 ± 1.00	17.42 ± 0.90	46.96 ± 0.42	49.42 ± 0.57	<u>26.81</u> ± 2.83
Semantic alignment	Single-turn	19.91 ± 0.95	15.58 ± 0.87	38.87 ± 0.41	41.65 ± 0.56	20.00 ± 2.56
Holistic acceptance	Single-turn	23.10 ± 1.01	17.00 ± 0.90	44.03 ± 0.42	46.79 ± 0.57	20.85 ± 2.60
Local validity → Semantic alignment	Single-turn	25.53 ± 1.04	<u>19.02</u> ± 0.94	47.38 ± 0.42	50.33 ± 0.57	25.96 ± 2.80
Local validity	Multi-turn	21.62 ± 0.98	16.05 ± 0.88	44.92 ± 0.42	47.28 ± 0.57	26.38 ± 2.82
Local validity → Semantic alignment	Multi-turn	22.10 ± 0.99	18.60 ± 0.93	46.39 ± 0.42	48.80 ± 0.57	23.83 ± 2.72
Local validity	Mixed (1:1)	24.76 ± 1.03	17.18 ± 0.90	<u>48.74</u> ± 0.42	<u>51.02</u> ± 0.57	25.53 ± 2.79
Local validity → Semantic alignment	Mixed (1:1)	<u>27.55</u> ± 1.07	**19.96** ± 0.95	**49.74** ± 0.42	**52.62** ± 0.57	**30.64** ± 2.95

The SFT single-turn control attains the highest HAR on MindCopilot Single-turn, while the mixed-trajectory Local-validity → Semantic-alignment policy performs best on the other four benchmarks. This pattern highlights the cross-benchmark benefit of combining trajectory diversity with curriculum learning.

E.3 Complete Results on Training-Trajectory Composition

Table 12 expands the trajectory-composition comparison in Section 4.4. The mixed training pool samples single-turn and multi-turn trajectories with equal probability (1:1). All variants use the same model and evaluation protocol; we compare the local-validity first stage with the complete local-validity-to-semantic-alignment curriculum while varying only the trajectory source.

Table 12: Complete HAR results for the trajectory-composition study. Panel A reports absolute HAR (%) on each benchmark and the two aggregate measures. Panel B reports the change after introducing the semantic stage, in percentage points. MCB-S/M denote the single-/multi-turn MindCopilot sets, and CA-H/I denote CoAuthor-Human/Interactive. Macro HAR weights the five benchmarks equally, whereas Weighted HAR pools all evaluation instances.

Panel A: Absolute HAR (%)

Trajectory	Reward schedule	MCB-S	MCB-M	CA-H	CA-I	ScholaWrite	Macro	Weighted
Single-turn	Local validity	22.81	17.42	46.96	49.42	26.81	32.68	43.84
	Local validity → Semantic alignment	25.53	19.02	47.38	50.33	25.96	33.64	44.64
Multi-turn	Local validity	21.62	16.05	44.92	47.28	26.38	31.25	41.89
	Local validity → Semantic alignment	22.10	18.60	46.39	48.80	23.83	31.94	43.35
Mixed (1:1)	Local validity	24.76	17.18	48.74	51.02	25.53	33.45	45.42
	Local validity → Semantic alignment	**27.55**	**19.96**	**49.74**	**52.62**	**30.64**	**36.10**	**46.88**

Panel B: Semantic-stage gain, △(Local validity → Semantic alignment − Local validity), percentage points

Trajectory	MCB-S	MCB-M	CA-H	CA-I	ScholaWrite	Macro	Weighted
Single-turn	+2.72	+1.60	+0.42	+0.91	−0.85	+0.96	+0.80
Multi-turn	+0.48	+2.55	+1.47	+1.52	−2.55	+0.69	+1.46
Mixed (1:1)	**+2.79**	**+2.78**	+1.00	**+1.60**	**+5.11**	**+2.65**	+1.46

The mixed setting is the only trajectory configuration for which the semantic stage improves HAR on all five benchmarks. Its Macro-HAR gain is also the largest ($+2.65$ points), with the strongest improvement on ScholaWrite ($+5.11$). In contrast, the single-turn and multi-turn configurations lose 0.85 and 2.55 points on ScholaWrite, respectively. This per-benchmark consistency supports the interpretation that prefix-only and history-conditioned trajectories provide complementary supervision. The weighted gain is identical for multi-turn and mixed training ($+1.46$) because the weighted aggregate is dominated by the two large CoAuthor sets; we therefore base the curriculum-interaction claim on balanced Macro HAR and cross-benchmark consistency.

F Judge Reliability and Statistics

We use GPT-5.6 Luna for both the training rewards and the primary HAR/KED evaluation. Because sharing a judge across optimization and evaluation may amplify judge-specific preferences, we assess its reliability in two complementary ways: per-instance agreement with human judgments and a cross-judge robustness check on the external evaluation. Qwen3-235B and GPT-OSS-120B are used here as comparison judges rather than as training reward models.

(i) The primary judge is validated against a shared human reference. On HAR, **GPT-5.6 Luna** reaches $\kappa = 0.779$ **and 90.33% per-instance agreement** with the human gold labels. Table 13 reports the full confusion matrices and agreement statistics for GPT-5.6 Luna, Qwen3-235B, and GPT-OSS-120B across HAR, semantic alignment, and local coherence. The reference set contains **600 instances**, each independently labeled accept/reject by **three annotators**, with majority vote used as the gold label. Figure 16 complements these aggregate statistics with examples where the LLM judge and human annotators identify semantic misalignment that conventional overlap- and embedding-based metrics fail to capture.

(ii) The human reference is itself reliable. The three annotators achieve Fleiss' κ values of 0.696–0.704 across the three dimensions, with unanimous labels on approximately 79% of instances (Table 14). These statistics establish that the comparison is grounded in a consistently labeled human reference rather than in agreement with a noisy aggregate.

(iii) The external-transfer conclusion is robust to judge choice. We re-score the same external-evaluation outputs with GPT-OSS-120B, Qwen3-235B, and GPT-5.6 Luna. The ordering **base** $<$ **PCL-Prefix and base** $<$ **PCL-History** holds in all six judge$\times$domain comparisons (Table 15). Thus, although GPT-5.6 Luna is shared by reward optimization and the primary evaluation, the observed external-transfer gains do not depend on its preferences alone. This judge-swap experiment specifically supports the generalization result; the comparison between PCL-Prefix and PCL-History is established by the in-domain evaluation.

Tables 13–15 report the complete agreement and robustness statistics.

Table 13: Judge vs. human agreement on 600 instances with majority-vote gold labels. Columns report counts of judge decisions cross-tabulated with the human gold label.

Dimension (gold acc./rej.)	Judge	Acc ∩ Acc	Acc ∩ Rej	Rej ∩ Acc	Rej ∩ Rej	Alignment	Cohen's κ
HAR (200/400)	GPT-5.6 Luna	165	23	35	377	90.33%	0.779
	Qwen3-235B	136	20	64	380	86.00%	0.667
	GPT-OSS-120B	146	95	54	305	75.17%	0.469
Semantic (223/377)	GPT-5.6 Luna	190	60	33	317	84.50%	0.676
	Qwen3-235B	188	83	35	294	80.33%	0.597
	GPT-OSS-120B	142	85	81	292	72.33%	0.410
Coherence (256/344)	GPT-5.6 Luna	228	42	28	302	88.33%	0.763
	Qwen3-235B	229	40	27	304	88.83%	0.773
	GPT-OSS-120B	222	45	34	299	86.83%	0.732

Table 14: Human inter-annotator agreement.

Dimension	Fleiss' κ	Avg. pairwise Cohen's κ	Unanimous (3/3)	Gold split (acc./rej.)
HAR	0.701	0.703	79.33%	200 / 400
Semantic	0.696	0.698	79.33%	223 / 377
Coherence	0.704	0.706	78.50%	256 / 344

Table 15: Judge-swap robustness of the external evaluation (mean score).

Setting	GPT-OSS-120B	Qwen3-235B	GPT-5.6 Luna
arXiv (100 papers)			
Qwen3-8B	3.806	3.532	3.643
+PCL-Prefix	4.133	3.895	3.971
+PCL-History	**4.246**	**4.013**	**4.135**
Textbook (100 books)			
Qwen3-8B	3.581	3.619	3.439
+PCL-Prefix	4.235	3.882	3.827
+PCL-History	**4.244**	**3.957**	**4.083**

G Generalization Analysis

G.1 External Continuation Evaluation

To test whether the learned behavior transfers to out-of-distribution domains, we construct two external test sets from arXiv papers and textbook passages. Each instance provides a document prefix and an original next sentence as the behavioral reference; models generate continuations under the prefix-only setting. An independent LLM judge scores each continuation along five dimensions (local coherence, intent alignment, discourse-role fit, style consistency, and scope control) using the prompt in Figure 18. Figure 9 summarizes the results as radar charts; Table 5 in the main text reports full per-dimension scores. The following paragraphs describe how each test set is constructed.

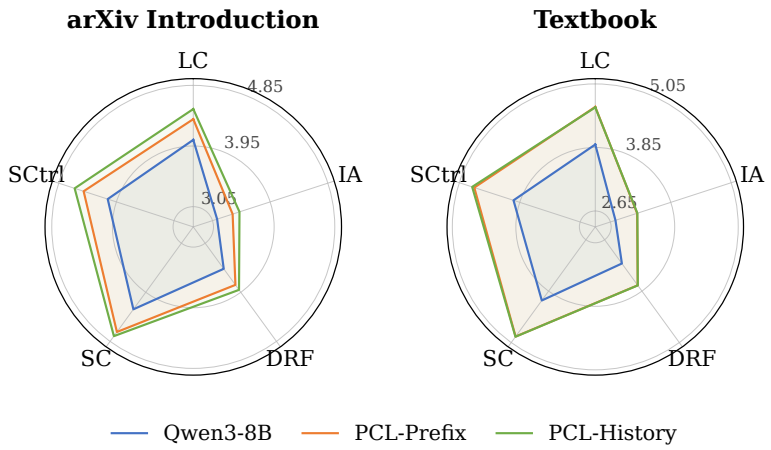


Figure 9: External continuation evaluation on arXiv introductions and textbook, along five dimensions: LC (local coherence), IA (intent alignment), DRF (discourse-role fit), SC (style consistency), SCtrl (scope control); axes are scaled per subplot. Both RL variants outperform the base model, most notably on IA and DRF.

arXiv test set construction. Breakpoints are sampled from the introduction sections of arXiv papers, which are well-suited for testing argumentative continuation: the next sentence typically serves a clear rhetorical role such as identifying a problem gap, articulating a challenge, providing motivation, pointing out a limitation of prior work, conveying method intuition, or bridging to a contribution. We retain instances where (1) the breakpoint falls within the middle of an introduction (not the first or last sentence), (2) the prefix already establishes research background or problem setup, and (3) the reference is a complete one or two sentences playing an identifiable discourse role. We exclude instances where the reference consists primarily of citation markers, mathematical symbols or theorem numbers, or content that depends heavily on figures or tables; we also exclude cases where the reference is a specific item from a contribution list, or where the method/model/dataset name appears in the reference but is not yet introduced in the prefix. Papers are sampled across NLP, ML, CV, and AI subfields to ensure breadth; Table 16 reports the distribution.

Textbook test set construction. Textbook passages are particularly well-suited for process-continuation evaluation because the next step in an expository sequence is typically predictable in function: explaining a concept, providing an example, introducing a formula, walking through a procedure, or transitioning from intuition to formal definition. This structured discourse pattern aligns naturally with the behavioral continuation objective, and the clear rhetorical progression makes it straightforward for the LLM judge to assess intent alignment. All textbooks are drawn from open educational materials, open courseware, and openly licensed documentation to avoid copyright concerns. Table 17 reports the category distribution.

Table 16: arXiv test set composition.

Subject Area	**Papers**
Computer Science	20
Economics	10
Electrical Engineering & Systems Science	20
Mathematics	10
Physics	10
Quantitative Biology	10
Quantitative Finance	10
Statistics	10
Total	**100**

Table 17: Textbook test set composition.

Subject Category	**Books**
Natural Sciences	20
Mathematics & Formal Sciences	20
Engineering & Technology	10
Humanities & Social Sciences	10
Language & Literature	10
Medicine & Life Sciences	10
Arts & Design	10
General & Interdisciplinary	10
Total	**100**

G.2 Context-Aware Thinking: Frequency, Association, and Measurement

We operationalize *context-aware thinking* (CAT) as four observable behaviors in the model's <think> trace: **S1**, summarizing the preceding document and interaction context; **S2**, identifying the discourse role that the next sentence should serve; **S3**, reflecting on prior user feedback or on why a previous suggestion was rejected; and **S4**, planning the content or form of the continuation before generating it. We call a trace *full-CAT* when all four behaviors are present. Figure 13 provides a fully annotated example.

Frequency across training. We apply the automatic CAT detector to all $N = 1{,}688$ traces in the stateful MindCopilot-Multi test set at each of five PCL-History checkpoints. Table 18 shows that discourse-role identification (S2) increases from 37.1% to 84.2% and feedback reflection (S3) from 27.7% to 88.7%; full-CAT correspondingly rises from 10.3% to 63.0%. In contrast, context summarization (S1) and pre-generation planning (S4) are already frequent at step 0 (86.2% and 97.3%). We therefore describe only S2 and S3 as emergent over training, rather than attributing emergence to all four components.

Table 18: Frequency (%) of context-aware thinking behaviors across PCL-History checkpoints on MindCopilot-Multi ($N = 1{,}688$ traces per checkpoint). HAR is included as a training reference.

Behavior	Step 0	Step 50	Step 100	Step 150	Step 200
S1: Context summarization	86.2	90.3	94.6	97.6	97.1
S2: Discourse-role identification	37.1	46.9	52.7	69.8	**84.2**
S3: Feedback reflection	27.7	34.1	56.1	81.4	**88.7**
S4: Pre-generation planning	97.3	98.4	98.0	99.2	98.8
Full-CAT (S1–S4)	10.3	12.5	21.2	56.9	**63.0**
HAR	8.99	13.70	16.76	20.11	22.39

Association with HAR. To avoid confounding CAT frequency with training progress, we compare traces only within the step-200 checkpoint. As Table 19 shows, full-CAT traces attain 26.9% HAR, compared with 14.7% for traces without full-CAT: a difference of 12.1 percentage points (95% CI $[8.3, 16.0]$). The association is statistically significant ($\chi^2(1) = 33.3$, $p < 10^{-8}$), with an odds ratio of 2.13 (95% CI $[1.64, 2.76]$). Thus, a full-CAT trace is associated with roughly 1.8 times the acceptance rate at the same checkpoint. This observational comparison links the reasoning pattern to the evaluation outcome but does not establish that CAT itself causes acceptance.

Table 19: Association between full-CAT and HAR at the step-200 checkpoint ($N = 1{,}688$). The comparison is within checkpoint and is reported as an association, not a causal effect.

Trace group	N	Accepted	HAR (%)
With full-CAT	1,064	286	**26.9**
Without full-CAT	624	92	14.7
Difference	—	—	**+12.1 pp**

95% CI for the HAR difference: $[8.3, 16.0]$ percentage points; $\chi^2(1) = 33.3$, $p < 10^{-8}$; odds ratio $= 2.13$ (95% CI $[1.64, 2.76]$).

Human validation of the detector. Because the preceding counts depend on automatic classification, we compare the detector against human annotation on 50 traces, sampled evenly from its positive and negative predictions. The confusion matrix in Table 20 yields 92% raw agreement and Cohen's $\kappa = 0.84$ (95% CI $[0.69, 0.99]$). At this balanced sampling point, positive and negative predictive values are 96% and 88%, respectively. The two off-diagonal counts do not indicate a directional error asymmetry (McNemar $p = 0.62$). This preliminary check supports using the detector for the frequency analysis, while the predictive values and κ should be interpreted under the balanced validation design.

Table 20: Reliability of the automatic full-CAT detector against human annotation ($N = 50$, balanced by the automatic label).

	Human: full-CAT	Human: no full-CAT
Automatic: full-CAT	24	1
Automatic: no full-CAT	3	22

Cohen's $\kappa = 0.84$ (95% CI $[0.69, 0.99]$); raw agreement $= 92\%$; PPV $= 96\%$; NPV $= 88\%$; McNemar $p = 0.62$.

G.3 Additional Analysis

Benchmarks and evaluation settings. We briefly describe each benchmark, its content, and the metric used.

- **AIME 2025** (Zhang & Team, 2025): 30 problems drawn from the official American Invitational Mathematics Examination; competition-level mathematical reasoning. We sample 32 responses per problem and report avg-pass@1 and Pass@32. Figure 14 shows a representative case where the base model produces an incorrect solution and the trained model arrives at the correct answer.
- **IFEval** (Zhou et al., 2023): An instruction-following evaluation benchmark comprising 541 prompts with verifiable constraints (e.g., format, length, keyword requirements). We report the strict prompt-level accuracy. Figure 15 shows a representative case where the base model violates a formatting constraint and the trained model satisfies it correctly.

H Human Evaluation

Participants. We recruit 10 participants (4 female, 6 male; age 22–30) with regular writing experience.

Procedure. Each participant completes two full writing sessions—one with Qwen3-8B and one with PCL-History—in counterbalanced order to control for order effects. In each session, participants are given a writing prompt and asked to compose a complete article (target length: $\sim$1000 words) using the AI assistant in a stateful co-writing interface. The assistant proactively proposes continuations at each step; participants may accept, revise, or reject each suggestion.

Measures. We record four metrics per session:

- **Acceptance rate**: proportion of suggestions accepted without modification.
- **Edit distance**: average token-level edit distance between an accepted suggestion and the final incorporated text.
- **Satisfaction** (1–5 Likert): "The assistant's suggestions matched what I intended to write next."
- **Fluency** (1–5 Likert): "The assistant's suggestions read naturally as a continuation of my writing."

Figure 17 shows representative screenshots of the co-writing interface used in the study.

I Prompts

This section collects all LLM prompts used in the paper. Figure 19 shows the prompt for the local coherence reward judge (r_{loc}), which determines whether a generated continuation directly follows the preceding context. Figure 20 shows the prompt for the semantic alignment reward judge (r_{sem}), which assesses whether the continuation conveys the same core meaning as the behavioral reference. Figure 18 shows the prompt for the independent LLM judge used in external continuation evaluation, scoring each continuation along five dimensions: local coherence, intent alignment, discourse-role fit, style consistency, and scope control. Figure 21 shows the hierarchical acceptance judge used for the primary HAR evaluation. Figure 22 shows the edit-action judge used to compute KED for accepted continuations.



Figure 10: Case study comparing prefix-only and history-conditioned context. History-conditioned context provides explicit guidance on length, content, and style for the next continuation. In this example, the interaction history reveals the writer's preference for bold emphasis ("**"-style markers), which the model correctly adopts in its subsequent suggestion—a behavior absent under prefix-only context where such stylistic cues are unavailable.

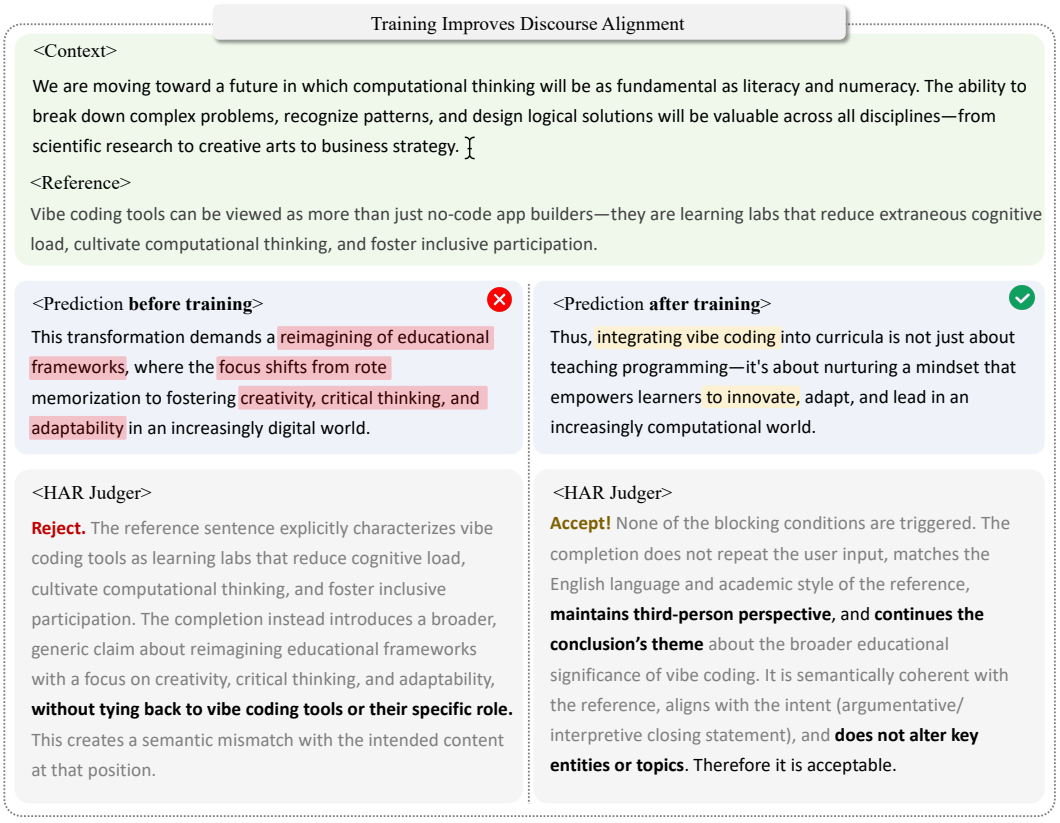


Figure 11: Case study illustrating how process-continuation training improves discourse alignment. Given a context about vibe coding tools and a reference sentence that continues the vibe-coding theme, the **pre-training** model drifts to a broader claim about reimagining educational frameworks—semantically plausible but topically misaligned, as it fails to tie back to vibe coding tools specifically. The HAR judge rejects it for introducing a generic educational-reform argument without connecting to the intended content at that position. The **post-training** model stays on topic: it continues the vibe-coding theme, maintains third-person perspective, and produces a closing statement that aligns with the argumentative direction of the reference. The HAR judge accepts it as a coherent, intent-aligned continuation.

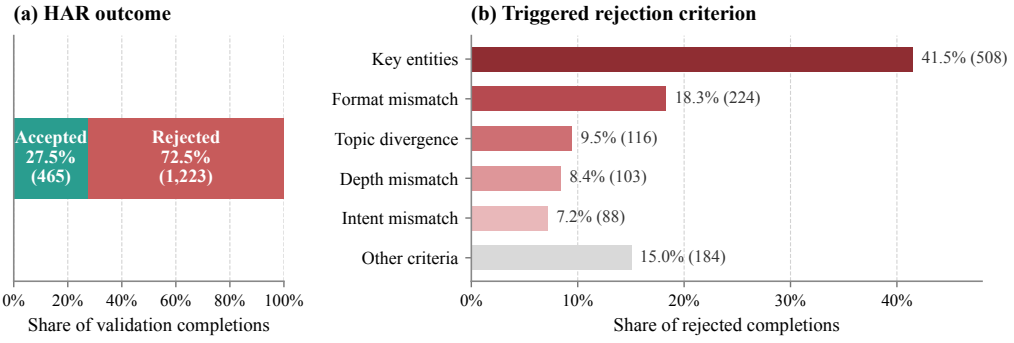


Figure 12: Validation analysis for the best Round-1 checkpoint, LOCAL VALIDITY→SEMANTIC ALIGNMENT/Mix, on the single-turn co-writing validation benchmark from Fang et al. (2026) ($N = 1{,}688$). **(a)** HAR accepts 465 completions (27.5%) and rejects 1,223 (72.5%). **(b)** Conditional distribution of the first triggered HAR criterion among rejected completions. Entity preservation is the dominant failure mode, followed by format mismatch; "Other criteria" aggregates the remaining checklist conditions. Percentages in panel (b) use rejected completions as the denominator. Results are from one stochastic evaluation round.



Figure 13: Annotated example of full context-aware thinking (cf. Appendix G.2). The model's internal `<think>` trace contains six segments spanning the four operational behaviors: *Context Summary* (S1) surveys the document topic and newly added material; *Current State Analysis* (S2) assesses what has been written, the writer's likely next intent, and the required discourse role; *Rejected Reflection* (S3) revisits why the previous suggestion was rejected and infers the writer's revised intent; and *Continuation Strategy* (S4) formulates a concrete generation plan. The aggregate analysis shows that S2 and S3, rather than all four behaviors, emerge during PCL-History training.

Math Reasoning step-by-step

<Problem>

Let $ A $ be the set of positive integer divisors of 2025. Let $ B $ be a randomly selected subset of $ A $. The probability that $ B $ is a nonempty set with the property that the least common multiple of its elements is 2025 is $ \\frac{m}{n} $, where $ m $ and $ n $ are relatively prime positive integers. Find $ m + n $.\nRemember to put your final answer within \\boxed{}.",

<Reasoning **before training**>

Okay, so I need to find the probability that a randomly selected nonempty subset B of set A (positive integer divisors of 2025) has LCM equal to 2025...First, let me factorize 2025... 2025 = 3^4 × 5^2. The number of divisors is (4+1)×(2+1) = 15.Now, B is a randomly selected subset of A. The problem says "nonempty set", so we need to consider all nonempty subsets. The total number of nonempty subsets is 2^15 - 1 ... By inclusion-exclusion: |X ∩ Y| = 2^15 - 2^12 - 2^10 + 2^8 = 27904 Therefore, the probability is 27904 / (2^15 - 1) = 27904 / 32767 ... GCD(32767, 27904) → GCD(27904, 4863) → GCD(4863, 3589) → ... → GCD(3, 1) = 1 ... Therefore, m + n = 27904 + 32767 = 60671.

<Reasoning **after training**>

Okay, so I need to solve this probability problem. Let me try to understand it step by step. First, 2025 = 3^4 × 5^2. The number of divisors is (4+1)×(2+1) = 15. Now, the total number of subsets of A is 2^15. But since we are considering nonempty subsets, the total is 2^15 - 1. **However**, the problem says the probability that B is a nonempty set ... **Wait,** but the problem says B is a randomly selected subset of A. So, the total number of subsets is 2^15. ...**But to be safe, maybe I should check both interpretations**...The probability is over all subsets, but the event is that B is nonempty and has LCM 2025 ... Therefore, the probability is [number of nonempty subsets with LCM 2025] / 2^15 ... By inclusion-exclusion: 2^15 - 2^12 - 2^10 + 2^8 = 27904... **Wait, the count includes empty set?** No. If you take the total subsets, which includes empty set, and subtract subsets that don't have a=4... the empty set has neither, so it is not included. The count already excludes it. Therefore, probability = 27904 / 2^15 = 27904 / 32768, 2^8*(2^7 - 2^4 - 2^2 + 1) = 2^8 × 109 Therefore, 109 / 2^7 = 109 / 128. **109 and 128 coprime?** 128 = 2^7, 109 is prime and odd. Therefore, m + n = 109 + 128 = 237. **Let me verify** with small example: 2^1 × 3^1 = 6. 16 - 4 - 4 + 2 = 10. Correct**! But wait, let me double-check**: ...Therefore, m + n = 109 + 128 = 237

Figure 14: Case study on AIME 2025 mathematical reasoning (finding $m + n$ where the probability that a randomly selected nonempty subset of divisors of 2025 has LCM equal to 2025 is $\frac{m}{n}$ in lowest terms). **Before training**, the base model applies inclusion-exclusion but makes an accounting error, arriving at the incorrect answer $m + n = 60671$. **After training**, the model exhibits more careful step-by-step verification: it explicitly checks whether the empty set should be counted, confirms that 109 and 128 are coprime, and validates the result with a small concrete example ($2^1 \times 3^1 = 6$) before committing to the final answer $m + n = 237$. The improved reasoning reflects structured internal planning that emerges from process-continuation training on mathematical solution traces.

Instruction Following constraint-aware

<Task>

Create a table with a 7-day trip itinerary for India, and a 7-day trip itinerary for China. Separate them with exactly 6 asterisk symbols: ******

<Response **before training**>

India 7-Day Trip Itinerary
Day	Location	Activities
1	Delhi	Explore Red Fort, Jama Masjid, and Chandni Chowk; enjoy street food.
2	Delhi	Visit Qutub Minar, India Gate, and attend a cultural dance performance.
3	Agra	Tour Taj Mahal, Agra Fort, and sunset view at Fatehpur Sikri.
4	Jaipur	Explore Amber Fort, City Palace, and Hawa Mahal; shop at markets.
5	Jaipur	Camel safari in Rajasthan Desert or visit Raj Mandir Cinema.
6	Pushkar	Visit Pushkar Lake, local temples, and enjoy a sunset camel ride.
7	Delhi	Free day for shopping, optional visit to Siri Fort or yoga session.
****
China 7-Day Trip Itinerary
Day	Location	Activities
1	Beijing	Tour Tiananmen Square, Forbidden City, and Wangfujing Street.
2	Beijing	Visit Great Wall (Mutianyu), Summer Palace, and enjoy Peking Duck dinner.
3	Xi'an	Explore Terracotta Army, Ancient City Wall, and Big Wild Goose Pagoda.
4	Xi'an	Day trip to Saros Lake or visit Shaanxi History Museum.
5	Shanghai	Tour the Bund, Yu Garden, and Huangpu River cruise.
6	Shanghai	Explore Shanghai Museum, Jin Mao Tower, and enjoy a night market.
7	Suzhou	Day trip to classical gardens and Silk Museum; return to Shanghai.

<Response **after training**>

India 7-Day Trip Itinerary
Day	Location	Activities
Day 1	Delhi	Visit Red Fort, Jama Masjid, and Chandni Chowk. Explore India Gate.
Day 2	Agra	Day trip to Taj Mahal and Agra Fort. Dinner at local restaurant.
Day 3	Jaipur	Explore Amber Fort, Hawa Mahal, and City Palace. Evening market shopping.
Day 4	Ranthambore Natl. Park	Wildlife safari to spot tigers, leopards, and birds.
Day 5	Udaipur	Visit Lake Pichola, City Palace, and Jagdish Temple. Evening boat ride.
Day 6	Jaipur	Free day for shopping or exploring Mehrangarh Fort.
Day 7	Return to Delhi	Departure for international flight or onward travel.
******
China 7-Day Trip Itinerary
Day	Location	Activities
Day 1	Beijing	Explore Forbidden City, Tiananmen Square, and the Summer Palace.
Day 2	Beijing	Visit the Great Wall (Mutianyu section) and the Temple of Heaven.
Day 3	Shanghai	Travel to Shanghai; visit the Bund, Yu Garden, and Oriental Pearl Tower.
Day 4	Hangzhou	Day trip to West Lake, Longjing Tea Plantation, and Hangzhou Silk Museum.
Day 5	Suzhou	Explore Humble Administrator's Garden, Tiger Hill, and Suzhou Museum.
Day 6	Hangzhou	Free day to relax or revisit West Lake.
Day 7	Return to Shanghai	Departure for international flight or onward travel.

Figure 15: Case study on IFEval instruction following. The task requires generating two 7-day trip itineraries (India and China) separated by *exactly* 6 asterisk symbols (******). **Before training**, the base model produces 7 asterisks (*******), violating the explicit count constraint while otherwise generating fluent content. **After training**, the model correctly outputs exactly 6 asterisks, demonstrating that process-continuation training improves precise constraint adherence even on formatting requirements not present in the writing training data.

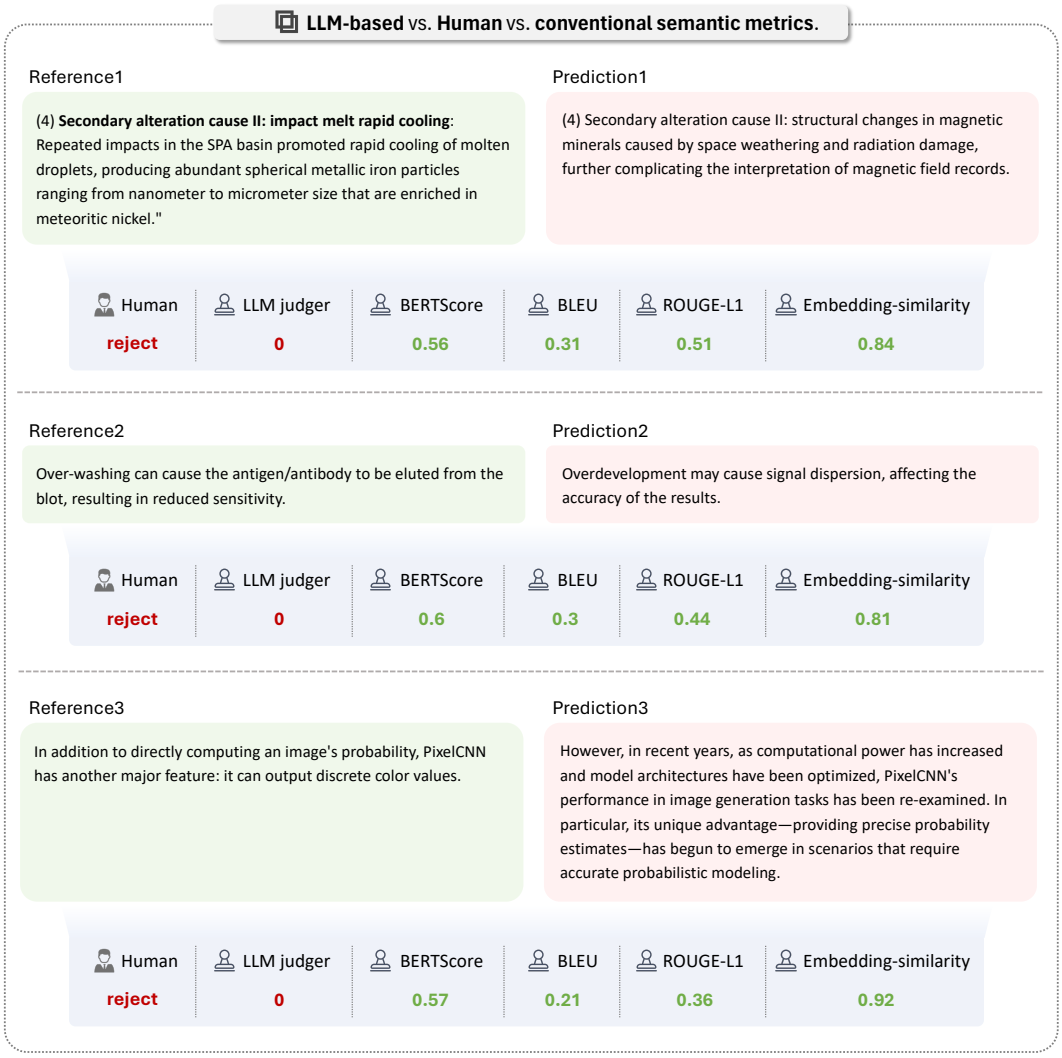


Figure 16: Case studies illustrating the divergence between LLM-based semantic judgment and conventional automatic metrics. Each example shows a model-generated continuation that both the LLM judge and human annotators reject as semantically misaligned with the reference, while string-overlap and embedding-based metrics (BERT-F1, embedding similarity, BLEU, ROUGE-L) assign relatively high scores. These cases highlight that conventional metrics can be misled by surface-level lexical overlap, whereas the LLM judge better captures whether the continuation conveys the same authorial intent as the reference.

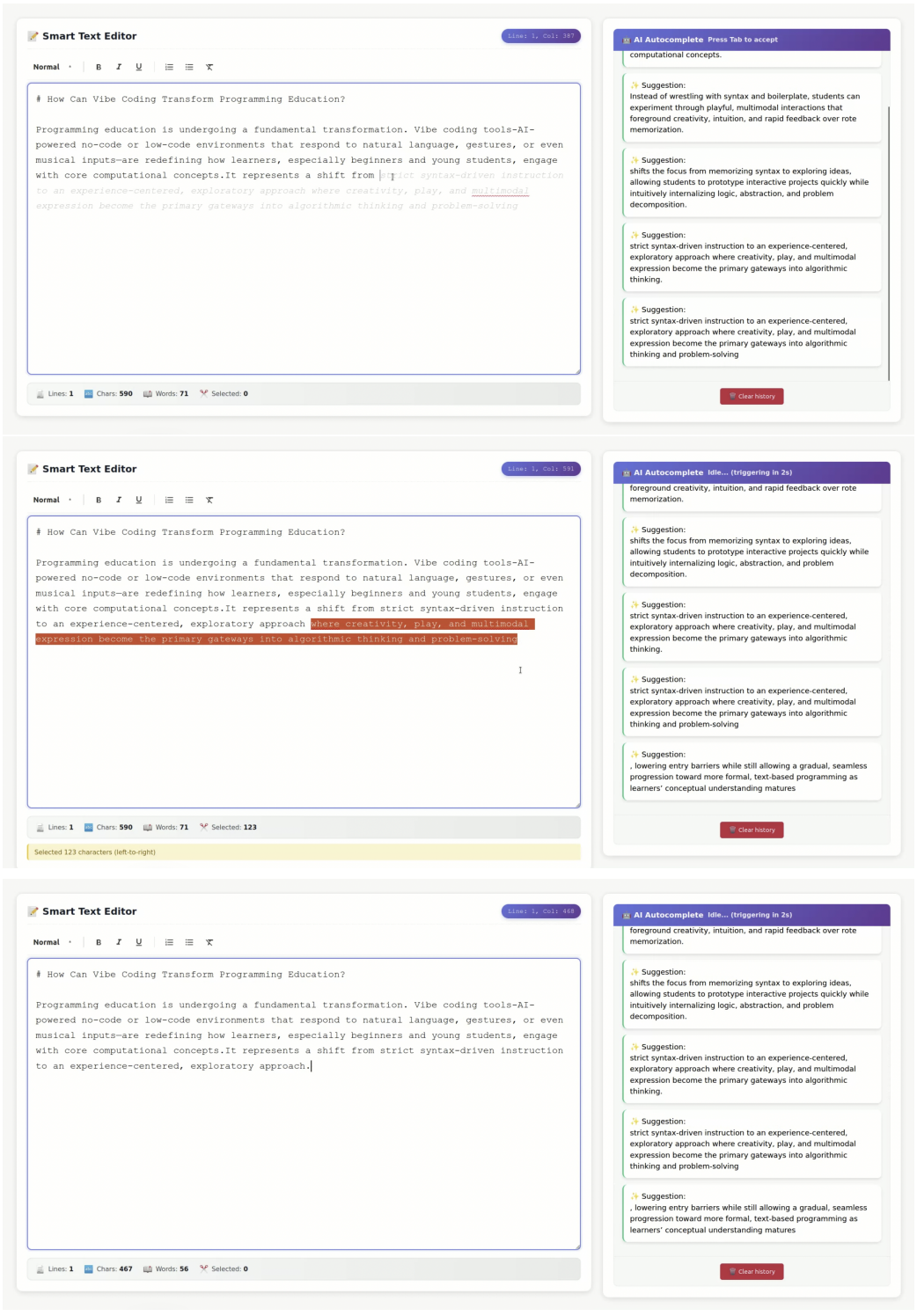


Figure 17: User study screenshot. *Top*: PCL-History predicts a continuation proposing a shift from strict syntax-driven instruction to an experience-centered, exploratory approach where creativity, play, and multimodal expression become the primary gateways into algorithmic thinking and problem-solving. *Middle*: the participant accepts the suggestion. *Bottom*: the participant removes the subsequent redundant content.

Prompt: External Continuation Evaluation Judge

You are evaluating process-continuation quality.
Given:
1. A writing prefix from a naturally occurring document.
2. The original next sentence/snippet as a behavioral reference.
3. A model-generated continuation.
Your task is to judge whether the model continuation is a good next step after the prefix. The reference is not a gold string to be copied; it indicates the likely authorial intent and discourse direction at this point. A good continuation may use different wording from the reference if it serves a similar next-step function.
Rate the model continuation on five dimensions from 1 to 5:
1. Local Coherence: Does the continuation naturally follow the prefix in grammar, topic, and immediate discourse flow?
2. Intent Alignment: Does the continuation match the likely next-step intent reflected by the reference, even if the wording differs?
3. Discourse Role Fit: Does the continuation play the appropriate rhetorical/discourse role at this point, such as defining, motivating, contrasting, explaining, or transitioning?
4. Style Consistency: Does the continuation match the style, tone, and register of the source domain?
5. Scope Control: Is the continuation appropriately scoped as the next sentence/snippet, without being overly broad, verbose, premature, or underspecified?
Return a JSON object with integer scores from 1 to 5 and a brief rationale.

Figure 18: Prompt used for the LLM judge in external continuation evaluation (Section G.1), scoring each generated continuation along five dimensions: local coherence, intent alignment, discourse-role fit, style consistency, and scope control.

Prompt: Local Coherence Judge (r_{loc})

System Prompt

You are a professional researcher collaborating with your Writing Assistant to complete a document. The Writing Assistant will fill in subsequent content based on the user's typed input <USER_INPUT>. Your responsibility is: given the user's typed input <USER_INPUT> and the Assistant-generated completion <COMPLETION>, determine whether this completion is a coherent, contextually appropriate, and logically consistent extension of a given prefix.

Decision Criteria (Checklist)

1. <COMPLETION> must logically follow from the <USER_INPUT>.
2. <COMPLETION> must maintain the same topic, style, and tone.
3. <COMPLETION> must smoothly continue the narrative or argument flow.
4. <COMPLETION> should feel like a natural next part of the <USER_INPUT>.
5. <COMPLETION> must **directly continue from the <USER_INPUT>** — the first character of the completion must immediately follow the last character of the <USER_INPUT>.
6. When <USER_INPUT> and <COMPLETION> are seamlessly concatenated, they should form a naturally expressed and logically coherent passage.

Evaluation Input

```
<USER_INPUT>: "{context}"
<COMPLETION>: "{predicted}"
```

Scoring Rules

If the <COMPLETION> satisfies *all* of the above checklist items, assign a score of **1**. If the <COMPLETION> fails to meet *any* of the checklist items, assign a score of **0**.

Output Format

```
Score: 0
or
Score: 1
```

Figure 19: Prompt used for the local coherence reward judge (r_{loc}) during training.

Prompt: Semantic Alignment Judge (r_{sem})

System Prompt
You are a careful semantic evaluation expert. Your task is to assess whether a predicted sentence is semantically equivalent to a reference paragraph.
Task
Given a **Predicted** sentence and a **Reference** paragraph, determine whether the Predicted text achieves at least **80%** semantic similarity with the Reference.
The inputs are provided in the following format:

```
Reference:
{reference}

Predicted:
{predicted}
```

Scoring Rules

- If the Predicted text expresses the same core meaning or main idea as the Reference, with more than 80% semantic similarity, assign a score of `1`.
- If the Predicted text shares less than or equal to 80% semantic similarity with the Reference, or conveys a clearly different meaning, assign a score of `0`.
- The judgment should focus on *semantic overlap* rather than exact lexical or surface-form matching.

Output Constraints

- Output only the score on a single line.
- Do *not* include any explanations, reasoning steps, or additional text.

Output Format

```
Score: 0
```

or

```
Score: 1
```

Figure 20: Prompt used for the semantic alignment reward judge (r_{sem}) during training.

Prompt: Hierarchical Acceptance Judge (HAR)

Task Description
You are a professional researcher collaborating with your Writing Assistant to complete a document. The Writing Assistant will fill in subsequent content based on the user's typed input <USER_INPUT>. Your responsibility is: given the user's typed input <USER_INPUT>, the Assistant-generated completion <COMPLETION>, and the reference text (original content at the completion position) <REFERENCE>, determine whether this completion should be "accepted" or "rejected".

Decision Criteria (Checklist)
Please execute strictly in order, first checking blocking conditions. When triggered, make a direct judgment without proceeding to subsequent comparisons.

1. **Start repetition:** if <COMPLETION> starts with a **repetition** of <USER_INPUT> → reject.
2. **Language mismatch:** if <COMPLETION> and <REFERENCE> are in different languages (e.g., English vs. Simplified Chinese vs. Traditional Chinese) → reject.
3. **Semantic coherence:** if <USER_INPUT> + <COMPLETION> form a contradictory or incoherent sequence → reject.
4. **Early semantic overlap with <REFERENCE>:** if the overlapping content between the beginning of <COMPLETION> and the beginning of <REFERENCE> accounts for more than 50% of the total <COMPLETION> content → accept.
5. **Paired punctuation mark closure:** unclosed quotes/brackets introduced in <USER_INPUT> remain unclosed in <COMPLETION> → reject.
6. **Markdown/LaTeX/Code closure:** any opened Markdown/LaTeX/code fences from <USER_INPUT> not properly closed in <COMPLETION> → reject.
7. **Format mismatch:** if format mismatch between <COMPLETION> and <REFERENCE> (headings, tables, lists, text) → reject.
8. **Format consistency with preceding text:** the format/style used in <COMPLETION> diverges from the format established by the preceding content of <USER_INPUT> and <REFERENCE>, e.g., changes in list type, indentation levels, heading levels → reject.
9. **Depth mismatch:** specificity/level of detail diverges from <REFERENCE> beyond tolerance ($\pm 30\%$) → reject.
10. **Style/Register mismatch:** academic vs. conversational vs. authoritative diverges from <REFERENCE> → reject.
11. **Sentence type mismatch:** declarative vs. interrogative vs. imperative diverges from <REFERENCE> → reject.
12. **Personal perspective check:** If <COMPLETION> shifts the narrative perspective (first person, second person, third person) established in <USER_INPUT> → reject.
13. **Subset acceptance (lists/tables):** if <COMPLETION> is a true subset of <REFERENCE> (same order, no new items; tables must keep columns and header order) → accept.
14. **Topic divergence:** if <COMPLETION> and <REFERENCE> address different topics → reject.
15. **Key entities:** missing or altered key entities (names, trial IDs, datasets, metrics, units, dates, statistics data, terminology) compared to <REFERENCE> → reject.
16. **Intent mismatch:** if the intent of <COMPLETION> differs from <REFERENCE> (e.g., summary vs. argument vs. instruction) → reject.
17. **Comprehensive judgment:** if none of the above conditions are triggered, please carefully analyze the <USER_INPUT> and <REFERENCE>, then make a comprehensive judgment on whether to accept the <COMPLETION> based on style, semantics, entities, and other factors.

Output Format
Please strictly use the following JSON format for output and enclose the entire JSON in `\boxed{ ... }`,

```
\boxed{
  "accept": true | false,
  "triggered_condition": "rule number and name (e.g., \"1. Start
     repetition\")",
  "reasoning": "Provide your reasoning, explicitly mentioning which
     rule number was triggered and why."
}
```

Task Input

```
<USER_INPUT>: "{context}"
<REFERENCE>: "{sentence_A}"
<COMPLETION>: "{sentence_B}"
```

Figure 21: Prompt used for the primary hierarchical acceptance rate (HAR) evaluation judge.

Prompt: Knowledge-Weighted Edit Distance (KED)

System Prompt
You are a professional researcher collaborating with your Writing Assistant to complete a document. The Writing Assistant will generate subsequent text <COMPLETION> based on the user's input <USER_INPUT>. Your responsibility is to evaluate the editing cost required to transform the Assistant-generated <COMPLETION> into the <REFERENCE> text. To achieve this, you must first holistically assess the differences between the two texts and then generate a clear **List of Editing Actions** given predefined actions. The final **Total Editing Cost** is the sum of the scores for all actions.

Actions

1. "ADD"
2. "DELETE"
3. "MODIFY"

Cost Assessment Rules
The cost of each action correlates with the **entities** involved and the **transitional phrasing**. Generally, complex entities incur higher costs than simple entities, and substantial transitional phrasing revisions are more costly than minor ones.

Cost Assessment of Entities

- A **complex entity** in action "ADD" or "MODIFY" is scored as **3 points**.
- A **simple entity** in action "ADD" or "MODIFY" is scored as **1 point**.
- Action "DELETE" is assessed as part of the transitional phrasing evaluation.

Cost Assessment of Transitional Phrasing

- Score **0** point if the relational description is identical or semantically equivalent.
- Score **1** point for minor, simple additions, deletions, or modifications of words.
- Score **2** points for substantial modifications to the core verbs, phrases, or overall structure of the relational description.

Complexity Assessment of Entities
Complex Entity (COMPLEX): Entities that require fact-checking, external knowledge, or precise recall. These typically include proper nouns (e.g., names of people, places, organizations), scientific, technical, or legal terminology, specific product or substance names, and precise numbers, models, or dates.
Simple Entity (SIMPLE): Common nouns, descriptive words, or common-sense concepts that can be intuitively and quickly verified or modified.

Assessment Procedure
Generate a list of editing actions. The list of editing instructions for the <COMPLETION> may consist of various combinations of actions. Intuitively, you prefer action sequences with lower cost. The cost is computed as follows:

1. **Single Action Cost Accumulation**: The cost for each action is the cumulative sum of the scores of all involved entities and the transitional phrasing complexity score.
2. **Total Action Cost Accumulation**: The final **[Total Editing Cost]** is calculated by summing the costs of all actions.

Input and Output Example
Input

```
<REFERENCE>: "The protein is eluted from the polyacrylamide gel and
    immobilized on the membrane surface."
<COMPLETION>: "The protein is transferred from the gel to the membrane."
```

Output
Please strictly use the following JSON format for the output and enclose the entire JSON object within a \boxed{. . . } block:

```
\boxed{
  {
    "edit_plan": [
      {
        "operation": "MODIFY",
        "instruction": "Revise the entity 'protein' to 'The protein'.",
        "cost": 1,
        "reasoning": "Modification of a single simple entity."
      },
      {
        "operation": "MODIFY",
        "instruction": "Revise the entity 'gel' to 'polyacrylamide gel'.",
        "cost": 3,
        "reasoning": "Involves the modification of one complex entity."
      },
      {
        "operation": "MODIFY",
        "instruction": "Revise the phrasing 'is transferred... to' to 'is
            eluted from... and immobilized on the... surface'.",
        "cost": 2,
        "reasoning": "Substantial modification to the core verb and overall
            structure."
      }
    ],
    "total_editing_cost": 6,
    "summary": "The total editing cost is 6 points, comprising the
        modification of one complex entity and one simple entity, along with
         a substantial revision to the transitional phrasing."
  }
}
```

Task Input

```
<REFERENCE>: "{sentence_A}"
<COMPLETION>: "{sentence_B}"
```

Figure 22: Prompt used to compute knowledge-weighted edit distance (KED) for HAR-accepted continuations.