CUSTOMNUDGE: LEARNING PERSONALIZED INITIATIVE POLICIES FOR LLM ASSISTANTS

Darynne Lee & Eyrin Kim
Department of Computer Science
Stanford University
Stanford, CA 94305, USA
{darynnel, eyrinkim}@stanford.edu

ABSTRACT

Existing AI assistants suffer from the grounding gap: they react passively to explicit queries but fail to proactively intervene based on evolving user context. This limits true human–AI collaboration, where an assistant exercises initiative to reduce cognitive load and help users maintain goal alignment. We introduce CustomNudge, a framework for learning personalized initiative policies from real-time multimodal observations. By leveraging Generative User Models (GUM) to transform desktop activity into semantic behavioral signals, CustomNudge identifies meaningful drift and employs a decision engine that refines its intervention policy through in-context learning informed by implicit behavioral feedback. In a study with 5 participants across 15 sessions, the system demonstrated highly selective initiative behavior (nudging only 14% of decision points) and successfully targeted moments of low goal alignment. The system showed signs of adaptation, with participants increasingly accepting nudges across sessions. Notably, an acceptance–compliance gap emerged: participants increasingly endorsed nudges as appropriate but did not immediately shift behavior, suggesting that enhanced delivery modalities could help translate agreement into action. Our findings show that LLM assistants can learn initiative policies from evolving real-world user feedback, marking a step toward mixed-initiative assistants that collaborate seamlessly with users and realize the promise of true human–AI partnership.

1 INTRODUCTION

Current large language model (LLM) assistants primarily operate in a reactive paradigm. They wait for explicit user prompts, process the request, and return a response. This interaction model fails to address the grounding gap. AI assistants lack access to, and understanding of, a user's real-time context and underlying goals, leaving them unable to initiate assistance appropriately. This gap limits fluid human-AI collaboration and places the cognitive load of initiating assistance on the user. To move toward mixed-initiative assistants (1), AI systems must learn to recognize when to intervene proactively and, crucially, when to remain silent in an adaptive, context-aware manner.

This paper presents CustomNudge, a system that learns personalized initiative policies for proactive assistance. We ground this inquiry in the domain of behavioral nudging, where the assistant monitors user activity and autonomously delivers nudges to foster focus, productivity, or well-being. The core challenge is not merely generating the content of a nudge, but determining the optimal policy for intervention: identifying the specific contextual and temporal patterns that make a nudge effective.

However, learning when to intervene is fundamentally challenging. User workflows evolve, task boundaries blur, and explicit labels about "good" or "bad" timing do not exist. This raises two central research questions: (1) How can AI assistants learn when to autonomously initiate interactions using implicit behavioral feedback? (2) Can personalized nudging serve as a testbed to reveal which contextual and timing patterns drive actual behavior change? Traditional alignment methods like RLHF (2) incorporate human feedback but focus on learning content preferences and rely on static labels that do not reflect evolving user preferences. CustomNudge instead utilizes an adaptive feedback loop: it observes the user's on-screen activity before and after a nudge and uses an LLM

Judge to interpret behavioral changes. This transforms moment-to-moment behavior into implicit feedback signals for learning initiative policies dynamically.

We demonstrate that CustomNudge successfully learns a selective intervention policy, intervening only when goal alignment is low. Through a multi-session user study, we show that in-context learning from past effective nudges improves user acceptance rates over time, validating the feasibility of assistants that learn initiative policies dynamically.

2 Related Work

Human-Centered Alignment. Standard alignment techniques such as Reinforcement Learning from Human Feedback (RLHF) (2) and Direct Preference Optimization (DPO) (3) align models with human preferences. However, these methods typically treat preferences as static and population-based. They fail to capture the dynamic, highly contextual nature of individual workflows. CustomNudge differentiates itself by learning dynamic initiative policies driven by real-time, implicit behavioral signals rather than static preference datasets. More importantly, these alignment methods optimize what models say, not when they act. They do not support initiative policy learning or context-driven timing decisions.

Contextual User Modeling. Recent advancements in user modeling, including Generative User Models (GUM) (5), demonstrate the ability to infer evolving context by transforming raw observations into semantic propositions without requiring labeled data. While GUM provides the necessary memory and reasoning layer for understanding context, it lacks a mechanism for intervention policy learning. It is a passive observer. CustomNudge builds on GUM, using its semantic propositions to ground an action-selection engine that learns from intervention outcomes.

Mixed-Initiative Interaction. Horvitz's principles of mixed-initiative user interfaces (1) established the importance of balancing automated services with direct manipulation. Recent work in Just-In-Time Objectives (6) infers momentary goals to specialize AI interactions. Our work extends this by closing the loop: we not only infer the goal but actively intervene to correct deviations, using the success of those interventions to refine the underlying policy.

Proactive and Context-Aware Systems. Earlier work on context-aware computing (7) established the importance of using context to provide relevant information and services. More recent systems, such as Microsoft's Cortana and Google Assistant, attempt proactive behaviors like suggesting calendar events and traffic alerts. However, while these systems now incorporate models for understanding, their execution layer still relies primarily on rule-based triggers, limiting their ability to generalize across dynamic workflows. In the productivity domain, tools like RescueTime and Freedom block distracting websites on predetermined schedules, but lack semantic understanding of user intent. For example, a deterministic blocker like 1Focus may restrict access to Reddit entirely, even when a user is researching a technical problem on a relevant subreddit. CustomNudge's semantic flexibility allows for goal-aligned interventions that deterministic systems cannot achieve.

Behavioral Interventions and Nudging. The concept of "nudging" originates from behavioral economics (8), referring to subtle interventions that guide behavior without restricting choice. In digital settings, nudging has been applied in domains such as promoting physical activity (9), reducing screen time (10), and encouraging mindful smartphone use (11). However, digital nudging typically rely on predetermined schedules or manual trigger conditions (e.g., time-based limits). CustomNudge represents a shift toward learned nudging policies that adapt based on observed effectiveness for each individual user.

3 Data

We conducted a user study with 5 participants (Stanford students ages 19-22) focused on productivity tasks. The study consisted of 3 sessions per user (15 total sessions). Participants engaged in authentic workflows on their personal devices. Unlike controlled lab tasks, these sessions involved real-world academic objectives defined by the users themselves. Examples of user goals included "Debugging Python script for CS224N," "Writing a history research paper," and "Organizing Zotero citations." Sessions lasted between 60 and 90 minutes. This setup ensured that goal alignment scores reflected genuine productivity states rather than artificial task compliance.

To standardize evaluation across participants, each session began with an explicit user-provided goal statement (e.g., 'Finish problem set'). Although CustomNudge can infer goals from context, providing explicit goals ensures isolation of the effects of intervention timing and initiative decisions when computing goal alignment metrics.

We utilize the GUM framework (5) to capture desktop screenshots, which are transcribed by a Vision-Language Model (Gemini 2.5 Flash) into textual descriptions. These are converted into "Propositions," confidence-weighted statements about user habits and states. The dataset includes detailed logs for each of 312 collected decision points. For each point, the system logs the inputs (current observations, goal alignment score), the decision output (Relevance, Urgency, Impact scores), and behavioral aftermath of that decision.

We established baselines grounded in self-reported behaviors and captured passive data without interventions. Then, we analyzed the three sessions differently: Session 1 acts as a low-context baseline, while Sessions 2 and 3 operate with accumulated contextual memory and serve to evaluate adaptive learning over time.

We also collect behavioral feedback quantitatively and qualitatively. The system captures both the continuous stream of GUM-generated observations and the "Post-Nudge States," a sequence of desktop snapshots and active-application traces over the 120-second window following a nudge. These implicit signals allow us to compute goal adherence and behavioral shifts relative to the pre-nudge context. Further, users explicilty provide "Accept/Reject" feedback on nudges and complete qualitative pre- and post-session surveys for perceived intrusiveness, timing accuracy, and helpfulness. The "Accept/Reject" feedback is not used for policy learning.

4 METHODS

CustomNudge operates as a closed-loop system comprising three distinct stages: an observation engine (GUM), a decision engine for intervention selection, and an adaptive learning loop.

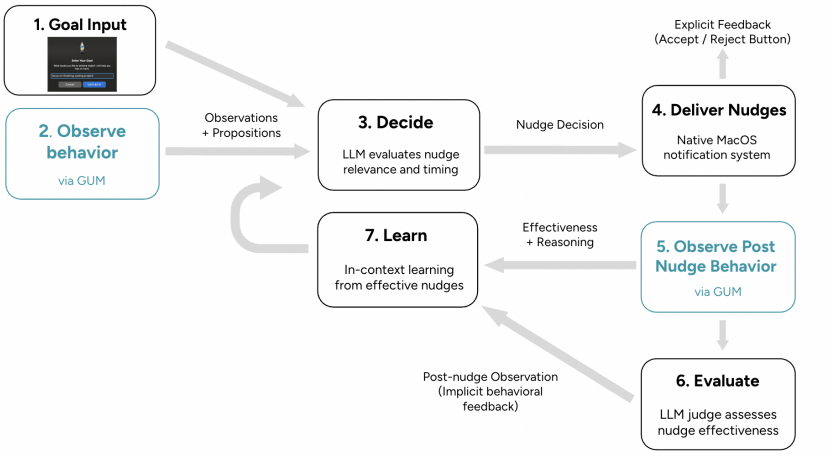

Figure 1: CustomNudge Architecture. Multimodal observations are processed into propositions via GUM and fed into the Decision Engine, triggering an intervention. Post-nudge behavior is used to update the context for future decisions.

4.1 FOUNDATION (GUM INTEGRATION)

We build our system on top of the GUM framework (5) to handle the heavy lifting of context extraction. GUM processes raw desktop screenshots and interaction logs (mouse / keyboard events) into semantic units called propositions.

- Ingestion: Desktop screenshots are captured at 10 frames per second (debounced to avoid redundant captures during static periods). Each screenshot is transcribed by a Vision Lan-

guage Model (Gemini 2.5 Flash), producing a textual description of the context and user interactions.

- Proposition Generation: Instead of reasoning over raw images or free-form descriptions, the system ingests GUM's propositions: structured tuples $\{P, c, t\}$, where P is a semantic statement (e.g., "User is scrolling Instagram"), c is a model-assigned confidence score, and t is a timestamp.
- Memory Retrieval: To provide temporal and historical grounding, the system indexes all propositions using BM25 Similarity Search. This allows the agent to retrieve historical context (e.g., "How long has the user been on Twitter today?" or "Does the user usually work on code at this hour?"), allowing the decision model to situate current behavior within longer-term patterns.

Batch Processing. To balance latency and efficiency, observations are stored in a persistent SQLite queue and processed once the minimum batch size is reached. Rather than reacting to individual frames, the Decision Engine evaluates the aggregate behavioral pattern, reducing noise from brief, inconsequential actions. This batching strategy stabilizes initiative decisions and reduces API usage by 60-90% (depending on batch size) compared to per-observation processing.

Together, these components transform unstructured multimodal on-screen activity into coherent, queryable representation of user behavior.

4.2 Decision Engine

The Decision Engine represents the core contribution of CustomNudge. It transforms the context-rich understanding provided by GUM into an actionable intervention policy, determining when the system should step in and when it should remain silent.

Model Configuration. The system uses Gemini 2.5 Flash across all components (observation transcription, decision-making, and the LLM Judge) to ensure consistent reasoning patterns and avoid variability across models. The model operates with default temperature settings (temperature=1.0) to preserve coherent but deterministic reasoning across sessions.

The engine processes by batch: once the observation queue reaches the minimum batch size (3 observations), the batch is processed and evaluated. In practice, this results in a decision point roughly every 30-60 seconds of active desktop use.

Scoring Heuristics. At each decision point, the model evaluates whether an intervention is warranted by scoring the user's current behavior along four dimensions:

1. General relevance (0-10): How relevant is this observation to behaviors the user likely wants to change?
2. Goal relevance (0-10): How related is this observation to the user's explicit goal?
3. Urgency (0-10): Does the divergent behavior risk long-term derailment?
4. Impact (0-10): Would a nudge meaningfully shift the user's trajectory?

In addition to these numeric scores, the model qualitatively evaluates timing (e.g., whether the user is at a natural stopping point or in a flow state) and novelty (whether the system has nudged recently or repeatedly). These qualitative signals influence the model's reasoning.

Goal Alignment Score. When the user provides a goal, the system computes a normalized goal alignment for offline evaluation by dividing the `goal_relevance_score` (0-10) by 10. This yields a value between 0 and 1 representing how well the current behavior aligns with the user's stated intent.

4.3 Adaptive Learning Loop

The Adaptive Learning Loop enables CustomNudge to improve its initiative policy over time by using implicit behavioral feedback. Instead of learning through gradient descent updates, the system learns through in-context learning (ICL) using retrieved examples of past successful interventions.

This mechanism effectively allows the policy to refine its initiative policy without explicit fine-tuning.

Delivery & Wait. The Decision Engine outputs a binary decision (`should_notify`) based on evaluating all dimensions. If affirmative, a native macOS notification is triggered. The system then enters a cooldown period of 120 seconds to allow for user behavior change and avoid nudge fatigue. This matches the observation window duration.

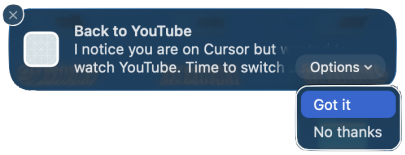

Figure 2: Nudge example.

LLM Judge. During the 120-second observation window, the system captures system state snapshots at regular 30-second intervals (5 snapshots total), along with logs derived from GUM. A separate LLM instance (The Judge) compares the pre-nudge state S_t with this multimodal sequence of post-nudge evidence. The Judge outputs an effectiveness score $y \in \{0, 0.5, 1.0\}$ and a reasoning trace R.

$$\text{Judge}(S_t, S_{t...t+120}, \text{Nudge}) \rightarrow \{y, R\}$$

Scores are assigned based on a two-stage process: evaluating snapshot compliance percentages and refining with activity logs. A score of 0.5 represents partial compliance or ambiguous cases.

Policy Update (RAG). We maintain a database of "Successful Interventions" where $y \geq 0.7$. This threshold ensures only nudges with clear positive impact are used as examples. When the Decision Engine constructs the prompt for a new decision at time t_{new}, it performs a retrieval step using bucket-based matching:

$$\text{Prompt} = \text{Base Instructions} + \text{Retrieve}(\text{Matched Successful Past Nudges}) + S_{t_{new}}$$

The system matches past effective nudges based on "Time Since Last Nudge" buckets and "Goal Relevance" buckets. This bucketed retrieval helps the policy generalize across contexts without overfitting to surface-level screenshot similarity. This effectively teaches the model: "Here are up to 5 times you nudged in similar contexts that actually worked. Use this logic to decide if you should nudge now."

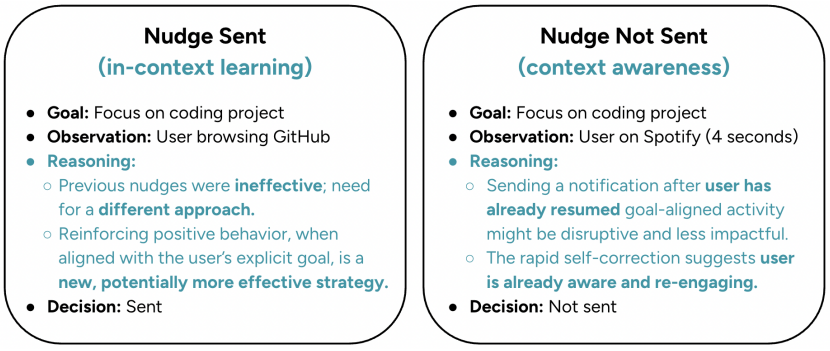

Figure 3: Decision logic examples. (Left) In-context learning enables new strategies. (Right) Self-correction detection prevents unnecessary nudges. See Appendix A for supporting images.

Through this loop, CustomNudge continuously tailors its intervention strategy to each user, learning when to act with increasing precision across sessions.

5 RESULTS

The evaluation focused on whether CustomNudge could (1) learn when to proactively intervene using implicit behavioral signals, and (2) whether personalized nudges could surface contextual and temporal patterns that meaningfully shape user behavior. Across five users and 15 sessions, we analyzed 312 decision points.

5.1 INITIATIVE POLICY LEARNING: SELECTIVE, CONTEXTUAL, AND DRIFT-AWARE BEHAVIOR

A primary concern for mixed-initiative systems is intrusiveness. CustomNudge demonstrated high selectivity: although it evaluated hundreds of opportunities to intervene, the assistant chose to nudge only 14% of the time. This indicates that the system internalized a practical "speak only when needed" principle.

Intervention Based on Goal Alignment. Table 1 compares the alignment scores during passive (non-notifying) windows and nudge-triggering windows. Alignment was normalized to a 0-1 scale. The nudges were highly correlated with actual need. The system remained silent when alignment was high (0.87) and only intervened when alignment plummeted to 0.24.

Table 1: Comparison of Goal Alignment Scores during passive windows vs. intervention windows.

State	Avg. Goal Alignment (0-1)
Passive Observation	0.87
Moment of Nudge	0.24

Participants also recognized this. One participant noted how the system helped with prioritization and reduced context switching as a primary point of distraction. Participants also praised the system's semantic flexibility compared to deterministic blockers, reporting that CustomNudge correctly distinguished productive context switching from distraction. One user remarked: "When I was on GPT or Reddit asking about LeetCode it didn't stop me despite I gave clear instructions to keep me only on LeetCode. Any other deterministic program didn't have this level of control."

Contextual Scoring Sharpens Intervention. Moments where CustomNudge intervened consistently showed elevated Relevance, Urgency, and Impact scores (average around 7-9), while non-nudge moments remained below 1 on all dimensions. This multi-factor pattern indicates that the system learned a nuanced heuristic: intervene only when the deviation is substantive and time-sensitive.

Temporal Spacing as Emergent Restraint. Across all users, nudges appeared only after 5 minute or longer gaps, even though the cooldown period is 120 seconds. This emergent "cooldown" revealed that the model naturally avoided over-intervention, aligning with human expectations for minimal disruption.

Adaptive Learning. Across sessions, participants increasingly endorsed the system's judgments. They were instructed to "Accept" the nudges if it felt timely and appropriate, and qualitative feedback suggests that this happened with increasing frequency. Participants reported that CustomNudge "knew exactly when I was getting off track" and "reminded me at the right times," reflecting growing trust in the assistant's drift detection.

5.2 BEHAVIORAL RESPONSE PATTERNS: NUDGE EFFECTIVENESS, TIMING, AND THE ACCEPTANCE-COMPLIANCE GAP

We assessed behavioral response using the LLM Judge's 120-second post-nudge evaluation. Across the nudges, partial compliance ranged from 20-80%. Although overall compliance was low, the structure of outcomes revealed meaningful patterns.

Effectiveness Depends on Timing Within the Drift Trajectory. Nudges delivered early in a drift episode (e.g., One participant shifting from data to manga) achieved higher compliance, while

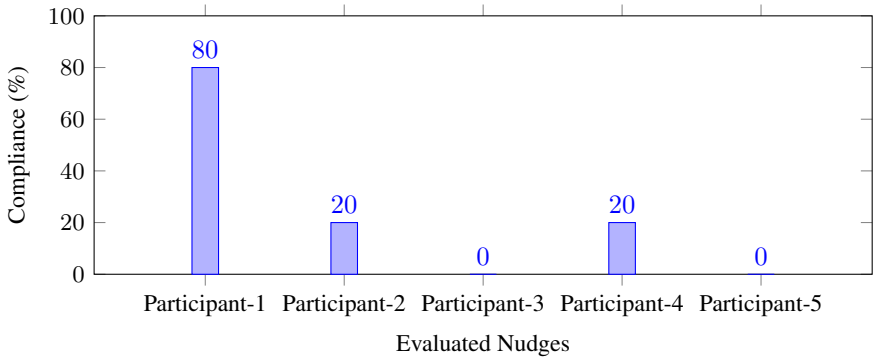

Figure 4: Compliance percentages for nudges.

nudges delivered late after the user had already stabilized into a new habit had negligible effect. This suggests that behavior change may be affected more by the temporal placement of nudges within the user's attention cycle than the frequency or phrasing of nudges.

User-Specific Responsiveness Profiles. Compliance varied by participant, as shown in Figure 3. This heterogeneity signals that users have distinct initiative responsiveness profiles that future systems could learn over time.

The Acceptance-Compliance Gap. While nudge acceptance rates (intention) were high, short-term behavioral compliance (action) was more variable. Across all nudges, the LLM Judge assigned effectiveness scores that reflected the following pattern: 20% received score 1.0 (fully effective), 35% received score 0.5 (partially effective), and 45% received score 0.0 (ineffective).

Qualitative feedback identifies the delivery modality, not the timing, as the primary bottleneck. It suggests that CustomNudge is identifying the right moments, and that future work lies in strengthening nudge delivery modality. As participants explained:

- "Using AI to detect when I'm off track was useful... but I didn't think the nudge system was strong enough."
- "It was helpful, but too subtle—I kept working in the wrong tab."
- "I wish I could adjust the nudge system to pop up in the middle of the screen or automatically minimize the distracting tab for me."

The discrepancy between high acceptance and low immediate compliance reveals a compliance gap: the system is learning when to intervene effectively, but the limiting factor is the delivery modality. The nudges were informative but not interruptive enough to consistently redirect behavior.

Agent-led Actions for Reducing the Compliance Gap. During one participant's session, an unexpected system behavior resulted in an app (Visual Studio Code) repeatedly opening whenever the user attempted to switch away from their stated goal ("Focus on coding for 30 minutes"). Though this behavior was caused by an unrelated local configuration issue rather than CustomNudge itself, it produced a noteworthy pattern in user behavior.

Initially, the recurring VS Code window was described as "annoying" and "unexpected." However, the unintended pop-ups increased friction against distraction, and the participant ultimately remained more focused on their goal in this session than in others. This illustrates how agents taking action on behalf of the user may be promising in supporting goal adherence when done thoughtfully and with respect for participant autonomy.

6 Discussion and Limitations

6.1 Discussion

The system's selective nudging behavior demonstrates that it can form a stable sense of when to intervene even under noisy, real-world task conditions. Participants' growing acceptance of the

nudges provides further evidence that users recognized these initiative decisions as appropriate and meaningful. Together, these signals point to an emerging policy that is goal-aware, drift-sensitive, and generally aligned with human expectations.

Although the LLM Judge identified limited short-horizon behavior change, users consistently endorsed the timing of nudges. This distinction suggests that compliance is also affected by cognitive load, task momentum, and interface friction all influence near-term compliance. A surprising episode, in which an unintended app popup repeatedly redirected a participant, illustrates how direct agent-led actions could improve goal adherence. Future systems could explore strategies where participants opt into stronger forms of support when drift risk is high.

6.2 Limitations

Methodological Trade-offs. The batch processing architecture introduces a latency–completeness trade-off. By waiting for several observations, the system reduces false positives but can introduce a 30–60 second delay, occasionally resulting in "stale" nudges. One participant noted that the system flet "too slow—it let [them] veer off path for a while before notifying." This batching choice reflects a balance between temporal precision and contextual robustness. Additionally, the 0.7 effectiveness threshold for learning represents a balance between example quality and quantity; a higher threshold might yield better examples but give the model less data in early sessions.

Absence of Ablation Studies. As a proof-of-concept validation extending prior GUM work (5), our study prioritized demonstrating feasibility over systematic component analysis. We did not conduct formal ablation studies isolating the contributions of batch processing, in-context learning, or the multi-dimensional scoring system. Future work could evaluate each component's individual contribution—for example, by comparing performance with and without ICL retrieval or exploring alternative batch sizes to understand how architectural choices influence initiative quality.

Limitations and Threats to Validity. This study faces several generalizability constraints. First, with 5 participants (aged 19–22), we lack statistical power to make population-level claims. All participants engaged in academically structured tasks, introducing potential biases related to technical literacy, similar goal structures, and shared cultural context around productivity. Results may not generalize to professionals in creative domains, older adults, or users with less clearly defined goals.

With only three sessions per participant (approximately 3–4.5 hours of total use), we captured the system's early-stage dynamics but could not assess long-term effects. Multi-task evaluations across diverse activity types (creative work, leisure, execution-heavy workflows) would further clarify whether the learned policy generalizes or requires domain-specific tuning.

Finally, participants were aware they were being observed and evaluated, which may have influenced their behavior in ways that do not occur in naturalistic use. This may partially explain the consistently high acceptance rates and could diminish over longer-term deployment.

7 Conclusion

This paper introduces CustomNudge, a proof-of-concept for intervention policies that adapt dynamically to users without requiring explicit supervision or static preference datasets. Through a multi-session user study, the system reliably identified meaningful moments of drift, intervened selectively, and earned increasing user trust over time. These results demonstrate the potential of using contextual and behavioral feedback, moving beyond reactive paradigms towards proactive, context-aware human-AI collaboration. Although short-term compliance varied, the system reliably detected the right moments to assist, and nudges delivered early in a drift trajectory demonstrated clear positive impact. The discrepancy between acceptance and immediate behavior change suggests not a failure of detection, but an opportunity: users are receptive and aligned with the assistant, and the next step is to strengthen delivery modalities to help translate intention into action.

Overall, CustomNudge contributes to the growing body of human-centered AI research aimed at augmenting human capabilities. By combining accurate drift detection with evolving user acceptance, the system lays a foundation for adaptive, preference-aligned initiative policies. Future work could explore richer learning signals, longer-term deployments, and principled escalation strategies that strengthen behavioral support while preserving user control.

8 Ethical Considerations

There is a clear privacy and data risk since CustomNudge relies on continuous screenshot capture and analysis. While processed locally and via secure APIs, this creates a high-fidelity record of user activity. In a deployed setting, strict data retention policies and on-device processing would be mandatory to prevent surveillance risks.

Further, over-reliance on an external agent for focus regulation could theoretically atrophy a user's intrinsic ability to self-regulate attention. Longitudinal studies are required to understand if users become dependent on the AI for accountability.

Acknowledgments

We are grateful to Professor Diyi Yang and TAs Avanika Narayan and Advit Deepak for their mentorship and thoughtful feedback.

Authorship Statement

Darynne Lee and Eyrin Kim both contributed to the conceptualization, system implementation, and evaluation of CustomNudge. Both authors worked on the GUM integration, decision engine, execution of the user study, and write-up. Eyrin focused on the observation pipeline and macOS integration, while Darynne focused on the evaluation logic and LLM Judge implementation.

References

[1] E. Horvitz. Principles of Mixed-Initiative User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*, pp. 159–166. ACM Press, 1999.

[2] Amazon Web Services. What is Reinforcement Learning from Human Feedback. 2023. Available at `https://aws.amazon.com/what-is/reinforcement-learning-from-human-feedback/`.

[3] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *arXiv preprint arXiv:2305.18290*, 2023. Available at `https://arxiv.org/abs/2305.18290`.

[4] S. Petridis, B. Wedin, J. Wexler, A. Donsbach, M. Pushkarna, N. Goyal, C. J. Cai, and M. Terry. ConstitutionMaker: Interactively Critiquing Large Language Models by Converting Feedback into Principles. *arXiv preprint arXiv:2310.15428*, 2023.

[5] O. Shaikh, S. Sapkota, S. Rizvi, E. Horvitz, J. S. Park, D. Yang, and M. S. Bernstein. Creating General User Models from Computer Use. *arXiv preprint arXiv:2505.10831*, 2025.

[6] M. S. Lam, O. Shaikh, H. Xu, A. Guo, D. Yang, J. Heer, J. A. Landay, and M. S. Bernstein. Just-In-Time Objectives: A General Approach for Specialized AI Interactions. *arXiv preprint arXiv:2510.14591*, 2025. Available at `https://arxiv.org/abs/2510.14591`.

A. K. Dey. Understanding and Using Context. *Personal and Ubiquitous Computing*, 5(1):4–7, 2001.

R. H. Thaler and C. R. Sunstein. Nudge: Improving Decisions about Health, Wealth, and Happiness. Yale University Press, 2008.

P. Klasnja, S. Hekler, S. Shiffman, A. Boruvka, D. Almirall, A. Tewari, and S. A. Murphy. Microrandomized trials: An experimental design for developing just-in-time adaptive interventions. *Health Psychology*, 34(S):1220–1228, 2015.

F. Okeke, M. Sobolev, N. Dell, and D. Estrin. Good Vibrations: Can a Digital Nudge Reduce Digital Overload? In *Proceedings of MobileHCI 2018*, 2018.

U. Lyngs, K. Lukoff, P. Slovak, R. Binns, A. Slack, M. Inzlicht, M. Van Kleek, and N. Shadbolt. Self-Control in Cyberspace: Applying Dual Systems Theory to a Review of Digital Self-Control Tools. In *Proceedings of CHI 2019*, 2019.

A Appendex: Additional Figures

Supplementary screenshots referenced in Figure 3.

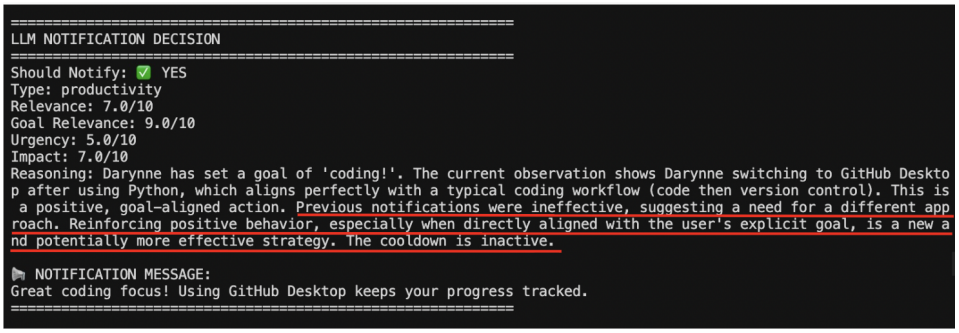

Figure 5: Raw screenshot for the left example in Figure 3.

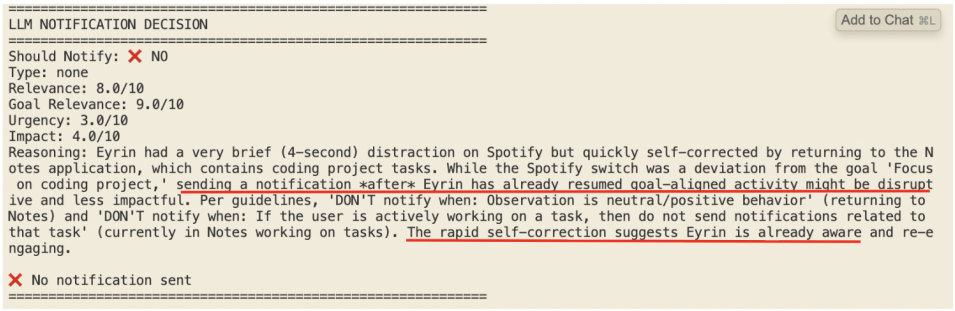

Figure 6: Raw screenshot for the right example in Figure 3.