The Effects of Update Interval and Reveal Method on Writer Comfort in Synchronized Shared-Editors

Yen-Ting Yeh*
y6yeh@uwaterloo.ca
Cheriton School of Computer Science, University of Waterloo
Canada

Nikhita Joshi*
nvjoshi@uwaterloo.ca
Cheriton School of Computer Science, University of Waterloo
Canada

Daniel Vogel
dvogel@uwaterloo.ca
Cheriton School of Computer Science, University of Waterloo
Canada

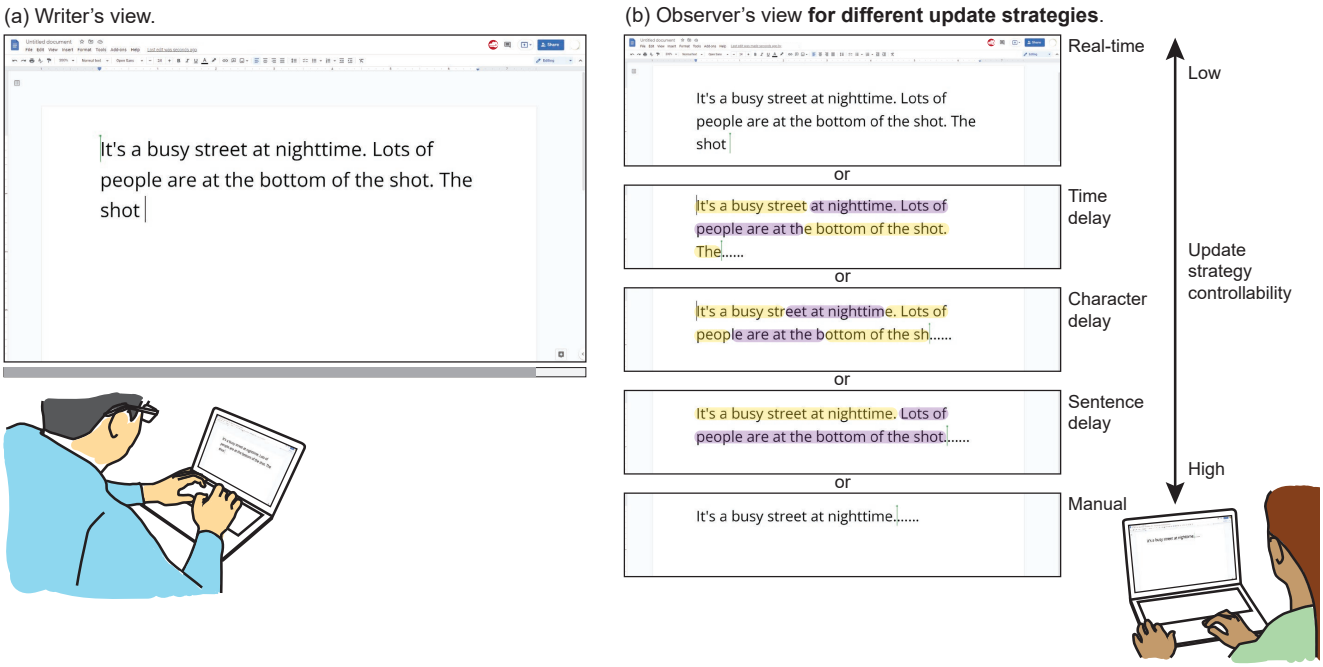

Figure 1: Strategies with different update intervals: (a) writer's view with a timer bar at the bottom; (b) observer's view of different update strategies. Yellow and purple tint illustrate updated text within one interval. Update intervals are ordered top to bottom by controllability (i.e., how much direct control the writer has over the update).

ABSTRACT

Synchronized shared-editors like Google Docs allow people to write together, but there is no "privacy of writing" which can make writers feel uncomfortable. We propose methods to give writers more control over when and how their edits are shown to collaborators to increase comfort. These are in the form of different update strategies composed of an update interval and a reveal method. Results from an experiment with simulated observers show that alternative update strategies can be beneficial, each having their own pros and cons. A follow-up experiment with writer and observer pairs validates these findings and shows that observers are amenable to experiencing short delays caused by alternative update strategies. Our work shows that synchronous writing tools should support alternative update strategies that preserve both collaborator awareness and writer comfort.

CCS CONCEPTS

• **Human-centered computing** → **Collaborative and social computing systems and tools**; **Interaction techniques**.

KEYWORDS

Collaborative writing, Writer comfort, Synchronized shared-editors

ACM Reference Format:
Yen-Ting Yeh, Nikhita Joshi, and Daniel Vogel. 2024. The Effects of Update Interval and Reveal Method on Writer Comfort in Synchronized Shared-Editors. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11–16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3613904.3642330

*Both authors contributed equally to this research.

CHI '24, May 11–16, 2024, Honolulu, HI, USA

1 INTRODUCTION

People often write together using synchronous shared-editors,[1] such as Google Docs or Overleaf, which support real-time interactions between co-writers in a "live" document. Specific writing behaviours, such as moving the cursor to new locations, highlighting text, and editing text, are immediately seen by all collaborators for increased awareness [5, 8]. This can improve collaboration [19, 32], but writers can also feel distracted by collaborator edits or feel concerned about how their edits are perceived by collaborators [20–22]. Wang et al.'s interviews [37] suggest this desire to write privately could be caused by *discomfort* in some writing scenarios.

Since not all editing activity needs to be shown to collaborators in real-time, we explore methods that change when and how edits are revealed in shared-editors. To lower discomfort, we adopt ideas from asynchronous writing tools, such as GitHub or Dropbox, where writers control what and when content is shown to their collaborators by manually "pushing" and merging content into shared documents. We propose modifying the *update interval* (Figure 1) to adjust *when* a writer's edits are shown to others. By giving writers some control over this property, they can briefly write privately to form thoughts and fix typos. Since edits are not immediately updated in the shared-editor, different *reveal methods* control *how* edits are shown to collaborators, such as pasting final text all at once like asynchronous tools. Combining different update intervals and reveal methods creates several *update strategies* that characterize how edits are shared. We investigate these to explore which update strategies could help writers feel more comfortable in synchronous shared-editors, the pros and cons of each, and how these are influenced by user preferences.

We conducted two exploratory experiments using a testbed editor that implements different combinations of three update intervals: time, character, and sentence delays; and three reveal methods: normal typing, which shows every keystroke; perfect typing, which shows every character of the final text when the update was triggered; and pasting, which shows the final text all at once. Two baseline update strategies are also tested: real-time to represent current synchronous shared-editing; and manual to represent asynchronous editing tools. Results from the first experiment with simulated observers suggested that alternative update strategies are beneficial, each with their own pros and cons when it comes to hiding intermediate writing, speed, providing collaborators awareness of edits, naturalness, and mental load. Perfect typing and pasting reveals are highly rated for comfort because writers can erase typos and fix grammatical errors before the shared document is updated. Sentence delays are highly rated across all metrics since writers have more control over when the sentence is ended, which allows writers to think and revise words. A follow-up experiment further validated this in a collaborative writing experience with writer and observer pairs, and showed that observers are amenable to short delays.

[1]Previous work has used different terms like "collaborative editor" [10] or "cooperative writing tool" [38], we use the term "shared-editor" [25] (and the act of 'shared-editing') because sharing encompasses both collaboration and other shared writing scenarios, and 'editor' focuses on the enabling software tool instead of the general activity. We also distinguish between *synchronized* shared editing tools like Google Docs and *asynchronous* shared editing tools like GitHub.

Our contribution is the novel idea of alternative update strategies for synchronous shared-editors, composed of different update intervals and reveal methods to make shared-editing more comfortable, and insights from two exploratory user studies that researchers and designers can build on to make collaborative writing more comfortable.

2 BACKGROUND AND RELATED WORK

Our work is related to synchronous and asynchronous shared-editors, visual delays when collaborating, and "the privacy of writing."

2.1 Synchronous and Asynchronous Shared-Editors

Researchers created synchronous shared-editors long before Google Docs. GROVE [9] allowed multiple writers to work on a shared outline for a text document. ShrEdit [25] introduced private and public edit windows, where public windows can be edited by multiple writers. SASE [1] simplified this to a single view for multiple writers to edit a document synchronously. These early systems set the foundation for today's synchronous shared-editors.

A technical goal for shared-editors is very fast updates for awareness of collaborator activities [5, 8] and to increase collaboration [19, 32]. Olson et al. [26] studied how students used Google Docs and found most took advantage of synchronous features, such as writing together during group meetings. Trust between group members led to using "documents as a place" for information like outlines or main text. These writers could seamlessly work through writing activities described by Posner and Baecker [28], enhanced by more social interactions. However, D'Angelo et al. [4] found that collaborative writers rarely write in the same regions at the same time, likely due to territorial behaviours [21, 31, 37]. Writers may feel uncomfortable writing synchronously, opting to temporarily work privately in another tool instead.

Asynchronous tools let writers collaborate on the same document in private, which could mitigate feelings of discomfort. Quilt [10] assigned collaborators editing permissions (e.g., reading, commenting, writing) and tracked document activity, including edits to the main body and annotations. PREP [24] focused on asynchronous document planning: information was organized into columns, providing space for writers to explain their plan or rationale, the content, and collaborator comments. MESSIE [29] allowed writers to edit shared documents through email with automatic integration of edits into a single document for other collaborators to access. Systems that support document "check-ins" and "check-outs" provide awareness of collaborator activities, which is especially important when attributing credit to someone, or blaming someone if there are issues [37].

Pe-Than et al. [27] examined how the fork-and-pull model, from software version control systems, can be adopted for collaborative writing. Document forks allow multiple asynchronous versions of the same base document to evolve over time for other purposes while providing an awareness of edits without conflicts. Yet, writing asynchronously may make collaborations less focused and take longer [8], or affect document integrity when edits are not integrated across collaborators [29]. A lack of awareness of collaborator

activity may result in duplicate work [5]. Maintenance-related activities, such as discussing and coordinating edits with collaborators are crucial in highly asynchronous writing, such as when contributing to Wikipedia [18, 33].

Synchronous tools enable closer collaborations between writers with less overhead and an increased awareness, while asynchronous tools enable private writing. Our work combines elements from synchronous and asynchronous tools to increase collaboration while providing the ability to write privately.

2.2 Visual Delays in Collaborative Tools

Previous work on delays has been motivated by network latency, leading to a focus on *time* delays. Some previous work has explored using visual delays in collaborative tools, but few have explored their use in writing tools specifically. In the context of collaborative puzzles and games, prior work suggests that time delays can negatively impact coordination [13], but overall, collaborators are able to adapt and adopt new strategies to collaborate effectively [12, 36]. When delays are made known to collaborators, they act accordingly to improve performance. Gutwin et al.'s delay decorators [14] visualize properties related to delays, like the duration, within user interfaces, which helped task coordination and reduced errors.

Ignat et al. [16] explored the effects of a time delay on a collaborative writing task where groups labelled and sorted a list of movies. The system inserted a delay before a writer's edit appeared in other writers' views. Their results showed that groups who labelled before sorting were not affected by the short delay, reinforcing the idea that the effect of a delay depends on the task context. Another study by Ignat et al. [15] explored time delays in collaborative note-taking of content from an audio recording. They found that a time delay can increase grammatical errors and text redundancy.

The task largely impacts delay tolerance. Ignat et al. found some writing tasks requiring high degrees of coordination, like closely editing or consolidating content, may not be as well-suited to longer delays. However, editing in the same region of a document at the same time may not be as common in practice [4, 21], and writing activities may not be as tightly coupled as note-taking or sorting tasks. But collaborators may still observe each other's writing, so including a delay could provide additional comfort. In addition to time delays, there are other ways writing can be delayed, like delays measured by characters or sentences, to give people the ability to write privately. Our work focuses on expanding the types of delays for collaborative writing, with writer comfort as the main motivation.

2.3 The Privacy of Writing

A primary motivation for exploring update strategies is when the collaborative writing context suggests a need for writer privacy, a concept formalized by Wang et al. [37]. Their interviews found that many people do not want to write together, one major reason being individual beliefs that writing should be a private activity. This does not solely mean writing *about* private content; "the privacy of writing" encompasses multiple reasons.

Wang et al.'s interviews showed that some writers felt distracted by constant awareness of collaborator activities, leading them to want to write privately. Interviews by Larsen-Ledet and colleagues [20–22] and Bindley [2] revealed similar writer sentiments, with writers being frustrated by collaborators editing their writing as they were writing, and "jumping" content and cursors in a shared-editor.

Another more subjective reason is that writers may be concerned with the presentation of self. Wang et al. found that writers were often concerned that collaborators would interpret their typos and grammatical errors as signs of struggle. Larsen-Ledet and Borowski [20] and Bindley [2] found that interviewees would only want to share text with their collaborators when it was "ready." Larsen-Ledet et al. [22] found similar results, and note that these feelings may be amplified when a writer has a higher standard to uphold (e.g., a professor not wanting their student to see them making a typo). Conversely, Bindley found writers feel anxious when their superiors are reviewing unpolished drafts. Larsen-Ledet and Korsgaard [21] found that collaborative writers can be territorial over written content, implicitly attributing ownership to specific sections based on previous editing behaviours, even for collaborative writing tasks. The need for privacy can encourage more territorial behaviour.

A common strategy for gaining control is working in a separate asynchronous writing tool and pasting final content in a shared document, which is known as *separation* [21]. Strobl [31] studied how Dutch native speakers with advanced German writing proficiency synthesized articles in German using Google Docs. The results showed that 80% of the shared document had no edit history, suggesting they wrote privately in a separate tool before pasting it in the document. Another common strategy is *demarcation*, indicating private regions using colour and text within the same document to make territories explicit to gain privacy and control when writing [21, 37].

Some previous work has proposed tools to allow people to write privately. Ignat et al. [17] created a shared-editor where writers choose how much of their editing information is seen by their collaborators. Visual filters, like blurring written content, allow writers to edit privately while providing collaborators awareness of live edits being made. Yu et al. [38] created a collaborative editor where writers can create focus regions to write privately, which prevents disturbances while providing awareness when writing. While these works addressed the desire to write privately, they focused on the implementations without evaluating their effectiveness in user studies. Our work investigates more fundamental methods, such as when and how to update the writing, before exploring more advanced features of interface design.

Previous work shows there is a desire to write privately which is caused by discomfort experienced in collaborative environments. We use *comfort* as a metric since it represents the reasons why people write privately (e.g., avoiding distractions and concerns of the presentation of self). Previous work has also shown there are advantages to both synchronous and asynchronous writing in terms of comfort, and there is evidence that users can adapt to time delays caused by network latency. We propose using different update intervals and reveal methods to improve comfort in a shared-editor, a hybrid between behaviours in synchronous and asynchronous text editors.

3 UPDATE STRATEGIES

We explore the spectrum of update strategies in between fully synchronous and fully asynchronous writing by controlling when and how edits are revealed in shared-editors.

3.1 Baseline Strategies

We identify two baseline update strategies that represent synchronous and asynchronous writing tools:

- *Real-Time Updates* — Changes in content are shown to other collaborators in real-time (i.e., no delay), and all editing activities including inserting and deleting a character, or moving a cursor are revealed. This best encapsulates current synchronous shared-editing, like Google Docs and Overleaf.
- *Manual Updates* — Changes in content are shown to collaborators after the user presses a button, only the final content is revealed, and this happens all at once. This approximates methods used by current asynchronous editing tools, like pushing content to a Git repository. It also captures an existing strategy of copying and pasting from another document [31, 37].

3.2 Update Interval

The property that differentiates these two baseline strategies is *when* changes in content are shown to collaborators, which we call the *update interval*. We propose three additional types of update intervals:

- *Time Delay* — Changes in content are shown to other collaborators at regular time intervals. In our editor, we use a time delay of 10s.
- *Character Delay* — Changes in content are shown to other collaborators at fixed character count intervals. In our editor, updates are displayed every 15 characters. Note that the character count would increase or decrease while typing or deleting.
- *Sentence Delay* — Changes in content are shown to other collaborators after every sentence is completed. This is triggered after the writer types end-of-sentence punctuation (i.e., a period, exclamation mark, or question mark).

We selected these delays for the following reasons. Time delays have been explored in collaborative writing before [15, 16], but from a latency perspective rather than intentional time delays that writers leverage as part of their writing process. Characters are the smallest unit of writing that people have control over, and people are used to working within character limits on platforms like Twitter. Small time and character delays may provide enough of a buffer for the writer to make small edits and corrections before the results are shared. Sentences have semantic meaning and represent a complete thought, which is a logical point to share writing that is more developed.

We conducted a pilot survey to learn about suitable delays that could be used in a shared-editor. Respondents were asked to describe an image in 3-5 sentences. In total, 35 responses were used to examine typing speed. On average, they spent 203 seconds writing, used 251 characters, and wrote at 1.7 characters-per-second (equivalent to 20.4 words-per-minute, which also includes time to think and edit). In our editor, assuming at least 15 updates when writing a paragraph to ensure participants have sufficient experience using each update interval, we chose 10 seconds for the time delay and 15 characters as a character delay.

The three alternative update intervals and two baselines form a continuum of user controllability (Figure 1b), which is the degree of control the user has over the timing of the update. A manual button press to update gives the writer total control, but at the cost of an explicit interaction. In contrast, a time delay provides only slightly more control over real-time updates, but requires no explicit interaction.

3.3 Reveal Method

A second property enabled by update interval is *how* changes in content are shown to collaborators, which we call the *reveal method*. When multiple edits are made within a single update interval, the visualization of these edits can be considered in terms of writing realism, writing efficiency, or writer comfort. We implement three reveal methods:

- *Normal Typing* — A simulated typing effect displays all keystrokes as characters or cursor movements performed since the last update. This includes typos and edits to the text that ultimately did not make it into the final content.
- *Perfect Typing* — A simulated typing effect displays only keystrokes that produce the characters in the final content. A text difference algorithm [23] is used to compare two sequential updates. For continuous deleted characters between two updates, the editor would select and delete them together, and for other edits, the editor would show them character-by-character.
- *Pasting* — Content changes are shown all at once.

As an example, suppose someone writes "I like cats" before backspacing to replace "cats" with "dogs" within a single update interval. With normal typing, collaborators would see "I like cats" typed character-by-character, followed by a sequence of cursor movements to delete "cats", and then "dogs" typed character-by-character. With perfect typing, collaborators would only see "I like dogs" typed character-by-character and would have no knowledge of the deletion of "cats." With pasting, the full string "I like dogs" would appear all at once.

Combining update intervals and reveal methods can create distinguishable update strategies. For example, when using a pasting reveal with a character delay, a shared-editor will reveal chunks of fixed length characters as the writer enters content, but when using a sentence delay, entire sentences will be revealed one by one over time. When delay and reveal are combined, they form nine different hybrid update strategies[2].

4 EXPERIMENT 1: WRITER WITH SIMULATED OBSERVERS

The goal of this experiment is to explore how different update strategies affect writer comfort in a shared-editor. To better understand the effect of update strategies, we create a slightly uncomfortable writing environment. Participants were asked to describe images [7] in five sentences and were told that one or more anonymous collaborators would be watching them. Describing images is a form of personal writing as participants are required to elaborate on

[2] All update strategies are demonstrated in the supplementary video.

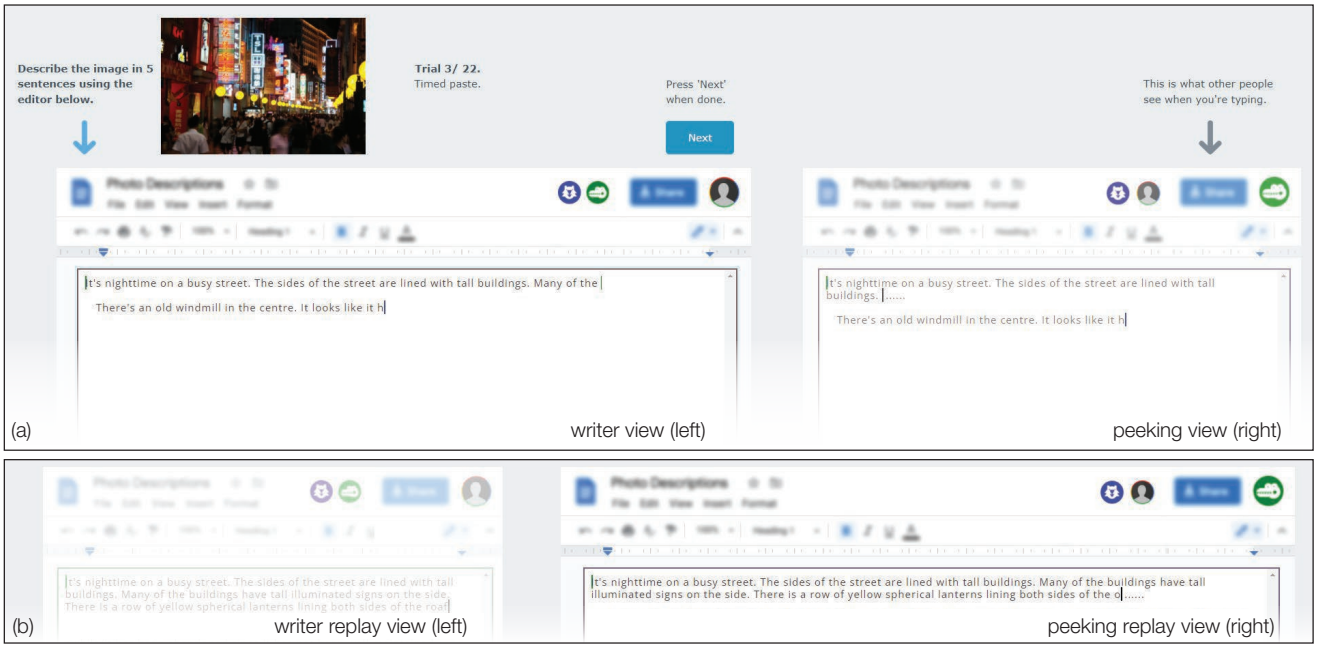

Figure 2: The experiment interface with the testbed shared-editor: (a) writing interface with a writer view where edits are made (left) and a peeking view to show what a collaborator will see (right); and (b) replay interface with a writer replay view (transparent, left) and a peeking replay view (right).

each scene with creative thinking based on their own personal interpretations. This may make some writers feel more territorial over their writing, invoking a need for privacy [21]. Although it happens less frequently, writing while being observed by others is an uncomfortable scenario, which occurs during activities like group brainstorming sessions or when collaborating with others in the workplace [2, 37]. For this initial exploration, we are mainly focused on exploring how individual writers feel using alternative update strategies. We do not consider factors related to coordinating edits, managing conflicts, and how other collaborators would feel knowing alternative strategies are being used, these are explored in a second experiment described later.

4.1 Apparatus

We implemented a simulated testbed shared-editor to explore the effects of combinations of update interval and reveal method in a controlled setting (Figure 2). The testbed shared-editor was implemented as a Node.js web application served to remote users using Ngrok. All keystrokes and mouse events are logged into a MongoDB database to visualize the normal typing method. We created two simulated interfaces, writing and replaying. Participants wrote their paragraph in the writing interface, then rewatched their writing activities to better understand the different update strategies. Source code is available on the project's public repository[3].

4.1.1 Writing Interface. The interface consists of two side-by-side text boxes: a writer view where edits are made, and a "peeking" view to show what a collaborator will see (Figure 2a). Both views are shown within blurred Google Docs interfaces to emphasise the synchronous nature of the writing environment. The image that the participant is asked to describe is shown above the two text boxes.

The writer view is shown on the left and is wider than the peeking view. Here, the user types as usual with content shown in real-time. For the time and character delays, a thin grey progress bar is shown below the writer view notifying them when the next update will occur, it shrinks as time passes or as more characters are typed (for the latter, it grows when the user deletes characters). For the manual baseline, a blue "reveal" button is shown instead. Clicking this will trigger an update. No progress bar or button is shown for the real-time baseline and sentence-based strategies. Immediately following an update, the border of the writer view flashes green and their content is revealed in the peeking view. The peeking view resembles the writer view except edits are shown using the update strategy. If there is a delay, the writer's edits are shown in the peeking view as an ellipsis, providing awareness of the edits without showing the content. Any unrevealed writing is automatically revealed after clicking the "next" button during the experiment.

Both the writer and peeking views show who is viewing the document. The writer view shows a grey user profile picture in the top right corner and two anonymous user profiles beside it. The positions of the user profile and anonymous profile pictures are swapped in the peeking view to emphasize it represents how edits appear to collaborators. Cursors corresponding to the profile picture colours are shown in both views to further increase realism.

Collaborator activities are simulated in both views. First, both views may have a lurking collaborator, a collaborator that is viewing the document but is not contributing. The collaborator's cursor is

[3]https://github.com/exii-uw/co-editing-privacy

shown at the beginning of the text box and does not move. Second, fake photo descriptions are automatically typed in using another collaborator's cursor. These describe another set of photos to avoid bias and they are shown below the area where the participant writes, mimicking the "separate writers" strategy [1]. As our focus is generating new content rather than editing, the participant cannot edit these fake photo descriptions.

4.1.2 Replay Interface. A writer's comfort could be influenced by understanding how their writing would be perceived by other collaborators. Since participants may not pay attention to the peeking view while writing, a replay stage switches their role from writer to observer to better understand how their writing activities are shown to collaborators (Figure 2b). The interface for the replay resembles the main interface, with two side-by-side text boxes: a writer replay view on the left and a peeking replay view on the right. The writer replay is thinner and more transparent, and automatically shows text and edits from all keystrokes exactly as the writer entered while interacting with the main writing interface. The peeking replay functions like the peeking view in the main writing interface, but during replay, the participant can focus on the collaborator view. Collaborator activities are simulated as they were in the main interface.

4.2 Participants

We recruited 23 participants, ages 21 to 42 (M=25.87, SD=5.18), of which 13 were men, 9 were women, and 1 preferred not to disclose. Participants were recruited using our institution's graduate student mailing list and word-of-mouth, and received $50 for successful completion of the study. All but one participant had previous experience using a synchronous shared-editor, with 13 (56%) using one daily or weekly, 8 (35%) using one monthly or less, and 1 participant described prior experience in free-form responses, but did not indicate frequency of use.

4.3 Procedure

The study was conducted remotely online. Each participant joined a video call with the experiment facilitator where they received an explanation of the different update intervals and reveal methods. During this time, they also practised using the different update strategies in the custom editor. The participant was told to practice for as long as they needed, typically 10 to 15 minutes.

Once they felt like they understood the different strategies, they began the main experiment. For a single update strategy, the participant completed Dunlop et al.'s image description task [7]. They were told to make each description five sentences long. Image description tasks have been used in other experiments [11, 30], as they provide additional control and inspiration for writing on demand, which can be difficult otherwise. For every writing task, participants were told they need to write directly in the provided web application and that someone will observe them write in a remote interface. In practice, the observer was the facilitator, but the participant was not told this.

Next, they watched the replay of their previous writing task. After, the participant completed a second image description task using a different image. Finally, they completed a short survey about their experience using the strategy. This was repeated for all eleven update strategies. After completing the main experiment, they answered additional open-ended questions about their experiences and preferences in terms of update interval and reveal method (e.g., *What did you think of the different update strategies? Were there any strategies you liked over others?*) [4]. The entire study lasted 90 to 120 minutes, depending on the participant's typing speed.

4.4 Design

This is a within-subject study design. Although it requires additional mental load from participants to understand all strategies, it enables participants to directly compare the pros and cons of each strategy. Two primary independent variables represented the update interval and reveal method: INTERVAL with 3 levels (TIME, CHARACTER, SENTENCE); and REVEAL with 3 levels (NORMAL, PERFECT, PASTE). Two baseline conditions were also included: REALTIME updates, and MANUAL updates. As such, there are a total of 11 update strategy conditions: (3 INTERVAL × 3 REVEAL + 2 baselines). The order of INTERVAL and the two baselines was counter-balanced using a Latin square and the order of REVEAL was randomized. Each REVEAL was used for a single INTERVAL before the next INTERVAL condition. There were 2 image description tasks per update strategy condition. Each image description used a random image from Dunlop et al.'s image repository [7]. In summary: we recorded 22 completed trials per participant, 506 trials in total.

Questionnaires after each of the update strategies provide 4 subjective measures: *Understanding* is how well the participant understood the update strategy; *Comfort* is how comfortable the participant felt when writing with the strategy; *Ease of Use* is how easy the strategy was to use, and *Satisfaction* is how satisfied the participant was with the update strategy overall. All measures are interval data on a 7-point numerical scale (left-most extreme was anchored with "strongly disagree" and the right-most extreme was "strongly agree," with numbers labelled on all choices).

4.5 Subjective Scores

Results for subjective scores are based on visually examining 95% confidence intervals in Figures 3 and 4 (in all graphs, right-most points are better). Confidence intervals are calculated using the bootstrapping method with 10,000 re-samples. This approach, known as estimation, is becoming recommended within HCI and the scientific community more broadly as it presents the same information as null hypothesis significance testing, but is often clearer and more concise [6]. We opted not to make pairwise comparisons of the conditions as there are many conditions and our goal is not to show which strategy is the best; rather, our goal is to get a sense of the conditions participants liked while writing and explore why they were liked. To get a sense of which conditions were well-received for each metric, we examine which conditions have means above a "neutral" score of 4, and which conditions have entire confidence intervals greater than 5. Five is selected as a cut off as it is on the higher end of the 1-7 scale and a condition with an entire confidence interval greater than 5 suggests that the population from which the sample was taken is likely to also rate it more positively for that metric.

[4] All questions used in the experiment and surveys are included in the supplementary materials.

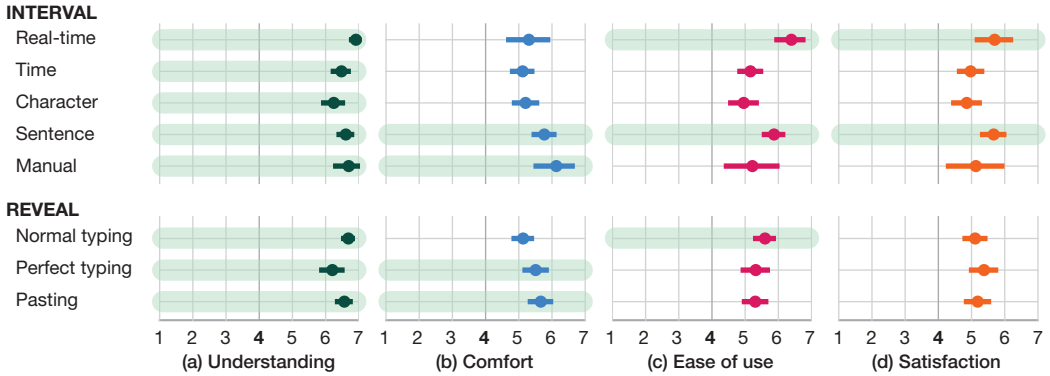

Figure 3: Subjective ratings by INTERVAL and REVEAL condition. Conditions with an entire CI greater than 5 are highlighted in green.

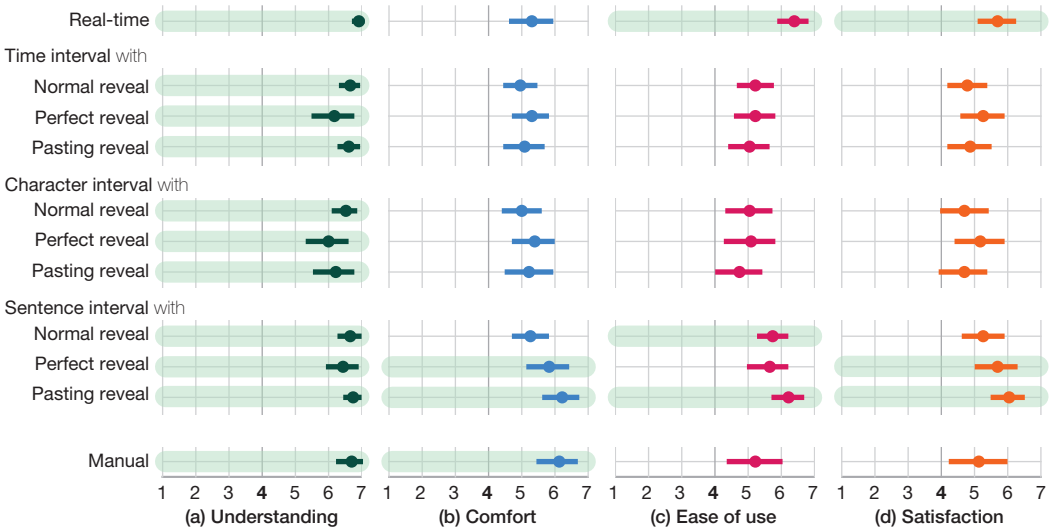

Figure 4: Subjective ratings for specific update strategies defined by INTERVAL and REVEAL, and two baseline strategies: REAL-TIME and MANUAL. Strategies with an entire CI greater than 5 are highlighted in green.

4.5.1 Update Interval and Reveal Methods. We examine overall trends by aggregating the scores across INTERVAL and REVEAL (Figure 3). Each baseline was grouped separately for INTERVAL. For REVEAL, REAL-TIME, and MANUAL were grouped with "normal typing" and "pasting" reveals, respectively. We observe that the means for all metrics are consistently above a neutral response of 4. Some intervals and reveals had entire confidence intervals above 5 (green highlight): all intervals and reveals for *Understanding*; SENTENCE and MANUAL intervals and PERFECT and PASTE reveals for *Comfort*; REAL-TIME and SENTENCE intervals and a NORMAL reveal for *Ease of Use*; and REAL-TIME and SENTENCE intervals for *Satisfaction*.

4.5.2 Specific Update Strategy Combinations. After examining strategies by specific update intervals and reveal methods, we observe that the means for all metrics are consistently above 4, much like the overall trends showed (Figure 4). The following strategies have entire confidence intervals greater than 5 (green highlight): all strategies for *Understanding*; SENTENCE & PERFECT, SENTENCE & PASTE, and MANUAL for *Comfort*; REAL-TIME, SENTENCE & PASTE, and SENTENCE & NORMAL for *Ease of Use*; and REAL-TIME, SENTENCE & PERFECT, SENTENCE & PASTE for *Satisfaction*.

4.6 Open-ended Question Responses

To better understand reasons behind ratings of the update intervals and reveal strategies, we grouped all question responses from the post-experiment questionnaire by INTERVAL and REVEAL and conducted an iterative inductive coding process within each INTERVAL and REVEAL [34]. Baselines were assigned to an INTERVAL and a REVEAL as explained in Section 4.5.1. All open-ended responses were grouped by theme using an open coding approach by the second author, and the first author reviewed the groupings to ensure there was agreement. We propose five main considerations: hiding intermediate writing, perceived speed, collaborator awareness, naturalness, and mental load.

4.6.1 Hiding Intermediate Writing. For 15 participants (65%), certain update strategies helped them conceal errors and re-wordings, which made them feel more confident when writing. Strategies that revealed with perfect typing or pasting generally received positive comments like: *"I thought the perfect typing and paste reveal strategies were interesting and at times nice. I tend to make spelling mistakes so not showing my many deletions is nice"* [P2].

Ten participants (43%) commented that strategies with longer intervals, like sentence delay or manual, provided more control and opportunities to fine-tune writing. Four (17%) participants felt the shorter 10s and 15 character intervals were enough time to make small edits. However, typos made right before an update caused discomfort for two participants (9%) since their editing error remained visible to observers until the next update.

4.6.2 Perceived Speed. Generally, the two simulated typing reveals were perceived to be slower than the pasting reveal. When considering perceived speed of update intervals, shorter ones led five (22%) participants to feel pressured to write. Four (17%) participants noted that they required more time to think about their writing, or sometimes wrote as part of the thinking process with comments like: *"Typing words helps me think of things and in this sense I found the 10s time delay to be more annoying than helpful"* [P4].

Delays that are more dependent on typing activities, like character and sentence delays, may reduce feelings of pressure as writers have more control over when the update occurs. Unlike a time delay, these allow writing slower or faster as needed, suggested by comments like: *"Character count delay means that if I type faster, it updates faster. I think I preferred that since a lot of the time-based updates felt pretty slow"* [P3]. Much like previous work [20–22, 37], one participant (P8) expressed concerns related to presentation of self by worrying they would be perceived as a slow writer with longer update strategies.

4.6.3 Collaborator Awareness. Although participants were not explicitly asked about their experience as observers, many commented on the impact the strategies would likely have on their collaborators. Five (22%) participants suggested faster update intervals made it easier to see a collaborator's writing. But two (9%) noted it matters less for different tasks. When considering the reveal method, four (17%) participants appreciated the speed of pasting and two (9%) saw the technique as a way to "make up" for the additional time spent writing privately. But lost details, like what intermediate edits were made and when, may be confusing to a collaborator, for example: *"[With pasting], if edits were made later, it was confusing as a viewer. Perfect typing was really pleasing [and] made it way more clear as a viewer what the person edited"* [P22].

One participant suggested that the reveal method provides awareness on the quality of thought and felt uncomfortable when they did not align. Specifically, if the thought is meant to be a draft or loosely formed, seeing it typed out perfectly implied overconfidence in an unpolished idea, which can reflect badly on the writer: *"[With perfect typing], a sloppy line of thought looks sloppier"* [P8].

4.6.4 Naturalness. Seven (30%) participants were positive about the real-time update strategy because they were familiar with it. Likewise, simulated typing reveals were familiar, with six (26%) participants commenting how this provided a realistic writer presence compared to the pasting reveal, for example: *"[Perfect typing] hides my typos but still gives a sense that the document is updating GRADUALLY instead of huge multi-line jumps"* [P3]. Another unnatural and distracting feature of some strategies was when updates occurred in the middle of words, which frequently happened with time and character delays. Sentence delays were overall well-received since it uses a semantically meaningful cutoff representing a complete thought, for example: *"[Sentence delays] make me feel the most comfortable because a sentence cutoff is semantically meaningful, I would have corrected spelling/word choice errors before finishing the sentence"* [P11].

4.6.5 Mental Load. Seven (30%) participants commented that it was challenging to anticipate when time and character updates would be triggered. However, for manual updates, the decision of when to trigger an update caused additional mental load: *"I have to think about it, which causes extra cognitive load while I'm trying to focus on writing"* [P11]. Six (26%) participants forgot to press the reveal button, which could negatively impact a collaborative writing task: *"Sometimes I forget to click the button so I feel like my partner will be held up in terms of time and efficiency, especially if they need to know what I am typing in order to type their own work"* [P18]. A sentence delay may be the best compromise as writers can control when to add terminal punctuation, and thus when text should be shared in a way that is in line with their natural writing behaviours, suggested by comments like: *"The sentence delay strategy felt like it allowed for the least amount of errors while also not becoming cumbersome to use"* [P14].

4.7 Summary

Overall, all of the update intervals, reveal methods, and strategies were positively received across all metrics (all > 4) and there are pros and cons to each when it comes to hiding immediate writing, speed, providing awareness to collaborators, naturalness, and mental load.

All delayed updates using a perfect typing or pasting reveal allowed participants to hide their intermediate writing to some extent. Real-time and time-based strategies may not allow enough time to think, and time-based strategies may cause additional pressure when writing. Faster intervals with character-by-character typing reveals may provide more awareness of collaborator activities, but this may come at the expense of a writer's comfort. Typing reveals and cutoffs that are less disruptive to collaborators and are semantically meaningful can make an update appear more natural, which may make writers feel more comfortable. Sentence-based intervals and manual updates may be easier to control than time or character intervals. But manual updates may also increase mental load as the writer needs to decide when content should be shared and needs to remember to press a button to share.

Sentence-based strategies may strike the best balance across all metrics and considerations. Overall, it is the only update interval where all confidence intervals are above 5 for all metrics (Figure 3). Likewise, SENTENCE & PASTE is the only strategy with entire confidence intervals above 5 for all metrics (Figure 4); the only other strategies to have higher confidence intervals for any of the metrics are the two baselines, SENTENCE & NORMAL, and SENTENCE & PERFECT.

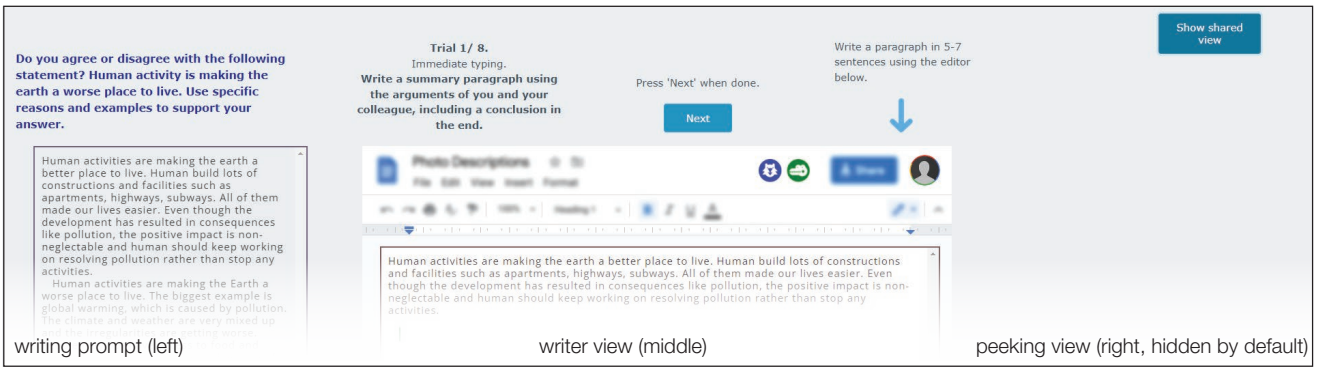

Figure 5: The pair-writing interface in the testbed shared-editor with a text writing prompt (left), a writer view where edits are made (middle), and a peeking view to show what a collaborator will see (right, hidden by default).

4.7.1 Limitations. During the study, participants could see two anonymous collaborators, one editing another paragraph below their writing and one watching them write. We told participants someone would be watching their final text after it was revealed, but it is unclear if participants were convinced and behaved in a wholly realistic way. Participants also speculated on the impact the strategies would have on their collaborators, likely from watching their own writing during the replay stage, but it is unclear if these conclusions match an actual collaborator's experience. To further examine these aspects, we conducted a follow-up experiment where pairs of participants used the most promising update strategies.

5 EXPERIMENT 2: WRITER AND OBSERVER PAIRS

As the first experiment did not feature actual observers, the goal of this follow-up experiment is to corroborate the takeaways from the first experiment by exploring how update strategies influence the collaborative writing experience as both a writer and an observer. This experiment uses a pair-writing task similar to previous work [19] where two participants write simultaneously and collaboratively on a persuasive essay. In our design, they alternated between writing together to formulate arguments and observing each other's writing to summarize their arguments as one paragraph.

To make completing these more demanding and time intensive tasks possible within an experiment session, we selected two of the most promising update strategies from Experiment 1: *Sentence & Perfect* and *Sentence & Paste* as they are the only strategies to have entire confidence intervals above 5 for at least three metrics (Figure 4). We also included the *Real-Time Update* and *Manual Update* strategies as two baselines.

5.1 Apparatus

We modified the writing interface to suit a task where two participants write together (Figure 5). A text writing prompt is used in lieu of an image, supplemented with a text box containing additional content (such as previous writing that needs to be summarized). The peeking view is hidden by default and can be made visible by pressing a button. The writing of both participants is shown in the writer's view, separated into individual paragraphs. Websockets are used to synchronize edits and updates between two writing interface instances, running on different computers for each participant.

5.2 Participants

We recruited 8 participants, ages 21 to 30 (M=24.4, SD=2.7), of which 6 were men and 2 were women. Recruiting used our institution's graduate student mailing list and word-of-mouth, and each participant received $30 upon completing the study. All had previous experience using a synchronous shared-editor (3 selecting weekly or more) and six had previous experience using an asynchronous shared-editor (2 selecting weekly or more). The participants worked in four pairs and paired participants did not know each other prior to participating.

5.3 Procedure

The study was conducted in-person. The two participants recruited for a session introduced themselves, received an explanation of the different update strategies, then practiced using them in the custom editor. During the main experiment task, the pair worked in the same room, but to simulate a remote collaborative setting, they were separated by a divider and could not talk to each other.

In the main task, both participants were given a writing prompt for a persuasive essay, formulated as an agree or disagree question (e.g., *Do you agree or disagree with the following statement? People communicate with each other less than in the past because of the popularity of television. Use specific reasons and examples to support your opinion*). To ensure that all prompts were roughly the same level of difficulty, the questions were taken from samples of the Test of English as a Foreign Language (TOEFL) exam [35]. Each essay was created in two stages. First, one writer chose a stance (agree or disagree) with the other taking the opposite stance (the choice order was alternated during the session). Then, each writer wrote a single paragraph for their stance in the shared document. They were told to respond to and refute each other's arguments in their own paragraphs. This encouraged them to read and watch each other's writing. Second, the participant who chose their stance summarized the two paragraphs from both writers into a single paragraph while the other participant observed. This process took

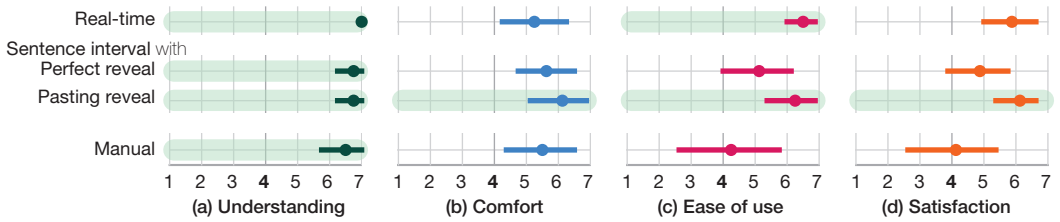

Figure 6: Subjective ratings of update strategies as a *writer*. Strategies with an entire CI greater than 5 are highlighted in green.

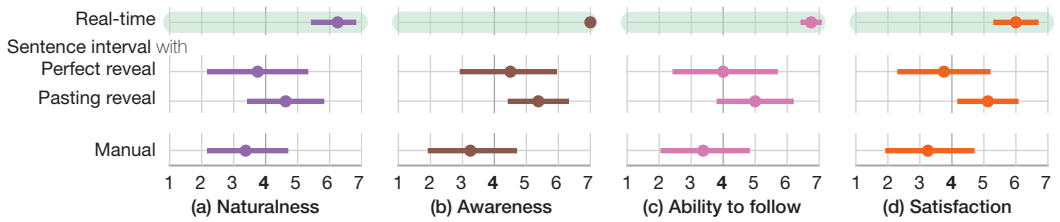

Figure 7: Subjective ratings of update strategies as an *observer*. Strategies with an entire CI greater than 5 are highlighted in green.

place twice, where the role of choosing a stance and summarizing, or observing, was reversed each time. Participants then were asked to complete a short survey about their experience as a writer and as an observer. This was repeated for all four update strategies. At the end of the study, they completed a general questionnaire about their experiences and preferences. [4] The entire study lasted 90 to 120 minutes.

5.4 Design

This is a within subjects design with an independent variable called STRATEGY with four levels: REAL-TIME; MANUAL; SENTENCE & PERFECT; SENTENCE & PASTE. The order of STRATEGY was counterbalanced using a Latin square. The writing prompt for each essay was randomized.

When writing, we use the same four subjective measures as the first experiment: *Understanding*, *Comfort*, *Ease of Use*, and *Satisfaction*. In addition, when observing, we consider four more: *Naturalness* is how natural their collaborator's writing appeared; *Awareness* is how aware they were of their collaborator's writing; *Ability to Follow* is how well they were able to follow their collaborator's line of thinking; and *Satisfaction* is how satisfied they were as an observer overall. These measures were all on a 7-point numerical interval scale.

5.5 Results

Like Experiment 1, we examine 95% confidence intervals for subjective scores (Figures 6 and 7).

5.5.1 Writer. Overall, the subjective scores from a writer's experience align with those in Experiment 1: all strategies had means above 4 for all metrics, indicating more favourable scores. Some strategies had entire confidence intervals above 5: all strategies for *Understanding*; SENTENCE & PASTE for *Comfort*; SENTENCE & PASTE and REAL-TIME for *Ease of Use*; and SENTENCE & PASTE for *Satisfaction*. SENTENCE & PASTE is the only strategy to be consistently highly rated across all metrics, as was the case in Experiment 1. Participant responses to open-ended questions from a writer's view followed the same pattern as Experiment 1.

5.5.2 Observer. Only REAL-TIME and SENTENCE & PASTE have means above 4 for all metrics. SENTENCE & PERFECT has a mean above 4 for *Awareness* and a mean of 4 for *Ability to Follow*. REAL-TIME is the only strategy to have confidence intervals above 5 for all metrics.

Although REAL-TIME was highly rated, the subjective feedback suggests there are some cases where alternative update strategies may be desirable for observers. Two participants noted that observing writers using a real-time strategy made them less comfortable with the writer's content: *"The real-time strategy made me more concerned. I got to see him going back and forth over the text, having typos, and changing stuff. It somewhat indicates that even the writer does not think that the writing is fine"* [P7]. Some observer discomfort may also arise if an update strategy disrupts their reading flow, which may happen more with a REAL-TIME strategy, for example: *"It would make me feel bit disturbed if I am continuously reading what is coming on the screen and understanding it, and suddenly someone backspaces the whole line or some words and write it again. It breaks my reading flow"* [P6]. In cases where high awareness is important, a collaborator that forgets to reveal their writing when using a manual strategy can be distressing for observers, who are forced to catch up, shown in comments like: *"I was not able to catch up with my collaborator's writing as he pressed [the reveal button] only once in a paragraph"* [P5]. Observers did not believe they would judge their collaborators for using alternative update strategies when writing, noting expectations were lower for certain tasks (e.g., drafting) and that there was a mutual understanding of discomfort: *"I myself worry about how my work appears to others so I am less likely*

Table 1: Summary of benefits and trade-offs for different update intervals and reveal methods.

Real-Time (status-quo)	Does not hide intermediate writing. Fast speed. High awareness of edits. Natural-looking. Easy to understand and use, but no control. Can break an observer's reading flow.
Manual update	Can hide a lot of intermediate writing. Slower speed. Slower speed may provide less awareness. Easy to understand and control, but more thought required to reveal. Collaborators need to catch up with large reveals.
Time Delay interval	Can hide some intermediate writing. Fast for some tasks. Fast speed may provide better awareness. Unnatural cutoffs. Not as easy to understand and control.
Character Delay interval	Can hide some intermediate writing. Fast, but can be controlled by typing. Fast speed may provide better awareness. Unnatural cutoffs. Not as easy to understand and control.
Sentence Delay interval	Can hide a lot of intermediate writing. Slower speed. Slower speed may provide less awareness. Natural cutoffs. Easy to understand and control.
Normal Typing reveal	Does not hide intermediate writing. Slower speed. High awareness of edits. Natural-looking and natural presence.
Perfect Typing reveal	Can hide lots of intermediate writing. Slower speed. High awareness of larger edits, but low awareness of smaller edits and level of thought. Natural-looking edits and natural presence.
Pasting reveal	Can hide lots of intermediate writing. Fast speed. Low awareness of edits. Unnatural jumps in the document.

to judge someone if they choose a strategy which gives them more control over what I can see as a collaborator" [P8].

5.6 Summary

To summarize, the feedback from the pair-writing experiment corroborates with the results from Experiment 1: writers rated all strategies highly, with SENTENCE & PASTE being consistently highly rated for all metrics. REAL-TIME was highly rated across all metrics for observers, but subjective feedback suggests that showing all edits may cause discomfort even among observers. MANUAL was consistently rated low by observers.

6 DISCUSSION

Overall, our exploratory studies suggest that alternative update strategies could help writers and observers feel more comfortable when writing collaboratively, justifying *the need for synchronous shared-editors that support alternative strategies.* Our studies suggest that each strategy has their own benefits and trade-offs that designers need to consider, which we summarize in Table 1.

6.1 Example Usage

Based on the benefits shown in Table 1, we provide suggestions for when and how different strategies could be used in a shared-editor. Future work should validate the effect of scenarios on strategy preferences.

For the two baseline strategies, real-time updates are ideal when editing the same region of the document, such as *revising and polishing a paper*, as this provides the fastest shared-context and highest awareness to ease the burden of managing conflicts between collaborators. Manual updates are not well-suited to writing that requires close coordination due to the awareness cost and an increased burden for other collaborators to "catch up" with potentially a large amount of new writing, like editing within the same region of the document or writing about the same content. But when a document could be shared with a stranger (e.g., *writing a company-wide article*) or collaborating with someone that a writer wants to impress (e.g., *working on a project proposal with supervisors* [2]), it provides the most control and a comfortable experience.

When considering the update interval, sentence intervals work well with a wide variety of writing tasks and collaborative settings (e.g., *writing different paragraphs within the same section simultaneously*), since writers can complete a full thought delimited by natural syntax. Time-based and character-based updates provide opportunities for hiding small edits like typos while maintaining a fast update speed, especially since they are not restricted by terminal punctuation. They could be used for informal writing tasks, such as *brainstorming* or *taking notes and summarizing a discussion during a conference call.*

When considering reveal methods, perfect typing or pasting reveal methods are desirable since they hide intermediate edits, such as when *trying to formulate a thought while ideating* or *live coding in front of the class.* However, a normal typing reveal can be useful to communicate the writer's thought process due to its natural presence. Writing tasks such as *creative writing to increase a reader's empathy* [3] can benefit from that. Pasting reveals are fast, but unnatural jumps in text can be frustrating for observing collaborators and potentially obfuscate smaller edits. A pasting reveal is likely better for *drafting*, with *editing* and *proofreading* more compatible with a typing reveal.

6.2 Conflicting Factors

Participant comments from the experiments reveal five main factors to consider, but many conflict. Slower update intervals provide more opportunities to hide more intermediate writing, which can increase comfort, but may require additional mental load through an explicit interaction and reduces collaborator awareness. Typing-based reveal methods appear natural and provide collaborators an awareness of smaller edits, but are slower than pasting reveals. There are also tensions between the needs of the writer and observer: the writer likely would prioritize hiding more intermediate writing with slower update strategies but the observer would likely prioritize faster and more natural strategies that provide better awareness. Ultimately, future shared-editors need to consider the best way to balance these points of tension between factors. One

possibility is defaulting to sentence-based strategies and allowing writers to change the update interval and reveal on demand.

6.3 Alternative Intervals and Reveals

There are many other types of update intervals and reveals that may further improve comfort in synchronous shared-editors and possibly resolve some tensions. Word delays could work like sentence delays by revealing new writing to collaborators after a space is typed, and there is a wide range of time and character delays. Manual updates, which we used to represent an asynchronous writing baseline, could be combined with typing reveal methods to appear more natural to collaborators. Different update intervals and reveal methods could also be combined. For example, if a user forgets to press the reveal button for manual updates, a secondary time-based interval could be used. A hybrid between normal and perfect typing reveals that hides small typos but shows larger edits like structural changes to content could balance providing more awareness about the writer's thought process, while still preserving their comfort.

6.4 Applicability Across Tasks

Collaborative writing tasks fall along a continuum, ranging from highly asynchronous, like contributing with others on Wikipedia, to highly synchronous, like simultaneously editing a paper right before a deadline. At the extremes of this continuum, some update strategies may not always be beneficial. A completely asynchronous writing task has no expectation of simultaneous writing, so there is no need for alternative update strategies to increase comfort. In contrast, highly synchronous writing tasks rely on high frequency updates for awareness to avoid conflicts. However, there is a wide range of writing tasks along this continuum where alternative update strategies could be beneficial, such as writing synchronously but in separate regions. For these in-between writing contexts, alternative update strategies could help improve writer and observer comfort. Furthermore, longer writing tasks often move through several stages from initial brainstorming and early stage drafting to making minor revisions and fixing typos in later stages.

6.5 Limitations and Future Work

We discuss limitations and possibilities for future work.

6.5.1 Ecological Validity. Our experiments may be lacking ecological validity. Specifically, the strategies were tested using a specific "separate writing" [1] context where each collaborator worked on their own image descriptions or persuasive arguments, but this is not representative of all writing scenarios. Future work should explore the effects of update strategies on other types of collaborative writing. Our pair-writing experiment featured only 4 pairs (8 participants in total), which seemed reasonable to further explore and validate the results from the first experiment. However, the perspective of observers in our experiment may not be ecologically valid and requires additional exploration.

6.5.2 Language Proficiency. Stobl's work [31] suggests that language proficiency may contribute to writers desiring privacy when writing. This research was conducted at an English-speaking institution. Although some participants did not consider English to be their first language, most self-reported native or bilingual proficiency. As such, we cannot examine potential correlations between language proficiency and comfort while writing with alternative update strategies, but future work could examine this effect with a more diverse sample.

6.5.3 Visualization of Shared-View. We used a simple ellipsis to indicate that some parts of the writing paragraph have not yet been updated into the shared view. This indicator could be expanded to provide more awareness, such as changing the number of dots to indicate how much of the writer's content has not been shared with their collaborators. Other visualizations such as displaying abstracted writing blocks [17] could address awareness issues.

6.5.4 Integration with Existing Tools. Alternative update strategies could be integrated into existing online shared-editing tools through editor-specific add-ons. A Google Docs extension could provide a panel that allows writers to adjust the update interval, reveal, and typing speed for typing-based reveals. All new writing would be subject to the settings set by the writer. Alternatively, writers could mark regions in a document by update strategy, allowing them to quickly switch between strategies as they move to write in different parts of the document. Machine learning could be used to automatically refine or select the best update strategy depending on the proximity of collaborators in-text.

7 CONCLUSION

We explore the idea of using different update strategies to improve writer comfort in synchronous shared-editors, which are defined by the update interval and reveal method. A testbed shared-editor implemented different kinds of update delays and reveal effects. This enabled two controlled experiments, one examining writers with simulated observers and another with pairs of people collaborating on a writing task as both writer and observer. Our work shows that update intervals other than the status quo real-time update can be desirable, each with their own pros and cons, and we hope our results will inform the future design of synchronous shared-editors.

ACKNOWLEDGMENTS

This work was made possible by NSERC Discovery Grant 2018-05187 and Canada Foundation for Innovation Infrastructure Fund 33151 "Facility for Fully Interactive Physio-digital Spaces".

REFERENCES

[1] Ronald M. Baecker, Dimitrios Nastos, Ilona R. Posner, and Kelly L. Mawby. 1993. The User-Centered Iterative Design of Collaborative Writing Software. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems* (Amsterdam, The Netherlands) *(CHI '93)*. Association for Computing Machinery, New York, NY, USA, 399–405. https://doi.org/10.1145/169059.169312

[2] Katherine Bindley. 2022. The Five Most Dreaded Words at the Office: 'Let's Start a Google Doc'. Shared documents are a workplace minefield of performance anxiety and awkward interactions. *The Wall Street Journal* (2022). https://www.wsj.com/articles/google-doc-sharing-work-collaboration-11654612774

[3] Dashiel Carrera and Sang Won Lee. 2022. Watch Me Write: Exploring the Effects of Revealing Creative Writing Process through Writing Replay. In *Creativity and Cognition* (Venice, Italy) *(C&C '22)*. Association for Computing Machinery, New York, NY, USA, 146–160. https://doi.org/10.1145/3527927.3532806

[4] Gabriele D'Angelo, Angelo Di Iorio, and Stefano Zacchiroli. 2018. Spacetime Characterization of Real-Time Collaborative Editing. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 41 (November 2018), 19 pages. https://doi.org/10.1145/3274310

[5] Paul Dourish and Victoria Bellotti. 1992. Awareness and Coordination in Shared Workspaces. In *Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work* (Toronto, Ontario, Canada) *(CSCW '92)*. Association for Computing Machinery, New York, NY, USA, 107–114. https://doi.org/10.1145/143457.143468

[6] Pierre Dragicevic. 2016. Fair Statistical Communication in HCI. *Modern Statistical Methods for HCI* (2016), 291–330. https://doi.org/10.1007/978-3-319-26633-6_13

[7] Mark D Dunlop, Emma Nicol, Andreas Komninos, Prima Dona, and Naveen Durga. 2016. Measuring inviscid text entry using image description tasks. In *Proceedings of the 2016 CHI Workshop on Inviscid Text Entry and Beyond*.

[8] Clarence A. Ellis and Simon J. Gibbs. 1989. Concurrency Control in Groupware Systems. *SIGMOD Rec.* 18, 2 (June 1989), 399–407. https://doi.org/10.1145/66926.66963

[9] Clarence A. Ellis, Simon J. Gibbs, and Gail L. Rein. 1990. Design and use of a group editor. *Engineering for Human-Computer Interaction* (1990), 13–25.

[10] Robert S. Fish, Robert E. Kraut, and Mary D. P. Leland. 1988. Quilt: A Collaborative Tool for Cooperative Writing. *SIGOIS Bull.* 9, 2–3 (April 1988), 30–37. https://doi.org/10.1145/966861.45414

[11] Margaret Foley, Géry Casiez, and Daniel Vogel. 2020. Comparing Smartphone Speech Recognition and Touchscreen Typing for Composition and Transcription. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3313831.3376861

[12] Darren Gergle, Robert E. Kraut, and Susan R. Fussell. 2006. The Impact of Delayed Visual Feedback on Collaborative Performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Québec, Canada) *(CHI '06)*. Association for Computing Machinery, New York, NY, USA, 1303–1312. https://doi.org/10.1145/1124772.1124968

[13] Carl Gutwin. 2001. *The Effects of Network Delays on Group Work in Real-Time Groupware*. Springer Netherlands, Dordrecht, 299–318. https://doi.org/10.1007/0-306-48019-0_16

[14] Carl Gutwin, Steve Benford, Jeff Dyck, Mike Fraser, Ivan Vaghi, and Chris Greenhalgh. 2004. Revealing Delay in Collaborative Environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vienna, Austria) *(CHI '04)*. Association for Computing Machinery, New York, NY, USA, 503–510. https://doi.org/10.1145/985692.985756

[15] Claudia-Lavinia Ignat, Gérald Oster, Olivia Fox, Valerie L. Shalin, and François Charoy. 2015. How Do User Groups Cope with Delay in Real-Time Collaborative Note Taking. In *ECSCW 2015: Proceedings of the 14th European Conference on Computer Supported Cooperative Work, 19-23 September 2015, Oslo, Norway*, Nina Boulus-Rødje, Gunnar Ellingsen, Tone Bratteteig, Margunn Aanestad, and Pernille Bjørn (Eds.). Springer International Publishing, Cham, 223–242. https://doi.org/10.1007/978-3-319-20499-4_12

[16] Claudia-Lavinia Ignat, Gérald Oster, Meagan Newman, Valerie Shalin, and François Charoy. 2014. Studying the Effect of Delay on Group Performance in Collaborative Editing. In *Cooperative Design, Visualization, and Engineering*, Yuhua Luo (Ed.). Springer International Publishing, Cham, 191–198. https://doi.org/10.1007/978-3-319-10831-5_29

[17] Claudia-Lavinia Ignat, Stavroula Papadopoulou, Gérald Oster, and Moira C. Norrie. 2008. Providing Awareness in Multi-Synchronous Collaboration without Compromising Privacy. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work* (San Diego, CA, USA) *(CSCW '08)*. Association for Computing Machinery, New York, NY, USA, 659–668. https://doi.org/10.1145/1460563.1460665

[18] Aniket Kittur, Bongwon Suh, Bryan A. Pendleton, and Ed H. Chi. 2007. He Says, She Says: Conflict and Coordination in Wikipedia. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 453–462. https://doi.org/10.1145/1240624.1240698

[19] Grete Helena Kütt, Kevin Lee, Ethan Hardacre, and Alexandra Papoutsaki. 2019. Eye-Write: Gaze Sharing for Collaborative Writing. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300727

[20] Ida Larsen-Ledet and Marcel Borowski. 2020. "It Looks Like You Don't Agree": Idiosyncratic Practices and Preferences in Collaborative Writing. In *32nd Australian Conference on Human-Computer Interaction* (Sydney, NSW, Australia) *(OzCHI '20)*. Association for Computing Machinery, New York, NY, USA, 339–354. https://doi.org/10.1145/3441000.3441032

[21] Ida Larsen-Ledet and Henrik Korsgaard. 2019. Territorial Functioning in Collaborative Writing: Fragmented Exchanges and Common Outcomes. *Comput. Supported Coop. Work* 28, 3–4 (jun 2019), 391–433. https://doi.org/10.1007/s10606-019-09359-8

[22] Ida Larsen-Ledet, Henrik Korsgaard, and Susanne Bødker. 2020. Collaborative Writing Across Multiple Artifact Ecologies. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376422

[23] Eugene W. Myers. 1986. An O(ND) Difference Algorithm and Its Variations. *Algorithmica* 1 (1986), 251–266. https://doi.org/10.1007/BF01840446

[24] Christine M. Neuwirth, David S. Kaufer, Ravinder Chandhok, and James H. Morris. 1990. Issues in the Design of Computer Support for Co-Authoring and Commenting. In *Proceedings of the 1990 ACM Conference on Computer-Supported Cooperative Work* (Los Angeles, California, USA) *(CSCW '90)*. Association for Computing Machinery, New York, NY, USA, 183–195. https://doi.org/10.1145/99332.99354

[25] Judith S. Olson, Gary M. Olson, Marianne Storrøsten, and Mark Carter. 1993. Groupwork Close up: A Comparison of the Group Design Process with and without a Simple Group Editor. *ACM Trans. Inf. Syst.* 11, 4 (Oct 1993), 321–348. https://doi.org/10.1145/159764.159763

[26] Judith S. Olson, Dakuo Wang, Gary M. Olson, and Jingwen Zhang. 2017. How People Write Together Now: Beginning the Investigation with Advanced Undergraduates in a Project Course. *ACM Trans. Comput.-Hum. Interact.* 24, 1, Article 4 (March 2017), 40 pages. https://doi.org/10.1145/3038919

[27] Ei Pa Pa Pe-Than, Laura Dabbish, and James Herbsleb. 2021. Open Collaborative Writing: Investigation of the Fork-and-Pull Model. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW1, Article 137 (April 2021), 33 pages. https://doi.org/10.1145/3449211

[28] Ilona R Posner and Ronald M Baecker. 1992. How people write together (groupware). In *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*, Vol. iv. IEEE, New York, NY, USA, 127–138 vol.4. https://doi.org/10.1109/HICSS.1992.183420

[29] Martina Angela Sasse, Mark James Handley, and Shaw Cheng Chuang. 1993. Support for Collaborative Authoring via Email: The MESSIE Environment. In *Proceedings of the Third European Conference on Computer-Supported Cooperative Work 13–17 September 1993, Milan, Italy ECSCW '93*. Springer Netherlands, Dordrecht, 249–264. https://doi.org/10.1007/978-94-011-2094-4_17

[30] Korok Sengupta, Sabin Bhattarai, Sayan Sarcar, I. Scott MacKenzie, and Steffen Staab. 2020. Leveraging Error Correction in Voice-based Text Entry by Talk-and-Gaze. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3313831.3376579

[31] Carola Strobl. 2014. Affordances of Web 2.0 technologies for collaborative advanced writing in a foreign language. *Calico Journal* 31, 1 (2014), 1–18. https://www.jstor.org/stable/calicojournal.31.1.1

[32] Yunting Sun, Diane Lambert, Makoto Uchida, and Nicolas Remy. 2014. Collaboration in the Cloud at Google. In *Proceedings of the 2014 ACM Conference on Web Science* (Bloomington, Indiana, USA) *(WebSci '14)*. Association for Computing Machinery, New York, NY, USA, 239–240. https://doi.org/10.1145/2615569.2615637

[33] Jennifer Thom-Santelli, Dan R. Cosley, and Geri Gay. 2009. What's Mine is Mine: Territoriality in Collaborative Authoring. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) *(CHI '09)*. Association for Computing Machinery, New York, NY, USA, 1481–1484. https://doi.org/10.1145/1518701.1518925

[34] David R. Thomas. 2006. A General Inductive Approach for Analyzing Qualitative Evaluation Data. *American Journal of Evaluation* 27, 2 (2006), 237–246. https://doi.org/10.1177/1098214005283748

[35] toeflresources.com. 2022. sample-toefl-essays. https://www.toeflresources.com/sample-toefl-essays/.

[36] Ivan Vaghi, Chris Greenhalgh, and Steve Benford. 1999. Coping with Inconsistency Due to Network Delays in Collaborative Virtual Environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (London, United Kingdom) *(VRST '99)*. Association for Computing Machinery, New York, NY, USA, 42–49. https://doi.org/10.1145/323663.323670

[37] Dakuo Wang, Haodan Tan, and Tun Lu. 2017. Why Users Do Not Want to Write Together When They Are Writing Together: Users' Rationales for Today's Collaborative Writing Practices. *Proc. ACM Hum.-Comput. Interact.* 1, CSCW, Article 107 (December 2017), 18 pages. https://doi.org/10.1145/3134742

[38] Weihai Yu, Gérald Oster, and Claudia-Lavinia Ignat. 2017. Handling Disturbance and Awareness of Concurrent Updates in a Collaborative Editor. In *Cooperative Design, Visualization, and Engineering*, Yuhua Luo (Ed.). Springer International Publishing, Cham, 39–47. https://doi.org/10.1007/978-3-319-66805-5_5