AnnotateGPT: Designing Human–AI Collaboration in Pen-Based Document Annotation

Benedict Leung
benedict.leung1@ontariotechu.net
Ontario Tech University
Oshawa, Canada

Mariana Shimabukuro
mariana.shimabukuro@ontariotechu.ca
Ontario Tech University
Oshawa, Canada

Christopher Collins
christopher.collins@ontariotechu.ca
Ontario Tech University
Oshawa, Canada

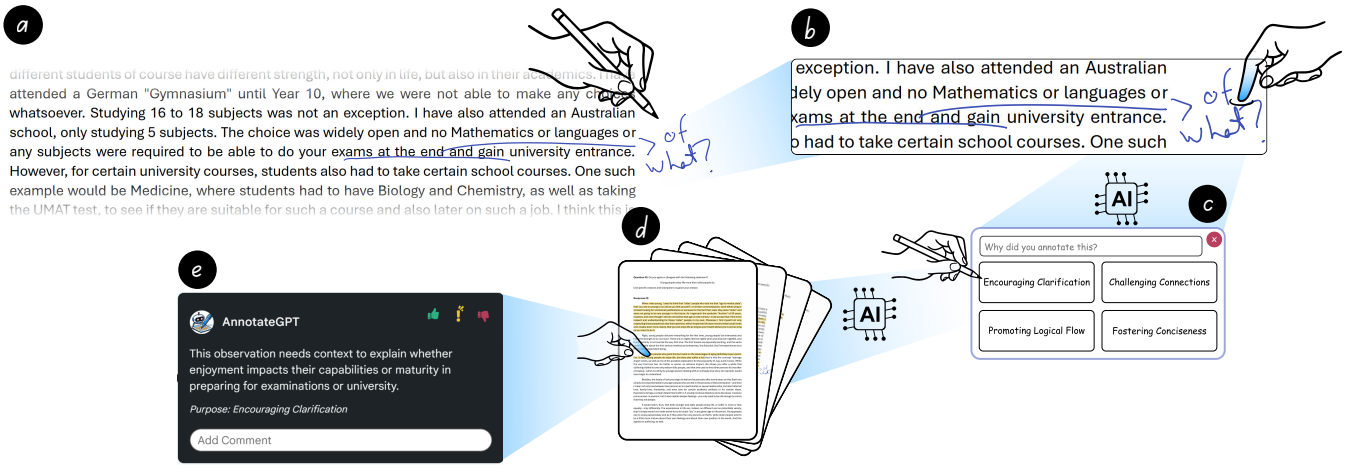


Figure 1: A high-level overview of the interaction design of AnnotateGPT. (a) The user manually annotates the document. (b) Tapping on an annotation will activate the assistant. (c) The assistant will guess the purpose of the annotation. (d) Selecting a purpose will prompt the assistant to provide further annotations (yellow highlights) based on the selected purpose. (e) Users can read, verify and continue the feedback.

Abstract

Providing high-quality feedback on writing is cognitively demanding, requiring reviewers to identify issues, suggest fixes, and ensure consistency. We introduce AnnotateGPT, a system that uses pen-based annotations as an input modality for AI agents to assist with essay feedback. AnnotateGPT enhances feedback by interpreting handwritten annotations and extending them throughout the document. One AI agent classifies the *purpose* of each annotation, which is confirmed or corrected by the user. A second AI agent uses the confirmed purpose to generate contextually relevant feedback for other parts of the essay. In a study with 12 novice teachers annotating essays, we compared AnnotateGPT with a baseline pen-based tool without AI support. Our findings demonstrate how reviewers used annotations to regulate AI feedback generation, refine AI suggestions, and incorporate AI-generated feedback into their review process. We highlight design implications for AI-augmented feedback systems, including balanced human-AI collaboration and using pen annotations as subtle interaction.

CHI '26, Barcelona, Spain

ACM ISBN 979-8-4007-2278-3/2026/04
https://doi.org/10.1145/3772318.3790867

CCS Concepts

• **Human-centered computing** → **Collaborative interaction**; *Gestural input*; **Pointing devices**; *User centered design*; *User studies*; • **Computing methodologies** → *Multi-agent systems.*

Keywords

annotation, digital pen, LLM, feedback

ACM Reference Format:
Benedict Leung, Mariana Shimabukuro, and Christopher Collins. 2026. AnnotateGPT: Designing Human–AI Collaboration in Pen-Based Document Annotation . In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 25 pages. https://doi.org/10.1145/3772318.3790867

1 Introduction

Providing high-quality feedback on writing is a cognitively demanding task. Reviewers must identify issues, provide practical suggestions, and maintain consistency throughout the document. Teachers often value handwritten annotations for their personal and specific tone [9]. Yet, these same practices struggle with legibility, time pressure, and inconsistent quality [5, 6, 9, 13, 49]. Digital annotation tools mitigate some of these problems but often reduce annotations to static marks, missing the opportunity to treat them as interactive, actionable inputs rather than final outputs.

Annotations are a routine part of reading and reviewing, guiding attention, aiding memory, and recording reflections in paper, or printed media [36] and, more recently, on digital formats [42]. Although annotating is a highly personal practice, users' workflows often follow recognizable patterns such as highlighting to signal future attention [36, 37]. Digital tools, such as Adobe Acrobat [1], support highlighting and sticky notes, while tools like loomp [23] allow annotations to be tagged semantically (e.g., question, discussion). Despite these tools, annotation remains burdensome, and pen-based feedback continues to be valued for its authenticity [9, 13]. Other AI-driven supportive tools, such as Grammarly [19], provide grammar and style suggestions, but typically at the sentence level, prioritizing correctness over capturing a reviewer's intent or integrating seamlessly with annotation practices.

The rise of transformer architectures [53] has enabled LLMs to possess incredible language and vision capabilities, and has been integrated into everyday tools, such as ChatGPT [60]. This presents an interaction design opportunity that leverages a familiar method, pen annotations, as implicit signals for reviewers' intent, guiding AI support. We present *AnnotateGPT*, a pen-based annotation system that treats handwritten marks not only as feedback in themselves but also as clues for AI collaboration. AnnotateGPT was designed to amplify reviewers' intent while lessening the burden of manual annotation. It interprets clusters of pen strokes, infers their likely purpose (e.g., revision request, praise, clarity issue), and generates contextually relevant feedback across the document. In doing so, it positions annotations as a form of implicit interaction that balances human judgment with AI scalability.

This paper makes three contributions: (1) we introduce *AnnotateGPT*, a pen-based system that integrates LLMs into document annotation through purpose inference and feedback propagation; (2) we report empirical insights from a study with 12 novice teachers that reveal how reviewers manage, appropriate, and negotiate AI-augmented feedback; and (3) we present design implications for AI-augmented feedback systems, framing pen annotations not only as feedback but as a broader interaction design paradigm for human-AI communication.

2 Related Work

Research shows that annotations support comprehension and reflection. However, education research emphasizes that effective feedback often suffers from issues such as legibility, timeliness, and lack of guidance. To address these challenges, AnnotateGPT uses LLMs to enhance both document annotation and pedagogical feedback, aiming to improve clarity, efficiency, and reflective engagement.

2.1 Studies of Annotation

Annotations are idiosyncratic and polymorphic, meaning they are specific to their process and can serve different purposes even for a single reader [18, 38]. Thus, studies have been conducted to better understand the nature of annotations. For instance, Marshall's analysis of student textbook annotations [36, 37] demonstrates how marks guide attention, aid memory, and structure engagement. With the rise of digital documents, annotation tools have expanded to support highlights, freeform ink, and semantic tags [17, 20, 39, 42, 51, 57].

Despite the move to digital, few systems process annotations directly. Noteable precursors to our work include XLibris [18] and Metatation [38]. XLibris [18] is a pen-enabled tablet display that uses single-stroke annotations as implicit information retrieval queries to find research papers related to the annotated text. Metatation [38] analyzes spatiotemporal stroke patterns and the underlying poem text to generate real-time, context-specific expanded annotation throughout the poem. While these works demonstrate the potential of implicit annotation-driven retrieval and augmentation, they are domain-specific, rely on heuristic pattern matching, and cannot infer the purpose of annotations beyond their predefined contexts.

More recently, LLM-driven systems can extend alternative inputs for creative tasks, such as Code Shaping [59], where sketches are used to generate code. In contrast, AnnotateGPT supports pen-based, text-centric annotation and review, producing knowledge artifacts rather than code. Together, these approaches highlight the diverse ways LLMs can augment human annotation practices, from creative authoring to reflective evaluation, and motivate the need for tools that scaffold interpretation in feedback workflows.

2.2 Challenges in Providing Feedback to Students

As educational institutions shift from traditional to digital annotation methods, teachers and students often prefer handwritten comments for their personal tone [9]. Yet, this practice is constrained by three recurring problems: poor legibility, limited time for detailed responses, and inconsistent or overly negative tone.

The degraded legibility of annotations is primarily due to physical and temporal constraints that affect the delivery of feedback to students and teachers. Students also struggle to interpret the annotations due to poor readability and lack of clarity [6, 13]. On the other hand, teachers are unable to provide detailed, thoughtful feedback due to time constraints and the importance of prompt return to students [5, 9, 49], affecting the quality of feedback. Feedback also tends to fixate on surface-level errors rather than providing balanced guidance [2, 44, 62].

This work focuses on education, where AnnotateGPT aims to address these three issues by having an LLM collaborate in the annotation process, offering legible, timely, and constructive feedback. Annotations are enhanced with AI-generated insights that preserve the personal tone of human comments while improving clarity and depth. The LLM supports teachers by drafting comments based on their annotations, reducing manual labour and enabling more consistent, high-quality responses. This hybrid approach retains the personalized nature of handwritten feedback while mitigating its common pitfalls, ultimately enhancing both the efficiency and pedagogical impact of annotation practices.

2.3 Human-AI Collaboration for Writing

Beyond annotation tools, a growing body of work explores how AI can support writing. Rather than replacing human abilities, these systems aim to complement them, fostering collaboration that balances strengths and weaknesses [30, 48, 61].

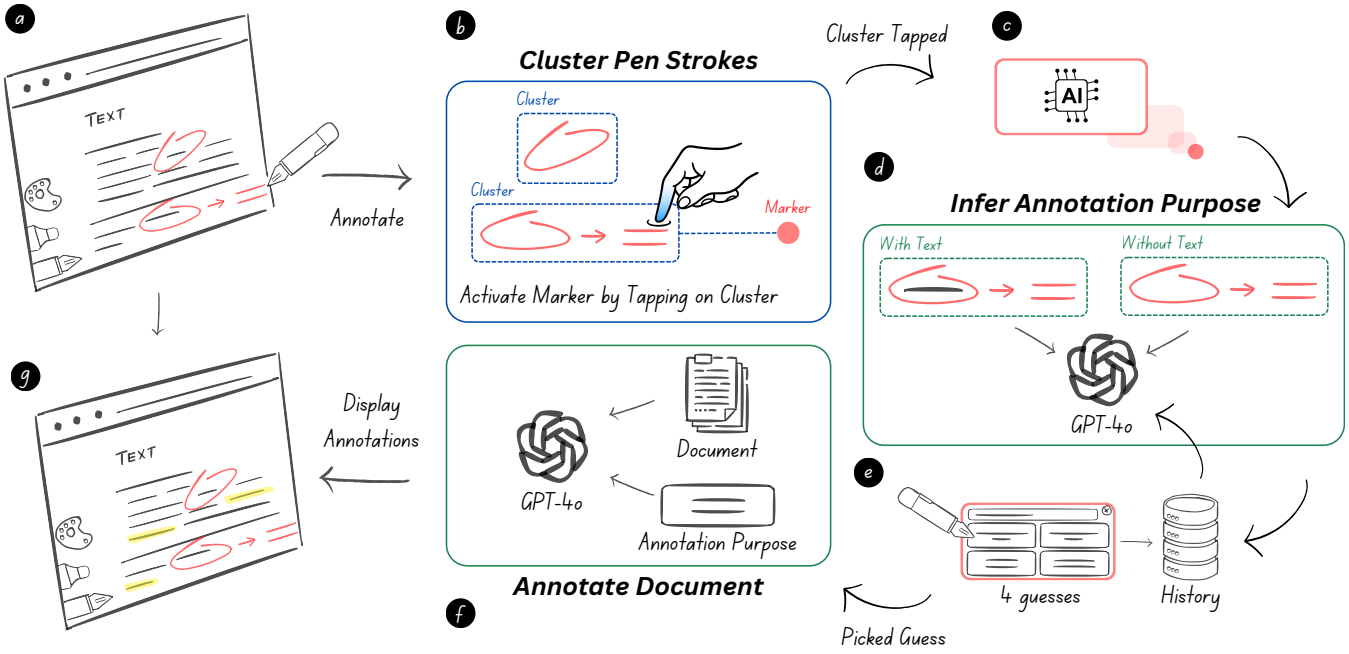


Figure 2: An overview of AnnotateGPT's framework: (a) The user first annotates the document. (b) AnnotateGPT then clusters the pen strokes based on spatiotemporal distance, representing an annotation. (c) The user taps on the cluster/annotation to activate and open the assistant. (d) The assistant captures two images from the cluster, one with the underlying text and one without, and makes four guesses about the annotation's purpose. (e) The user then selects a purpose, which AnnotateGPT will remember for future inferences. (f) Finally, AnnotateGPT generates annotations based on the selected purpose and (g) highlights them on the document.

LLMs have been used as collaborative agents in creative contexts, such as story ideation or combining text with AI-generated images to support creative flow [12, 14, 14, 16, 24, 47, 48, 50, 55]. Similarly, AnnotateGPT positions the model as a partner in sensemaking, rather than generating new content. It collaborates with readers by interpreting pen annotations and expanding them into clearer, more actionable feedback.

Mainstream applications, such as Grammarly and ChatGPT, scaffold revision through grammar, style, clarity, and tone suggestions [19, 58]. However, their unprompted or bulk suggestions can overwhelm writers and blur the boundary between human and machine contributions. Research prototypes, such as InkSync [29], address this by embedding AI edits within documents, allowing users to accept, verify, or audit changes through a chat interface, thus improving transparency and trust.

AnnotateGPT differs in framing pen annotations as implicit commands that guide the AI. This design places control with the reviewer, ensuring AI suggestions extend rather than replace human judgment. By grounding augmentation in users' own annotations, AnnotateGPT offers a more purposeful and interpretable integration of LLMs into feedback workflows.

3 AnnotateGPT

This work introduces AnnotateGPT, a pen-based annotation tool to assist teachers in providing high-quality feedback on writing. It leverages two LLM-powered agents: (1) to infer the form and purpose of annotations, and (2) to generate additional low-level, context-specific feedback. Pen strokes are classified and clustered into annotations, each represented by a marker. When activated, agent (1) proposes four purposes; the user selects one, which triggers agent (2) to generate feedback. Figure 2 shows the overall framework.

3.1 Stroke Clustering

Pen annotations are often idiosyncratic, informal, and ambiguous, making it difficult to determine where one ends and another begins. Prior systems required explicit pauses [15, 40], or single-stroke drawings [31], limiting natural interactions. Alternatively, clustering algorithms group strokes into distinct annotation clusters, without prior knowledge of cluster count or shape [11, 27, 38]. Drawing from past approaches, we adopt hierarchical agglomerative clustering with single linkage [11, 38], which groups strokes by spatiotemporal distance; combining spatial and temporal differences to iteratively merge the closest strokes into annotations (Figure 2b). To determine the number of clusters in advance, the merging would stop if there is a sharp jump in distance, indicating a "forced merge". A virtual stroke is added to the centre of the page to prevent fragmentation of a lone complex annotation. This method has been applied to diverse domains such as freehand drawings [27]

Document: ... a student will be able to get more of the grades and could get more attention if he has the collection of various innovative ideas accompanied with the facts told by him

vs.

ChatGPT: A student will be able to get more attention if he has various innovative ideas accompanied with the facts told by him.

Figure 3: Illustration of the difference in the document and ChatGPT output. Therefore, a more rigorous approach is necessary to accurately match sentences from ChatGPT to the document.

and document annotation [38] to cluster pen strokes. Formal definitions of the spatiotemporal distances and the stopping criterion are provided in Appendix G.

3.2 Stroke Classification and Text Extraction

Each stroke, defined as a sequence of digital pen xy coordinates, is heuristically classified to support text extraction for LLM input. This work classifies three types:

1. HORIZONTAL LINES with three subclasses: HIGHLIGHTING, CROSSING OUT, and UNDERLINING: Horizontal lines are identified using y-coordinate proximity and variance thresholds to account for jitter, then sub-typed based on relative position to words.
2. CIRCLING: Enclosures are detected by sliding windows at stroke ends and "loose" intersections between segments to accommodate imperfect enclosures, with text extracted if at least half the content is enclosed.
3. ANNOTATED: All other strokes default to this category, capturing nearby text within the bounding box or the closest paragraph if the bounding box is empty, ensuring robustness to drawing errors.

3.3 Annotation Classification and Purpose Inference

Stroke classifications, extracted text, and cluster images are fed into an LLM, which proposes four possible annotation purposes (Figure 2d). A system prompt specifies stroke types, extracted text, and common annotation categories [36, 37], guiding the model to infer purpose from context. Prior annotation history is integrated through retrieval-augmented generation (RAG) to personalize interpretations, summarize its findings, and produce alternative explanations using different personas [7]. Through pilots, annotations can elicit richer forms of engagement with the document, navigating between detailed textual cues and higher-level themes. To better accommodate the user's unique annotation styles, it aims to understand how the user engages with content on a word level, determining whether their intent is specific to the text or more general. This is only invoked when no stroke annotates more than two words.

The user prompt then includes two images: one shows the underlying text, while the other isolates the annotation, reducing visual noise and clarifying overlapping handwritten marks. Since annotations may contain multiple types, the ANNOTATED label is reserved for cases where no other type applies. By combining structured hints, multi-branch reasoning, and personalization, the system guides the LLM toward accurate, context-sensitive interpretations, consistent with prior evidence that structured inputs improve

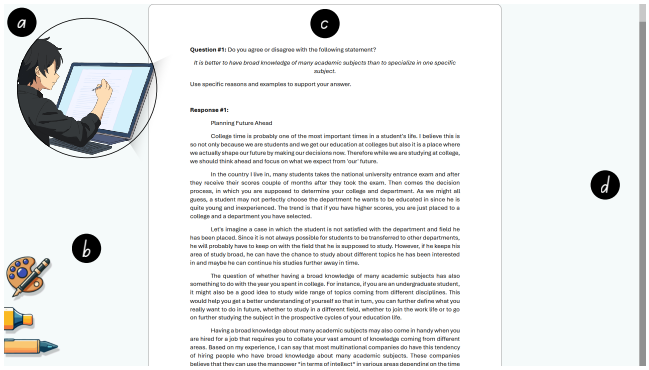


Figure 4: (a) Illustration of a user annotating on the Microsoft Surface Studio desktop. Prototype interface of AnnotateGPT comprises three components: (b) the toolbar (from top to bottom: colour palette, highlighter, and pen), (c) the document, and (d) the specialized scrollbar.

reasoning [34]. The resulting output is four possible interpretations of the annotation's purpose, expressed at both specific (word-level) and broad (conceptual) levels of granularity, or all four at the broad level (see Table 5 for prompts).

3.4 Generating Low-Level Context-Specific Annotations

Once a user selects a purpose, the LLM generates annotations consisting of a target sentence, its target words within, and the associated feedback (see Table 6 for prompts). During testing, ChatGPT occasionally omitted words in its RAG output yet still returned valid sentences, requiring a more flexible approach to finding the correct passage to highlight (see Figure 3).

After ChatGPT produces a new annotation, AnnotateGPT needs to find the relevant passage to mark it in the interactive interface. The search algorithm works across two scopes (*single-page* and *cross-page*) and at two levels (*exact* and *fuzzy*):

Single-page. Extracts all text from one page simultaneously.

Cross-page. Sentences can cross over to the next page. Therefore, this scope extracts N characters from the end of one page and the start of the next, where N is the length of the targeted sentence.

Exact. Searches for an exact match within the document. Non-alphanumeric characters are filtered out and removed (e.g., commas and periods) to optimize computation.

Fuzzy. In some cases, ChatGPT outputs a valid sentence with words omitted, where *exact* search cannot detect. To handle this, a fuzzy search is applied using the Levenshtein distance [32]. A match is accepted if similarity exceeds $N/2$ characters, ignoring non-alphanumeric symbols. At this level, the sentence shown in Figure 3 would correctly align with the document.

The algorithm proceeds from single-page / exact to cross-page / fuzzy, advancing if no match is found. The same approach is applied to previously found texts to avoid duplicates that the assistant may output.

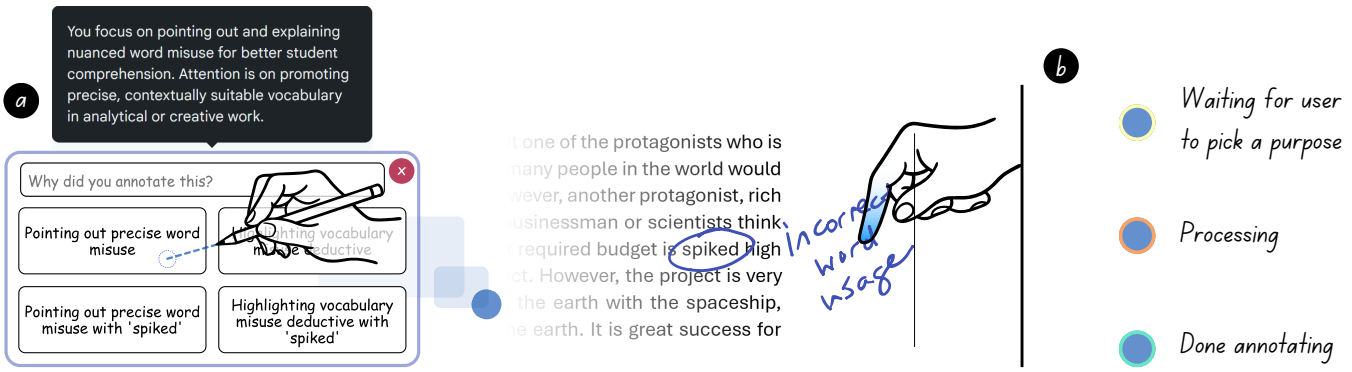


Figure 5: (a) Tapping an annotation will open an assistant marker on the left side of the document, displaying suggested purposes. An input box is also provided for the user to type a purpose. Hovering over an option will provide more details about the purpose. (b) The assistant marker has three possible states: *waiting for purpose* (yellow), *processing* (orange), and *done* (green).

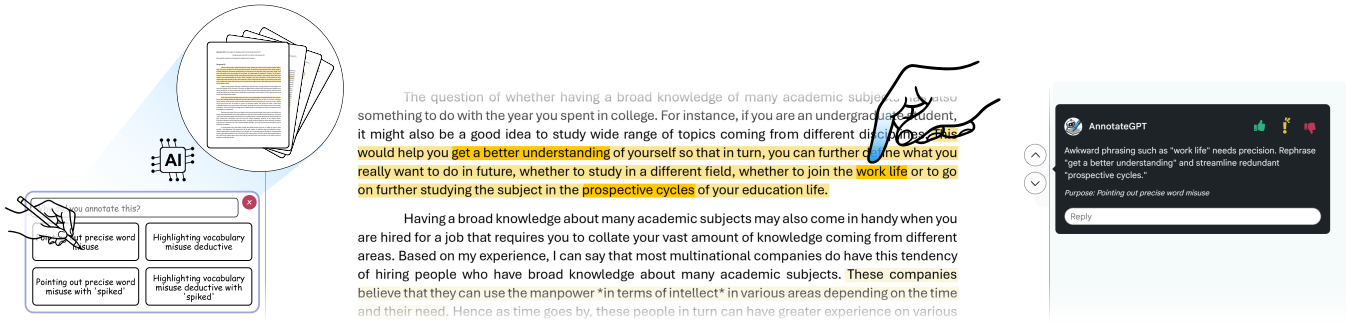


Figure 6: Once the user inputs a purpose, AnnotateGPT will generate additional annotations and display them in yellow highlights across the document. Tapping on the highlights will display a tooltip, where users can read the associated feedback by AnnotateGPT. Additionally, they can reply to continue the feedback. To verify the annotation, three ratings are given (left to right): *accept*, *helpful*, and *reject*.

3.5 Implementation

AnnotateGPT was implemented as a web application in Next.js [54]. The OpenAI API [41] executes prompts with `gpt-4o` for generating annotations and `gpt-4o-mini` for purpose inference, both integrated with RAG. The interface comprises three components: a toolbar, the document, and a specialized scrollbar (see Figure 4). The toolbar includes a colour palette, highlighter, and pen, which sits on the non-dominant side (i.e. on the left for right-handed users) for simultaneous thumb and pen interaction for effective mode switching [45]. The source code is available at https://github.com/vialab/AnnotateGPT.

3.5.1 Document Parsing. The system parses the PDF into two layers: (1) the image layer, where it displays an image for each document page, and (2) the text layer, where it displays text that is not visible to the user. The text layer aligns the position of the text with the image. It also parses the text into words where each word and character is marked by `<span>` tags in HTML for text extraction (subsection 3.2) and displays AnnotateGPT's annotations (subsection 3.4).

3.5.2 Assistant Marker. Each annotation cluster (subsection 3.1) is assigned a hidden assistant marker that serves as the entry point for AI support. The marker is revealed when the user taps on a cluster with their finger, making interaction lightweight and consistent with pen-based workflows. This gives the users agency over when the assistant is needed [30, 61].

Once activated, the cluster is processed (subsection 3.3). When the purpose inference is complete, it displays four possible purposes for the annotation. If none are suitable, users can type their own purpose in the provided input box (see Figure 5a). After a purpose is confirmed, the assistant generates further annotations based on that choice. The marker's states (waiting, processing, done) are colour coded (yellow, orange, green) to keep the user aware of progress (see Figure 5b). This design enables users to seamlessly switch between manual annotation and AI-augmented feedback, without disrupting their workflow.

3.5.3 Verifying AnnotateGPT's Annotations. Lastly, the user may review the annotations made by AnnotateGPT. AnnotateGPT's annotations appear in yellow highlights, whereas the darker yellow highlights indicate the words that are the focus of the annotation,

as seen in Figure 6. Upon tapping, a tooltip displays the associated feedback for the sentence. Sometimes, a sentence can be annotated multiple times from separate assistant invocations, in which the tooltip will display all feedback.

The user can give one of three ratings inside the tooltip. (1) *accept* if the annotation matches the purpose. Otherwise, (2) *reject*. (3) If the annotation gave helpful feedback but does not match the purpose, they can rate it *helpful*. Rating it *accept* or *helpful* will change the highlights to green (and dark green), while *reject* will remove the annotation entirely.

A reply box is provided in the tooltip to extend AnnotateGPT's comments, support dialogue, and help refine the AI's feedback, as interactive annotation systems encourage clarifications and responses often missing in footnote-style interfaces [57]. The arrow buttons beside the tooltip will navigate to the previous or next annotation. Alternatively, the user can tap another annotation.

3.6 Specialized Scrollbar

As the assistant markers process, the user can continue to annotate the document. To assess progress from assistant markers and quickly navigate annotations that require attention, a specialized scrollbar was designed to assist in navigation. It displays the locations and states of annotations and assistant markers. The scrollbar is separated into three columns: left, middle and right (see Figure 7).

Left. Shows the location and colour of the user's annotations.

Middle. Shows the state (based on colour, see Figure 5b) and location of the assistant markers.

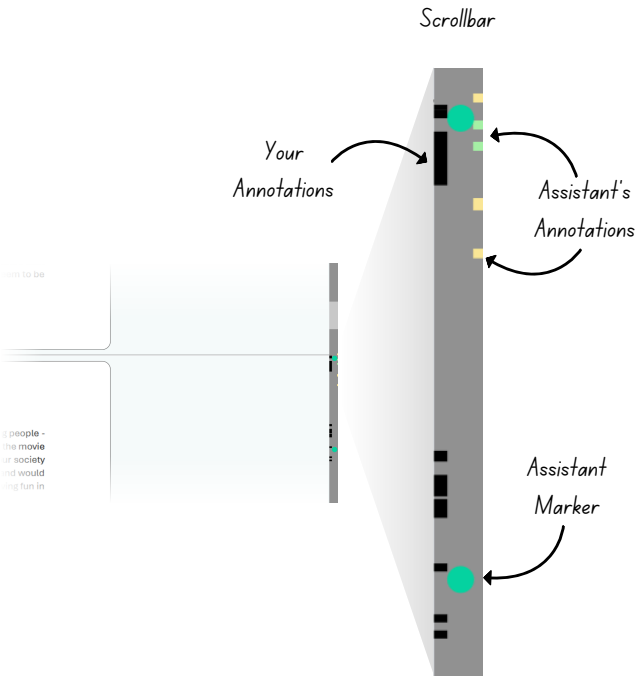


Figure 7: A specialized scrollbar to display the location and state of the annotations and assistant markers.

Right. Shows the location and state (*accept*, *helpful*, or *unrated*) of the assistant's annotations.

4 User Study

Our goal was to explore annotations as a form of human-AI communication. We aimed to understand how it supports annotation practices, facilitates the generation of annotations, and integrates into existing workflows. To frame this goal in a practical context, we chose to focus on feedback for English essays, a well-established scenario that requires nuanced, text-based annotations, as discussed previously. We conducted a comparative user study to evaluate the assistive effectiveness of AnnotateGPT compared to traditional digital annotation interfaces. The baseline system supported only free-form, digital ink annotations (i.e., AnnotateGPT without the assistant marker). Our study was approved by our institutional Research Ethics Board (File #18136).

4.1 English Tests Creation

The data comes from essays written in response to two standardized English tests, created using the ETS Corpus of Non-Native Written English [8]. This dataset comprises 12,100 essays written by non-native English speakers as part of the TOEFL (Test of English as a Foreign Language) English proficiency exams conducted in 2006 and 2007. Each essay is categorized into one of eight questions (e.g., "Young people enjoy life more than older people do") associated with a low, medium, or high score level.

From each of the eight questions, we randomly selected an essay with a high score level and compiled them into a single document. A spelling and grammar checker was applied to reduce the number of superficial annotations. The document is then exported into a PDF file. This process was repeated for the second document curation to ensure that participants worked with different documents across the systems, as well as the training document, which had only one question. The resulting three PDFs were used for the final study documents.

4.2 Participants

We recruited 12 pre-service teachers and teachers (1 left-handed), of whom nine are aged 18–24 and three are aged 25–34 with a self-declared minimum proficiency level of upper-intermediate English (B2 on the CEFR). Participants were recruited by a mass email sent to all students from the Faculty of Education. The participation criteria included full mobility of the hand and wrist to allow for handwriting with a pen, normal or corrected-to-normal vision (e.g., glasses or contact lenses), and the ability to read and comprehend English. Recruitment was conducted on a first-come, first-served basis. Participants received the equivalent of $40 CAD for their time.

Based on the screening questionnaire, participants had 1 to 7 years of teaching experience (MDN = 1). Among them, eight individuals teach in STEM education, two teach English and history, one focuses on educational studies, and one teaches kindergarten. Furthermore, participants shared their experiences with text annotation: for paper, 2 do it daily, 7 weekly, 1 monthly, 1 yearly, and 1 never; for digital annotation, 3 do it daily, 5 weekly, 1 monthly, 2 yearly, and 1 never.

4.3 Apparatus & Software

Using the Surface Pen, participants sat before the Microsoft Surface Studio 2 (28" touchscreen). They adjusted the screen until they felt comfortable with its position. The web/study application ran locally, and video recordings were made of the screen and over their shoulders to assess qualitative factors of system performance and capture quotes.

4.4 Study Design

The study followed a within-subjects study design with one primary independent variable, TECHNIQUE, with two levels (ANNOTATEGPT and BASELINE). QUESTIONS form secondary independent variables with 8 levels/questions. For each TECHNIQUE, each participant will mark one of two English tests consisting of 8 QUESTIONS. The order of the TECHNIQUE and English test order is counterbalanced to minimize learning and fatigue effects — half start with BASELINE and half start with English test 1.

The primary measures taken included the strokes made, annotation ratings (*accept*, *reject*, *helpful*) and interactions from the assistant marker (purpose inference and generated annotations). Additionally, questionnaires and interviews provided subjective measures.

4.5 Tasks & Procedure

Pre-study & Instructions. To begin, participants were informed about the work's objective and signed a consent form. They then adjusted the screen for comfort and completed a demographics questionnaire.

Training. Participants watched a video tutorial (~2 minutes for the BASELINE, and another ~2 minutes for ANNOTATEGPT) on the features of each technique. They then practised the technique on the training document. The following tasks must be completed before proceeding (BASELINE only has the first task): (1) Make an annotation. (2) Activate the assistant marker. (3) Generate additional annotations using the marker. (4) Navigate the annotations. (5) Verify the annotations. Participants can continue to practice until they are comfortable.

Annotating. Participants had 30 minutes to mark and annotate the English test by giving feedback, ensuring realistic working conditions, consistent time constraints across techniques, and reduced fatigue effects. For ANNOTATEGPT, digital ink is disabled after 25 minutes to allow them to finalize their ratings (*accept*, *reject*, *helpful*). Additionally, participants received a time update every 10 minutes. The same updates were given for the BASELINE. Participants were not required to finish annotating all questions and rating all annotations, as the priority is to evaluate the overall experience and performance rather than task completeness.

Questionnaires. After each technique, participants completed questionnaires, including NASA-TLX [21, 22] and SUS [10]. The study used a 7-point and 5-point Likert scale for TLX and SUS, respectively. Additional questionnaires asked about the participants' self-perceived performance in evaluating the test and the features of AnnotateGPT.

Interview. A post-study semi-structured interview was conducted to gather information about (1) which system they prefer and to provide examples of why, and (2) what issues they sought during the test. Additional questions were asked based on observed participants' trends, such as annotation behaviours and comments on feedback quality. The study took approximately 90 minutes to complete.

5 Results

We report findings on how participants annotated using AnnotateGPT and the baseline. We describe common workflows, annotation types, and behaviours, followed by quantitative measures of annotation density, duration, usability, workload, and participant perceptions of feedback quality.

5.1 Data Analysis

Following previous annotation work [36–38, 59], the analysis employed an inductive thematic analysis to examine all collected data, including annotations, system logs, questionnaires, observational notes, screen recordings, and audio recordings of interviews. To complement the qualitative analysis, statistical analyses were conducted using the Wilcoxon signed-rank test and ART ANOVA [56], along with its post-hoc tests using the Holm-Bonferroni correction. We report 95% confidence intervals, estimated through resampling methods (bootstrapping, 10,000 iterations) to better capture uncertainty with our sample size.

5.2 Workflows Observed

In our analysis of collaboration with AnnotateGPT, we identified two interrelated workflows: (1) *annotation workflows*, which described how participants created and submitted their annotations, and (2) *interaction workflows*, which detail how they engaged with AI-generated annotations during verification and refinement. These workflows reveal the strategies participants used, such as sequential or batched annotation, and their approaches to sharing agency with the AI for generation, verification, or follow-up tasks.

There were three common workflows regarding how participants manually annotate to collaborate with AnnotateGPT:

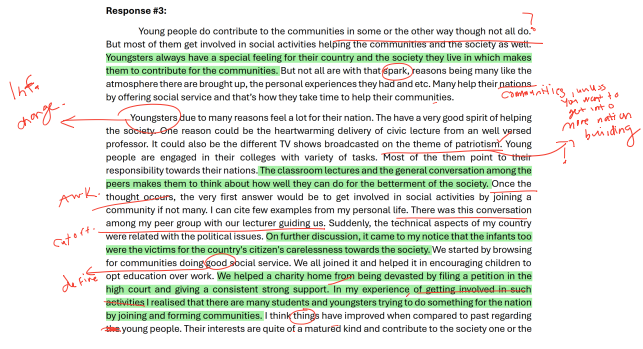
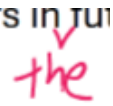
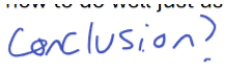
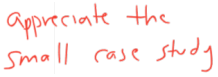


Figure 8: Example of how P6 fills in the gaps of the automated annotations, where the annotations were filled around the highlights.

Table 1: Types of annotations observed in the study are placed into two dimensions: form and purpose. Each type has the number of counts observed for the baseline annotations (N_B), AnnotateGPT user-authored annotations (N_A), purpose inference (N_P), and expanded annotations generated from purpose inference (N_E).

	Category	Subcategory	Description	N_B	N_A	N_P	N_E	Example	
Purpose	Grammar	Tense	Correct and consistent use of verb forms.	110	31	13	182		P4
		Preposition	Proper use of prepositions in phrases and expressions.						
		Punctuation	Accurate use of commas, periods, etc.						
		Capitalization	Correct use of uppercase or lowercase letters.						
	Vocabulary	Word Choice	Suggesting precise, context fitting words.	93	33	9	139		P9
		Spelling	Correct spelling of words.						
		Collocation	Suggesting a better natural combination of words.						
	Sentence Structure	Clarity	Suggesting unclear or confusing sentences.	157	59	17	305		P12
		Run-ons	Avoid long, improperly joined sentences.						
		Fragment	Avoids incomplete sentence fragments.						
	Organization & Coherence	Logical Flow	Ideas are disjointed and lack clear flow.	94	26	25	407		P5
		Paragraphing	Proper use of paragraphs to group related ideas.						
	Task Achievement	Completeness	Fully responds to the question or task requirements.	19	10	1	16		P2
		Encouragement	Offering constructive and encouraging feedback.						
Form	Type	Telegraphic	A personal opaque coding.	93	73	48	—	see Figure 9, right	
		Explicit	Clear and explicit meaning, usually textual.	380	86	17	—	see Figure 9, left	

1. ANNOTATE-INTERPRET: Participants (N = 10) makes one annotation, then has the assistant interpret it and waits for the result.
2. ANNOTATE-K-INTERPRET-K: Participants (N = 9) makes K annotations, then has the assistant interpret them concurrently. Participants used this workflow to initially focus on making annotations while the assistant processed previous ones, or they first annotated the entire question and then allowed the assistant to interpret multiple annotations afterward.
3. ANNOTATE-FOLLOWUP: Participants (N = 5) make an annotation to fill in the gaps for AnnotateGPT, to address anything AnnotateGPT has overlooked, where AnnotateGPT did not highlight, as seen in Figure 8.

In addition to the annotation workflow, participants have three common workflows for how they interact with AnnotateGPT:

1. GENERATE-VERIFY: Participants (N = 6) generate additional annotations based on one of their annotations and verify afterwards. Participants used GENERATE-VERIFY to provide feedback on the first question and then propagate the same feedback to the other questions using AnnotateGPT one at a time.
2. GENERATE-K-VERIFY: Participants (N = 9) generated additional annotations based on K of their annotations and verified all at once. Participants used this workflow to first manually annotate the first question without using AnnotateGPT, then interpret and generate at the end.
3. VERIFY-COMMENT: Participants (N = 7) verified and continue to comment on the annotation.

5.3 Types of Annotations

The types of annotations can be placed into two dimensions: *form*, which ranges TELEGRAPHIC-EXPLICIT, and *purpose*, which ranges MICRO-MACRO (see Table 1). The TELEGRAPHIC-EXPLICIT range describes whether the annotations are personal opaque codings versus explicitly textual feedback. The MICRO-MACRO range describes whether the annotation targets fine-grained textual features (such as spelling) versus broader, structural aspects of the text (such as

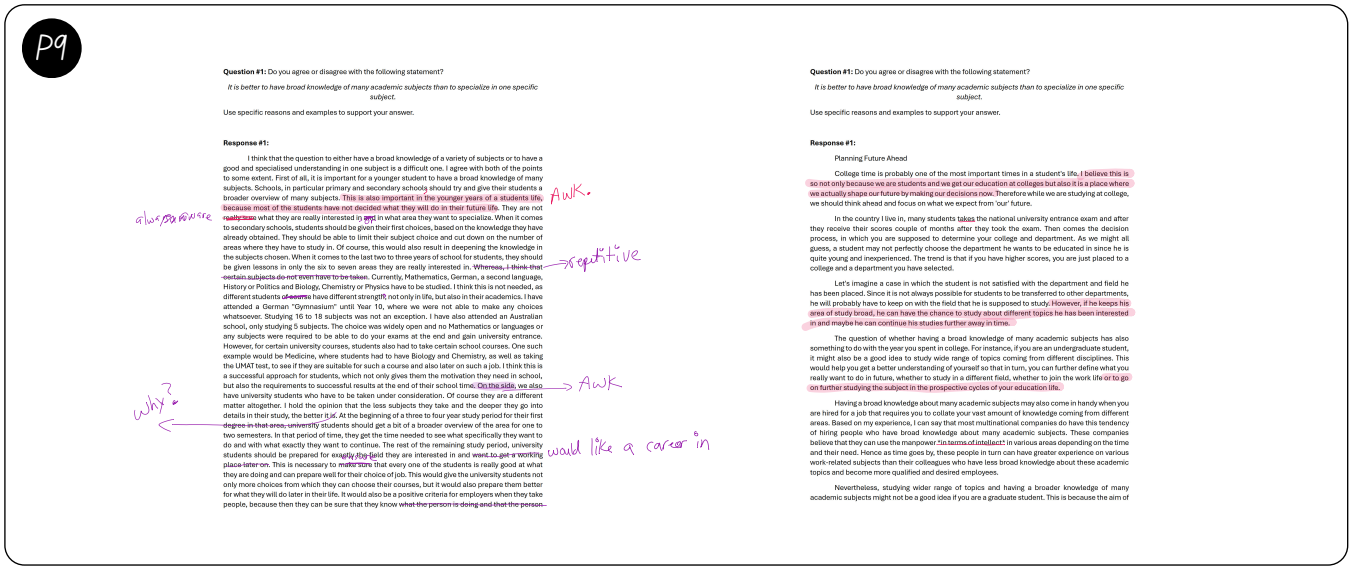


Figure 9: Screenshots of the first page for P9. The left side is with the baseline, and the right side is with AnnotateGPT. It demonstrates different annotation approaches, with the baseline annotations focusing on identifying issues such as unclear phrasing and negative tone using textual feedback, while in the AnnotateGPT condition, P9 only used highlights with no textual feedback.

logical flow). Participants have used a combination of these types of annotations to give feedback on the English tests. In total, 473 and 159 annotations were manually made for the BASELINE and ANNOTATEGPT, respectively.

5.4 Annotation Behaviours

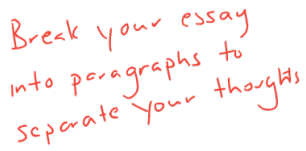
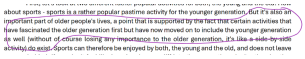
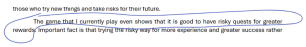
Participants were also asked about their annotation behaviours, as it was observed that telegraphic annotations were used more frequently than explicit annotations for ANNOTATEGPT. Six participants (P1, P4, P5, P9, P10 and P12) exhibited drastic differences in annotation behaviours between the two techniques (see Figure 9). These participants produced significantly more telegraphic annotations with AnnotateGPT compared to the baseline. In these cases, they provided little explicit textual feedback themselves, instead relying on the system to generate fuller comments. For example, one participant explained that they would *"let [AnnotateGPT] come up with the comments"'* [P4], while another noted there was *"no point writing it if the AI would come up with it itself"* [P9].

Several noted that the assistant could *"classify the purpose of my annotation"* [P7], so their own input was reduced to shorthand, quick marks, because they no longer felt the *"need to add [explicit] comments"* [P5]. Some participants described needing to *"make sure the assistant is picking up the right things"* [P8], while others welcomed that the tool could *"scan the whole document"* [P9] or even *"look at all for me, so I could focus on structure"* [P10]. One participant contrasted this with the BASELINE, saying they felt *"more responsible"* [P1] when annotating alone. Another noted that they often only highlighted text because they were *"seeing if it would pick up on it"* [P12] and liked *"how it can detect highlights [...] If I select the highlight, it will recognize what I highlighted and give me feedback"* [P2]. Even one acknowledged they sometimes found themselves *"missing the regular [textual] feedback"* [P10] as their attention shifted away from writing explanatory comments and towards curating AI suggestions. Overall, although annotations became increasingly telegraphic, AnnotateGPT was generally able to infer their intended purposes, with 63% being correctly inferred (see Table 2).

5.5 Misclassifications

There were instances where AnnotateGPT failed to accurately infer the purpose of the annotations. Out of 65 inferences, 24 were manually typed by the participants. All of the failed cases involved telegraphic annotations, causing the LLM to produce vague and general descriptions of the annotations. This guided the LLM to make very broad guesses, which would not suffice to pinpoint the purpose (see Table 2). Further analysis shows that this pattern reflects the annotation behaviour discussed earlier: participants often under-specified their intent by providing no textual feedback. In these cases, AnnotateGPT lacked sufficient semantic grounding to reliably infer the annotation purpose, leading to misclassification in 24 of 48 telegraphic annotations (50%). In contrast, annotations with explicit textual feedback were consistently classified correctly across all 17 explicit cases (100%). Overall, the system inferred 41 of 65 purposes correctly (63%), underscoring the challenge of interpreting highly telegraphic input. From a user experience perspective, participants did not report these misclassifications as disruptive. When the inferred purpose did not align with their intent, participants typically typed the correct purpose after reviewing the system's suggestion, treating the inference as an editable starting point rather than a final decision.

Table 2: Examples of purpose inference along with its annotation description. Green text indicates AnnotateGPT correctly inferred the purpose, while red text indicates participants manually typed the purpose.

Type	Annotation	Description	Listed Purposes
Telegraphic	~~vast amount~~ of knowledge ~~coming from different~~	The annotation shows a crossing out of the phrase 'coming from different', indicating a need for clarity and precision in language use.	1. Promoting Clear Language 2. Fostering Coherence in Writing 3. Promoting Clear Language (word-specific) 4. Fostering Coherence in Writing (word-specific)
Explicit		The annotation is a handwritten note in red pen stating, 'Break your essay into paragraphs to separate your thoughts.' This instruction emphasizes the importance of organizing the essay clearly.	1. Improving Essay Structure 2. Constructive Criticism 3. Emphasizing Clarity 4. Fostering Skill Development
Telegraphic		The annotation consists of a circle around the phrase [...] This indicates that the user is focusing on this passage for scrutiny during the grading process.	1. Encouraging Critical Analysis 2. Highlighting Key Content for Revision 3. Promoting Clarity in Expression 4. Identifying Areas for Conceptual Clarity 5. Run On Sentence
Telegraphic		The annotation consists of a line circling the phrase [...] The use of a circle emphasizes significance, indicating it's noteworthy in the context of the English test.	1. Emphasizing Life Lessons 2. Focus on Clarity 3. Inspiring Courage 4. Critical Analysis 5. Not Connected to the Answer

5.6 Density of Annotations

The inter- and intra-density of annotations significantly differ between TECHNIQUE and QUESTIONS. ART ANOVA revealed a significant main effect of TECHNIQUE ($F_{1,70} = 34.50$, $p < 0.0001$, $\eta^2_G = 0.30$), with an average of 44 strokes (95% CI : [30, 61]) and 13 strokes (95% CI : [9, 17]) per annotation for BASELINE and ANNOTATEGPT, respectively (see Figure 10a). This suggests that telegraphic annotations were used significantly more frequently for ANNOTATEGPT.

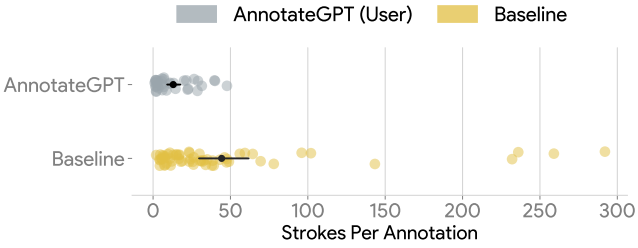
Regarding the number of annotations per question, ART ANOVA revealed significant main effects of TECHNIQUE ($F_{1,70} = 16.80$, $p < 0.001$, $\eta^2_G = 0.22$) and QUESTIONS ($F_{7,70} = 25.18$, $p < 0.001$, $\eta^2_G = 0.35$), where participants significantly created fewer annotations for ANNOTATEGPT (M = 4, 95% CI : [3, 5]) than the baseline (M = 7, 95% CI : [6, 9]) (see Figure 10b). In terms of QUESTIONS, the post-hoc test only reveals significant effects ($p < 0.05$) for question 1 from 5 and 7. However, the average number of annotations decreased after question 1. This suggests that only some participants experienced fatigue effects after the first question. The same trend is observed for ANNOTATEGPT, but this is due to the first question being the primary interaction space of the assistant, where they introduced what they are looking for and curated AI suggestions. Subsequent questions serve as follow-ups, requiring fewer new annotations.

5.7 Annotating Duration

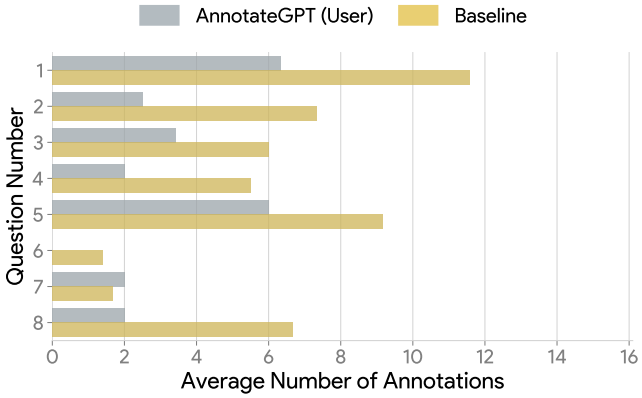
The annotating duration for each question is the time elapsed from the first to the last stroke being drawn on said question. ART ANOVA revealed a significant main effects QUESTIONS ($F_{7,70} = 5.14$, $p < 0.0001$, $\eta^2_G = 0.31$). However, only question 1 differed significantly ($p < 0.05$) from questions 2, 4, and 7. When considering the number of annotations, participants spent the most time on the first question, creating more annotations, further suggesting that fatigue effects were present. Additionally, participants spent more time on the first question for ANNOTATEGPT than the BASELINE due to the diverse workflows described earlier, where participants verify suggestions made by AnnotateGPT and then continue to annotate the first question.

5.8 Questionnaires

Participants also rated AnnotateGPT's features (Figure 12) and their overall experience in annotating the tests (Figure 13) on a 5-point Likert scale. Participants agreed that the automated annotations (MDN = 4.0, IQR = 0.5) and feedback were very helpful (MDN = 4.0, IQR = 0.25) and easy to understand (MDN = 4.0, IQR = 1.25). Participants also agreed that AnnotateGPT was able to provide helpful guesses on the purpose of their annotations (MDN = 4.0, IQR = 0.0), despite their annotations being highly telegraphic. However, participants found



(a) Strokes per annotation for each technique



(b) Average number of annotations per question

Figure 10: Plots representing the (a) intra-density: strokes per annotation for each technique, showing that participants used more explicit annotations in the baseline condition. (b) Inter-density: average number of annotations per question, indicating that participants focused more heavily on the first question. Only user-authored annotations are included.

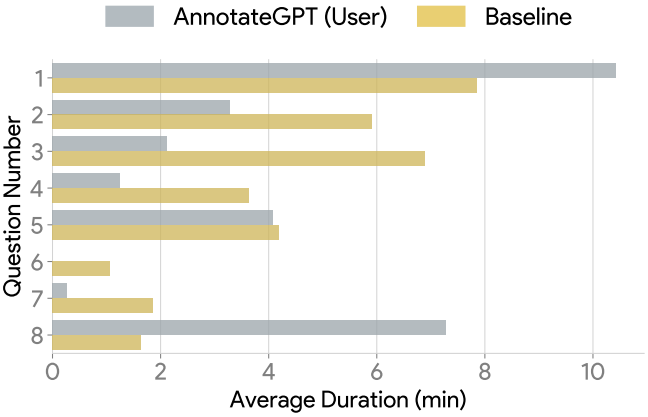


Figure 11: Annotating duration for AnnotateGPT and baseline for each question. Only user-authored annotations are included. It shows participants spent the most time on the first question.

that AnnotateGPT was unable to find all the issues (MDN = 3.0, IQR = 1.0). These findings suggest that AnnotateGPT can accurately infer the purpose of its annotations and provide meaningful feedback to the identified sentences. However, there were some inconsistencies in targeting all issues based on the ratings.

Regarding their overall annotating experience, there were no statistical differences between TECHNIQUE. Participants found grading the test (ANNOTATEGPT: MDN = 4.5, IQR = 2.25; BASELINE: MDN = 4.0, IQR = 1.0) and providing feedback to be easy (ANNOTATEGPT: MDN = 4.5, IQR = 2.0; BASELINE: MDN = 4.0, IQR = 1.0). They also reported comparable confidence in the final results of their annotations (ANNOTATEGPT: MDN = 3.0, IQR = 2.0; BASELINE: MDN = 3.0, IQR = 2.0) and in identifying all issues in the test (ANNOTATEGPT: MDN = 2.5, IQR = 1.25; BASELINE: MDN = 2.0, IQR = 2.0). Finally, both techniques were rated similarly in terms of ease of navigation (ANNOTATEGPT: MDN = 4.5, IQR = 1.25; BASELINE: MDN = 5.0, IQR = 1.0), indicating overall similarity in user experience across conditions.

5.9 Feedback Ratings

Participants, on average, made 10 (95% CI: [7, 12]) user annotations when collaborating with AnnotateGPT. Participants also verified each AnnotateGPT's annotations. On average, there were 47 (95% CI: [31, 64]) *accepted* annotations, 19 (95% CI: [11, 28]) *rejected* annotations, 8 (95% CI: [5, 12]) *helpful* annotations. ANNOTATEGPT significantly ($W = 0.0$, $p < 0.001$, $r = 0.88$) made more annotations (see Figure 14).

5.10 Feedback Quality

Participants generally perceived AnnotateGPT as providing broader and more efficient support during the annotation process compared to the baseline. Participants praised the assistant's ability to scan more widely and surface issues they would have missed: one observed it could *"look at a bigger scope than me"* [P1], another said it caught *"other points that I may have missed"* [P12] and *"made my life easier to [go] quicker"* [P9] when they had many students to grade, and *"graded more in total using [AnnotateGPT] and was able to find things a lot faster"* [P3]. This suggest that AnnotateGPT had increased coverage and throughput, especially under time pressure.

A common theme in the difference between each technique's feedback is that the baseline only states what the problem is (e.g. "awk" or "reword"), while AnnotateGPT explicitly states the reason (see Table 3). Several participants acknowledged AnnotateGPT's strengths in content-related feedback. For example, one participant said *"[AnnotateGPT] gave better feedback than I would write"* [P4] and *"knows the correct terms to use"* [P11], while another explained they preferred the assistant when *"looking more at the content of answers"* [P5]. Several teachers appreciated that AnnotateGPT *"would help me come up with more ideas to edit English work"* [P9] and make feedback writing faster.

However, this expanded capability sometimes introduced new concerns about over-reliance. For example, one participant noted, *"I*

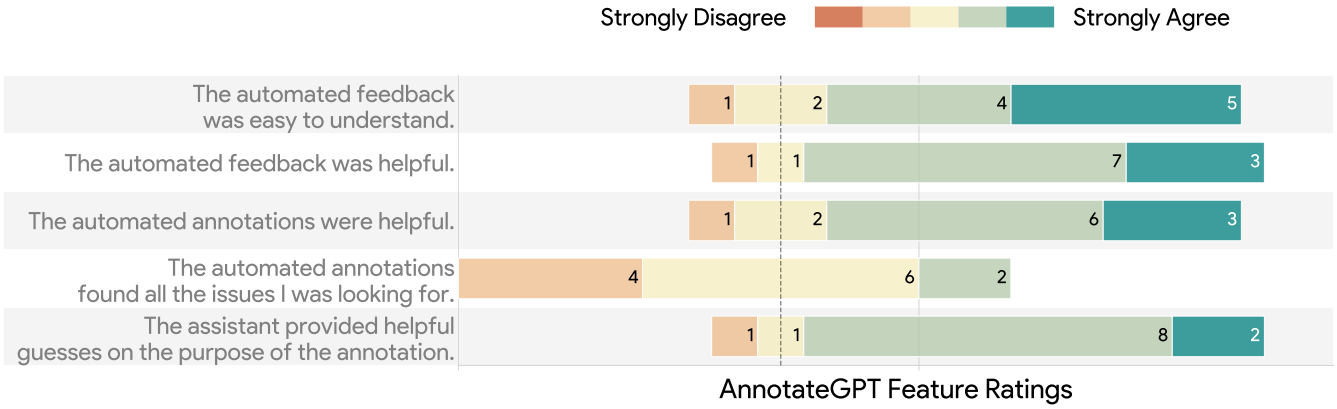


Figure 12: Questionnaire responses for AnnotateGPT's features on a 5-point Likert scale. Participants agreed that AnnotateGPT accurately inferred the purpose of their annotations and provided helpful feedback. However, it was unable to identify all issues for a given purpose.

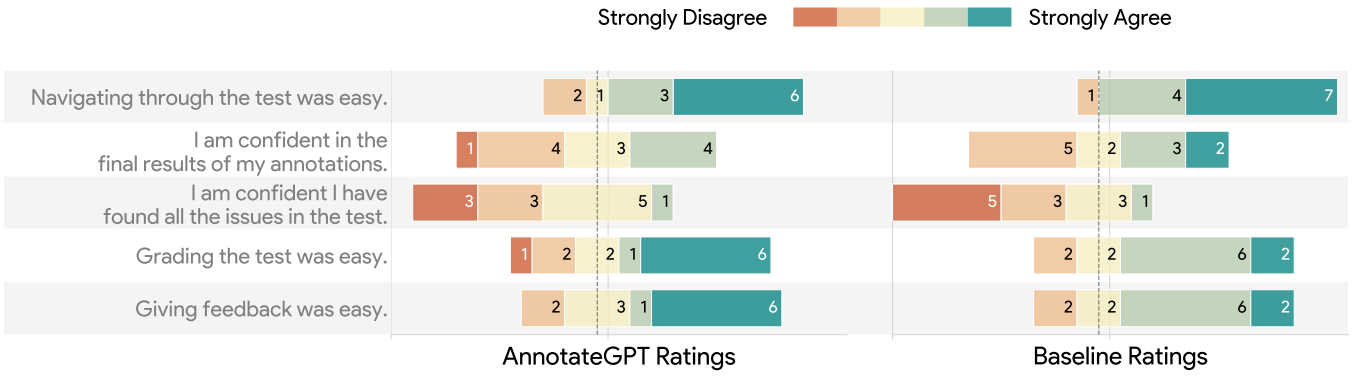


Figure 13: Questionnaire responses for the evaluation experience on a 5-point Likert scale. Participants found it easy to grade and provide feedback on the essay, but they are not confident that they found all the issues for both techniques.

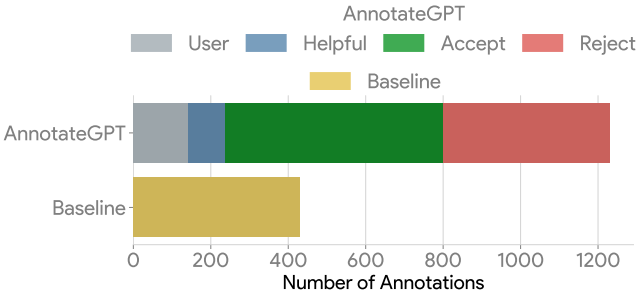


Figure 14: Number of annotations between AnnotateGPT and baseline, along with human-rated annotations for AnnotateGPT.

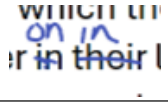
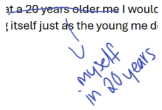
liked the suggestions it gave for [AnnotateGPT], I didn't find it always consistent" [P6] in finding the issues, while another reflected that they were letting *"the AI would write for me in a way"* [P5]. One noted they were *"relying on AnnotateGPT to read it for me and find all those errors"* [P5], but *"it wasn't perfect, there were a lot of grammar things it didn't pick up"* [P1] or occasionally missing context (e.g., *"it gave me a concrete example, but there was an example further down the paragraph"* [P7]).

Participants generally found AnnotateGPT beneficial for creating faster and more in-depth feedback, particularly in complex or time-sensitive situations. However, they also emphasized the importance of balancing it with manual review. As one participant proposed, a better workflow might be to *"go through it and mark it first, and then use the assistance as the secondary tool"* [P5], which followed the workflows of P3 and P6. AnnotateGPT was regarded as a valuable support tool for surfacing major issues and enhancing the feedback process (*"I would definitely use it for humanities and arts"* [P10] and *"this would be helpful [...] especially for beginner writers"* [P12]), especially when used in conjunction with manual annotation.

5.11 Summary

A user study ($N = 12$) was conducted to evaluate AnnotateGPT, a system that leverages LLMs to generate feedback based on user annotations during the evaluation of English essays. Participants engaged with AnnotateGPT and a baseline annotation tool to evaluate

Table 3: Example feedback from the baseline and AnnotateGPT from each category, and has been accepted by the participants.

Category	Example Annotation	Purpose	Example AnnotateGPT Feedback
Grammar		Grammar and Punctuation Issues	Grammar issue: Change "brought him in" to "brought him into."
Vocabulary		Vocabulary Enhancement	Revise "having a blast" to a formal or descriptive term that suits the discussion's tone.
Sentence Structure		Promoting Clarity in Expression	Sentence is overly complex and verbose; divide into smaller sentences for clarity and focus.
Organization & Coherence		Importance of Transitional Phrases	Use preceding transitions to connect the personal example to the broader argument smoothly. Example: "This pursuit often results in...".
Task Achievement		Recognition of Strength	Emphasizes the emotional aspect of guided tours effectively. Specify how guides enable this emotional connection.

essays, revealing a tendency to rely on the model for interpreting short or unclear annotations, but AnnotateGPT often accurately inferred the purpose of the annotations. Participants acknowledged the high quality of feedback but noted some inconsistencies in coverage (i.e. grammar). These insights indicate that LLM-based systems can significantly enhance human annotation by providing context-sensitive suggestions, particularly when user input is explicit and intentional.

6 Discussion

Study findings suggest several design directions for LLM-integrated annotation tools. These include using LLMs to augment feedback workflows, support educational equity for diverse educators, and rethinking annotation as a rich interaction paradigm for guiding AI output. Together, these insights suggest that users can collaborate with LLMs by using annotations, while also raising important considerations around balancing agency and over-reliance.

6.1 Balancing Agency and Over-Reliance

Building on prior work emphasizing integrated, user-directed AI support, our work similarly designs annotation workflows that preserve user agency by embedding AI assistance within the workspace while maintaining manual control [25]. Our analysis revealed substantial variability in participants' annotation workflows, ranging from single-pass approaches (ANNOTATE–INTERPRET and GENERATE–VERIFY) to batched workflows (ANNOTATE-K-INTERPRET-K and GENERATE-K-VERIFY). While these strategies produced comparable feedback quality, batched workflows appeared to promote more deliberate engagement with the text (P3 and P6). Such sequencing may help sustain user agency by providing structured moments for reflection and re-engagement, rather than encouraging continuous delegation of interpretive tasks to the model. This observation aligns with human–AI interaction principles that emphasize controllability and reliance, as well as human-centred AI that augments rather than replaces human cognition [3, 4].

Future work could extend this insight by incorporating lightweight behavioural metrics to promote more engaged workflows. In our study, annotation density and duration dropped significantly after the first question, suggesting changes in attention and engagement over time. If such metrics fall below typical thresholds, the system could prompt users to reflect or annotate more thoughtfully before invoking AI feedback, leading to more explicit annotations and improving the LLM's ability to infer the annotation's purpose. Prior work shows that prompting users to elaborate before relying on AI can strengthen psychological ownership [26]. Such approaches could help preserve the teacher's voice and identity within the annotation process, balancing agency and over-reliance, a central tension in effective human–AI collaboration.

Equally important, systems must be designed to respect cultural variation and pedagogical diversity, supporting teachers' voices rather than overshadowing them with dominant perspectives. With AnnotateGPT, we propose one promising approach is sequencing: participants suggested marking a document manually first, then allowing the AI to propose additional feedback, a practice echoed by prior work showing that writing without AI followed by revision with AI fosters ownership and more strategic integration [28].

6.2 LLMs as Cognitive Augmentation in Feedback Workflows

Our study shows that AnnotateGPT acted as a cognitive augmentation, expanding brief or telegraphic annotations into fully articulated feedback. Participants often relied on AnnotateGPT to infer intent from minimal to no textual cues, and in most cases, found the model's interpretations aligned with their intentions. This aligns with previous works on LLM-integrated systems [12, 14, 16, 24, 47, 50, 59], which support expression rather than idea generation, allowing users to concentrate more on the overall content and less on specifics. For example, Code Shaping [59] enables programmers to create and edit code by making annotations

and sketches, shifting the programming process from a syntax-focused method to one that prioritizes code structure and flow.

In our study, AnnotateGPT played a similar role for teachers. By externalizing the work of phrasing and formatting, it allowed educators to focus cognitive effort on identifying issues rather than constructing detailed responses. For example, participants marked a telegraphic annotation on a student essay, and AnnotateGPT returned a comment explaining the problems across paragraphs. The users confirmed it was what they intended by verifying the annotations, but would not have typed it out themselves. This demonstrates that LLMs are not just automation tools, but cognitive collaborators that can empower educators to focus on tasks that require critical expertise, provided the system maintains alignment with pedagogical intent.

6.3 Support Equity in Education

AI-driven systems such as AnnotateGPT may help address inequities in education, particularly where teachers have limited time or training to provide detailed, individualized feedback. Our findings suggest that AnnotateGPT can expand shorthand annotations into fuller comments, potentially supporting novice teachers and teaching assistants in giving feedback that is clearer and more consistent. In our study, for example, participants focused on grammar and sentence-level issues more often in the baseline condition, while AnnotateGPT was used to address organization and coherence, broadening feedback toward higher-level concerns. This shift suggests that annotation-driven AI can help educators move beyond surface corrections toward more substantive guidance.

By grounding AI generation in teacher-provided annotations, AnnotateGPT aims to preserve educators' voices and styles, reducing the risk of homogenized or overly standardized feedback. Still, our study did not examine student outcomes or long-term pedagogical effects. Future work should investigate how AI-augmented feedback affects student learning, trust, and motivation, and how systems can adapt to diverse teaching practices rather than impose uniform standards.

More broadly, prior work shows that LLMs can "level the playing field" by providing access to domain-specific knowledge or communication skills users may lack [33, 52]. In educational contexts, AnnotateGPT extends this benefit by enabling novice teachers to deliver feedback that is timely, accurate, and aligned with curriculum objectives, even at scale. For instance, teachers could guide AnnotateGPT with annotations aligned with curriculum objectives, ensuring generated feedback remains tailored to course goals. Curriculum materials could also be directly provided to the model to further tailor feedback generation.

6.4 Annotation as an Interaction Paradigm

Our study highlights annotation as a central component, not just as a tool, but as an interaction paradigm for engaging with AI systems. While annotations have traditionally been private or instructor-focused, combining them with LLMs like AnnotateGPT transforms them into expressive inputs that shape system behaviour and feedback. This reflects a broader trend toward low-friction, in-context interfaces that leverage users' micro-actions (e.g., highlights, marginal notes, shorthand feedback) as signals to AI systems [3].

In our study, participants often employed the system to elaborate brief annotations into comprehensive suggestions, effectively making annotations a shared language between humans and AI. This aligns with recent work on intent-aware AI [35] and prior research that situates annotation as a space of interpretation and sense-making [36, 57]. Rather than requiring deliberate commands or structured forms, AnnotateGPT supports opportunistic interaction: users write freely, and the AI infers intent, reducing overhead and effort. Although our study focused on essay feedback, several aspects of AnnotateGPT's design, such as purpose inference and stroke clustering, are domain-independent and can be extended to other domains. To adapt the system to new domains would primarily require re-specifying domain taxonomies and the annotation generation prompt to match the context.

Future systems can extend this paradigm with multimodal annotations or real-time previews of AI inferences to enhance transparency and accuracy. Ultimately, framing annotation as an interaction paradigm opens a design space for collaborative, intent-aware interfaces that augment rather than replace human input. We next illustrate how AnnotateGPT could be applied in everyday teaching contexts and beyond.

7 Educational Scenarios: Applying AnnotateGPT in Practice

We illustrate the implications of AnnotateGPT in practice through two educational scenarios. These examples highlight the potential for teachers to integrate the system into their existing workflows, demonstrating both efficiency gains and shifts in feedback practices.

7.1 Lightweight Annotation for High-Volume Feedback

Justina teaches English at a public high school, where her two classes total nearly 50 students. The current assignment, a persuasive essay on environmental responsibility, has generated a stack of essays that need review. With limited time between classes and meetings, she needs to provide meaningful feedback while staying consistent.

She turns to AnnotateGPT, rather than asking ChatGPT to provide the grades directly, because she wants to retain control and agency over the grading process. Working quickly, Justina highlights phrases and leaves brief, often fragmentary annotations, such as "vague," "support?," or "good rhetoric." Rather than forcing her to slow down and draft full comments, AnnotateGPT treats these annotations as intent-rich cues and generates expanded feedback suggestions on the fly.

Justina reviews each suggestion in line. She accepts many as written, editing some to match her preferred tone for students by commenting, and rejects others if redundant. Because AnnotateGPT operates within her natural annotation process, she does not need to switch modes or perform additional steps. Her shorthand becomes a trigger for expanding feedback, enabling a smooth flow between recognition and response.

Over the weekend, she is able to provide detailed feedback on each submission more quickly, while maintaining consistency and

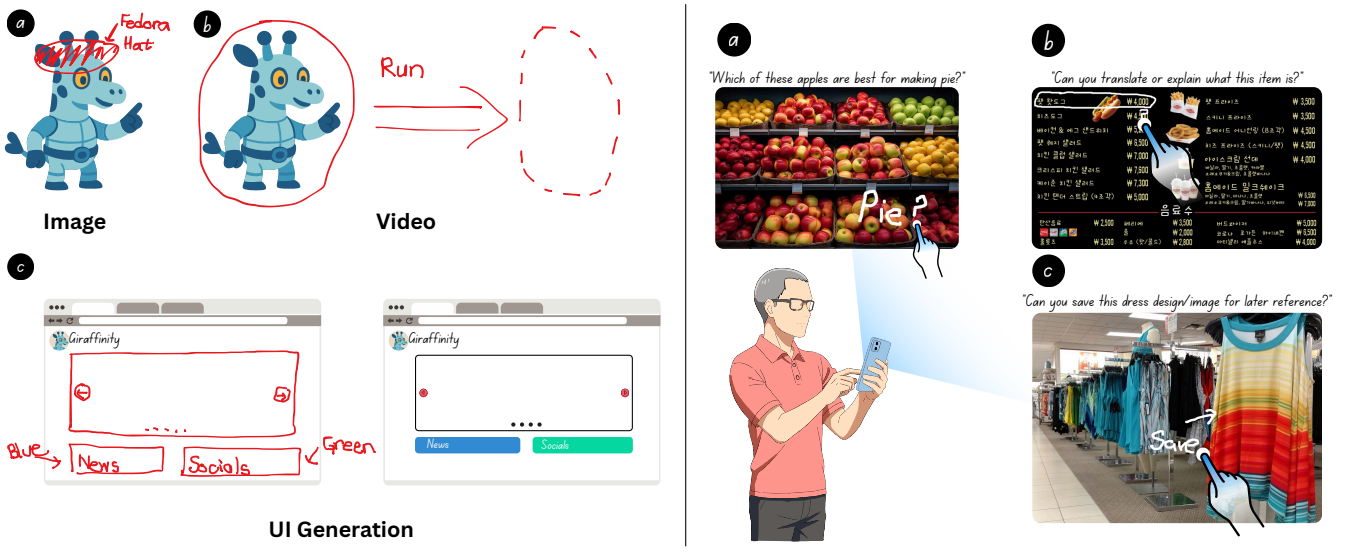


Figure 15: Sample use cases of using annotations: Left: editing and refining generative content, including (a) refining image generation, (b) making action flows for video generation, and (c) creating and editing user interfaces. Right: interacting with a phone using on-screen annotations and form complex queries, such as (a) which apples are best for apple pies, (b) translation, and (c) saving an item for later reference. All sample queries are sourced from ChatGPT and paired with the corresponding image.

structure across a large number of essays. In this scenario, AnnotateGPT serves not as a substitute but as a cognitive amplifier, supporting efficiency while preserving her pedagogical intent.

7.2 Leveraging Annotation History Across Subjects

Ash is a teaching assistant responsible for evaluating writing assignments in two undergraduate courses: an introductory academic writing class and a course on the ethics of computer science. While both involve writing, the standards and priorities differ. In academic writing, he emphasizes structural clarity, effective transitions, and logical organization. In ethics, the focus shifts to argumentative rigour, ethical frameworks, and engagement with real-world case studies.

Ash uses AnnotateGPT in both contexts. As he works through assignments each week, AnnotateGPT builds an annotation history, noting patterns in his annotations, identifying the types of annotations used for each context, and linking those patterns to the course context and content.

Now, evaluating an ethics essay, Ash highlights a paragraph with a vague appeal to fairness in AI. He writes "needs clarity." Rather than suggesting feedback on vague writing mechanics, as might be provided in a writing course, AnnotateGPT draws on prior annotations related to the ethics course and infers that "clarity" in this context referred not to sentence-level writing but to conceptual framing. In contrast, the same input in the writing class would have triggered feedback about transitions or sentence rephrasing.

By retaining past annotations, AnnotateGPT enables Ash to teach ethics while preserving the subtleties of his teaching style. The tool adapts to the context of each class, enhancing the quality of feedback, aligning with learning objectives, and providing timely feedback.

8 Beyond Education: Envisioned Applications of Annotation as Interaction

While our study focused on educational feedback, annotations can serve more broadly as a lightweight, intentional interaction paradigm for AI systems. Their contextual and in-situ nature makes them valuable wherever explicit input is costly or disruptive. We highlight two speculative directions. Future work should evaluate their practicality, user acceptance, and integration with domain-specific workflows.

8.1 Editing Generative Content

Generative AI tools often rely on text prompts, but users struggle to craft effective prompts for refining outputs [43]. Annotations could instead define spatial constraints or indicate reference elements directly within an image, video, or interface design. For example, a scribble might specify where an object should appear in an image or mark a sequence for a generated video. Compared to repeated prompting, annotations offer a low-friction alternative for guiding iteration.

8.2 Annotating the World

While LLMs can now process images and live video feeds, users are often still limited to voice or text for interaction. Voice commands are disruptive in public spaces, while typing out descriptions of a live scene is slow and often inaccurate. On-screen annotations provide a discreet and precise alternative. For example, a user can

simply mark a specific area on their camera feed, such as writing "Pie?" next to a display of apples, to ask "Which of these apples are best for making pie?" (see Figure 15). This enables users to seamlessly direct the AI's attention and feedback in the real world without speech or text.

9 Limitations & Future Work

Our study has several limitations. First, the participant pool was small ($N = 12$) and consisted entirely of pre-service teachers at a single institution. While this limits generalizability, our goal was to explore interaction patterns in depth rather than make broad claims about pedagogy. Future work should test AnnotateGPT with larger and more diverse groups of educators and students.

Second, our evaluation measured teacher experiences rather than student outcomes. As such, we cannot claim effects on learning, trust, or motivation. Understanding these downstream impacts will be essential for future research.

Third, although AnnotateGPT grounds generation in teacher annotations, errors and misclassifications still occurred, and our study did not measure their influence on teaching practice or student reception. More systematic evaluation of failure cases and mitigation strategies is needed.

Fourth, AI support may risk overshadowing the teacher's identity; transparent design choices, such as disclosing AI involvement and preserving the educator's voice, are crucial. Future work could also test dynamic generation scopes to align with varied workflows. For example, restricting the scope to one question at a time or across multiple students' answers may encourage more annotations, while detecting when an annotation is complete before activating assistance may promote ownership [28].

Finally, although AnnotateGPT was designed for classroom settings, our study took place in a controlled environment with limited time. Longer-term adaptation of workflows, such as trust building, habit formation, efficiency, and the risks of AI over-reliance, should be investigated through longitudinal deployments spanning weeks or semesters.

Future work on system design could explore multimodal interactions to enhance efficiency, such as speech recognition (i.e., DrawTalking [46]), which would allow teachers to specify intent without relying solely on pen gestures. Beyond explicit multimodal input, background inference could be leveraged to precompute initial guesses, further reducing system friction while keeping users in the loop. More broadly speaking, AnnotateGPT emphasizes user-initiated interaction to maintain agency and control. Although proactive AI suggestions can pinpoint areas for assistance, they may also limit individual agency and ownership [25, 26]. Further investigation of hybrid approaches that combine pre-computing inferences and proactive suggestions with user-driven annotation feedback could clarify how different degrees of automation affect control, trust, and perceived ownership in future human–AI annotation systems.

Although stroke clustering was effective in our study due to the high use of telegraphic annotations, these annotations are typically sparse and visually consistent, which simplifies the clustering process. While we did not observe failures of our heuristic approach even for annotations accompanied by explicit text, more advanced methods may perform better as users annotate more naturally or extensively. Future work should investigate advanced clustering methods, like machine learning models, to enhance robustness and ensure accurate AI interpretation in diverse scenarios.

While our study focused on education, future work could explore how annotation-driven AI applies in other domains, such as peer review, collaborative writing, or note-taking. These contexts may benefit from its emphasis on precision and interpretability, clarifying how annotation-based interaction can augment human judgment across diverse settings.

10 Conclusion

This work explored how large language models can augment human annotation practices, transforming handwritten feedback from a static artifact into a collaborative interaction. Through our study of AnnotateGPT, we showed that even brief and telegraphic annotations can serve as rich signals for AI, enabling the generation of fuller, more consistent feedback while keeping educators in control. These findings highlight annotation not only as a support for reading and review, but as a broader interaction paradigm for guiding AI behaviour. Looking forward, the challenge is to design systems that preserve teacher intent, respect pedagogical diversity, and extend this paradigm into new domains where lightweight, in-situ cues can shape meaningful human–AI collaboration.

Acknowledgments

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). We also thank our colleagues in the Faculty of Education for their participation and support in this study.

References

[1] Adobe. 2024. *Adobe Acrobat.* https://acrobat.adobe.com/

[2] Natalie M. Agius and Ann Wilkinson. 2014. Students' and teachers' views of written feedback at undergraduate level: A literature review. *Nurse Education Today* 34, 4 (2014), 552–559. doi:10.1016/j.nedt.2013.07.005

[3] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. 2019. Guidelines for human-AI interaction. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems.* ACM, New York, NY, USA, 1––13. doi:10.1145/3290605.3300233

[4] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. 2019. Guidelines for human-AI interaction. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems.* ACM, New York, NY, USA, 1–13. doi:10.1145/3290605.3300233

[5] Richard Bailey and Mark Garner and. 2010. Is the feedback in higher education assessment worth the paper it is written on? Teachers' reflections on their practices. *Teaching in Higher Education* 15, 2 (2010), 187–198. arXiv:https://doi.org/10.1080/13562511003620019 doi:10.1080/13562511003620019

[6] Elaine Ball, Helen Franks, Jane Jenkins, Maureen McGrath, and Jackie Leigh. 2009. Annotation is a valuable tool to enhance learning and assessment in student essays. *Nurse Education Today* 29, 3 (2009), 284–291. doi:10.1016/j.nedt.2008.10.005 Special Issue: Selected papers from the 2nd Int. Nurse Education Conf. Research and Innovation in Int. Nurse Education 9-11 June, 2008, Dublin, Ireland.

[7] Karim Benharrak, Tim Zindulka, Florian Lehmann, Hendrik Heuer, and Daniel Buschek. 2024. Writer-defined AI personas for on-demand feedback generation. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems.* ACM, New York, NY, USA, Article 1049, 18 pages. doi:10.1145/3613904.3642406

[8] Daniel Blanchard, Joel Tetreault, Derrick Higgins, Aoife Cahill, and Martin Chodorow. 2013. TOEFL11: A corpus of non-native English. *ETS Research Report Series* 2013, 2 (2013), i–15. doi:10.1002/j.2333-8504.2013.tb02331.x

[9] Lyn Brodie and Birgit Loch. 2009. Annotations with a tablet PC or typed feedback: Does it make a difference? doi:10.25916/sut.26288488.v1

[10] John Brooke. 1995. SUS: A quick and dirty usability scale. *Usability Eval. Ind.* 189 (11 1995).

[11] Patrick Chiu and Lynn Wilcox. 1998. A dynamic grouping technique for ink and audio notes. In *Proc. ACM Symp. on User Interface Software and Technology*. ACM, New York, NY, USA, 195—-202. doi:10.1145/288392.288605

[12] Jean-Peïc Chou, Alexa Fay Siu, Nedim Lipka, Ryan Rossi, Franck Dernoncourt, and Maneesh Agrawala. 2023. TaleStream: Supporting story ideation with trope knowledge. In *Proc. ACM Symp. on User Interface Software and Technology (UIST '23)*. ACM, New York, NY, USA, Article 52, 12 pages. doi:10.1145/3586183.3606807

[13] Andrew D. Cohen and Marilda C. Cavalcanti. 1990. *Feedback on compositions: Teacher and student verbal reports*. Cambridge University Press, Cambridge, United Kingdom, 155–177.

[14] Min Fan, Xinyue Cui, Jing Hao, Renxuan Ye, Wanqing Ma, Xin Tong, and Meng Li. 2024. StoryPrompt: Exploring the design space of an AI-empowered creative storytelling system for elementary children. In *Extended Abstracts of the CHI Conf. on Human Factors in Computing Systems*. ACM, New York, NY, USA, Article 303, 8 pages. doi:10.1145/3613905.3651118

[15] Manuel J Fonseca, César Pimentel, and Joaquim A Jorge. 2002. CALI: An online scribble recognizer for calligraphic interfaces. In *AAAI Spring Symp. on Sketch Understanding*. AAAI Press Menlo Park, AAAI Press, Menlo Park, CA, USA, 51–58.

[16] Yue Fu, Sami Foell, Xuhai Xu, and Alexis Hiniker. 2024. From text to self: Users' perception of AIMC tools on interpersonal communication and self. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, New York, NY, USA, Article 977, 17 pages. doi:10.1145/3613904.3641955

[17] Joaquín Gayoso-Cabada, Antonio Sarasa-Cabezuelo, and José-Luis Sierra-Rodríguez. 2019. A review of annotation classification tools in the educational domain. *Open Computer Science* 9, 1 (2019), 299–307. doi:10.1515/comp-2019-0021

[18] Gene Golovchinsky, Morgan N. Price, and Bill N. Schilit. 1999. From reading to retrieval: Freeform ink annotations as queries. In *Proc. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*. ACM, New York, NY, USA, 19––25. doi:10.1145/312624.312637

[19] Grammarly. 2024. *Grammarly*. https://grammarly.com/

[20] Feng Han, Yifei Cheng, Megan Strachan, and Xiaojuan Ma. 2021. Hybrid paper-digital interfaces: A systematic literature review. In *Proc. ACM Designing Interactive Systems Conf.* ACM, New York, NY, USA, 1087––1100. doi:10.1145/3461778.3462059

[21] S. G. Hart. 2006. Nasa-task load index (NASA-TLX); 20 years later. *Proc. of the Human Factors and Ergonomics Society Annual Meeting* 50 (2006), 904–908. https://api.semanticscholar.org/CorpusID:6292200

[22] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (task load index): Results of empirical and theoretical research. In *Human Mental Workload*, Peter A. Hancock and Najmedin Meshkati (Eds.). Advances in Psychology, Vol. 52. North-Holland, Amsterdam, Netherlands, 139–183. doi:10.1016/S0166-4115(08)62386-9

[23] Annika Hinze, Ralf Heese, Alexa Schlegel, and Markus Luczak-Rösch. 2012. User-defined semantic enrichment of full-text documents: experiences and lessons learned. In *Theory and Practice of Digital Libraries*, Panayiotis Zaphiris, George Buchanan, Edie Rasmussen, and Fernando Loizides (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 209–214.

[24] Md Naimul Hoque, Tasfia Mashiat, Bhavya Ghai, Cecilia D. Shelton, Fanny Chevalier, Kari Kraus, and Niklas Elmqvist. 2024. The HaLLMark effect: Supporting provenance and transparent use of large language models in writing with interactive visualization. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, New York, NY, USA, Article 1045, 15 pages. doi:10.1145/3613904.3641895

[25] Nikhita Joshi and Daniel Vogel. 2025. Designing and evaluating AI margin notes in document reader software. arXiv:2509.09840 [cs.HC] https://arxiv.org/abs/2509.09840

[26] Nikhita Joshi and Daniel Vogel. 2025. Writing with AI lowers psychological ownership, but longer prompts can help. In *Proc. of the ACM Conference on Conversational User Interfaces (CUI '25)*. ACM, New York, NY, USA, Article 72, 17 pages. doi:10.1145/3719160.3736608

[27] Levent Burak Kara, Leslie Gennari, and Thomas F. Stahovich. 2008. A sketch-based tool for analyzing vibratory mechanical systems. *J. of Mechanical Design* 130, 10 (09 2008), 101101. doi:10.1115/1.2965595

[28] Nataliya Kosmyna, Eugene Hauptmann, Ye Tong Yuan, Jessica Situ, Xian-Hao Liao, Ashly Vivian Beresnitzky, Iris Braunstein, and Pattie Maes. 2025. Your brain on ChatGPT: Accumulation of cognitive debt when using an AI assistant for essay writing task. arXiv:2506.08872 [cs.AI] https://arxiv.org/abs/2506.08872

[29] Philippe Laban, Jesse Vig, Marti Hearst, Caiming Xiong, and Chien-Sheng Wu. 2024. Beyond the Chat: Executable and verifiable text-editing with LLMs. In *Proc. ACM Symp. on User Interface Software and Technology*. ACM, New York, NY, USA, Article 20, 23 pages. doi:10.1145/3654777.3676419

[30] Vivian Lai, Chacha Chen, Alison Smith-Renner, Q. Vera Liao, and Chenhao Tan. 2023. Towards a science of human-AI decision making: An overview of design space in empirical human-subject studies. In *Proc. ACM Conf. on Fairness, Accountability, and Transparency*. ACM, New York, NY, USA, 1369—-1385. doi:10.1145/3593013.3594087

[31] J.A. Landay and B.A. Myers. 2001. Sketching interfaces: Toward more human interface design. *Computer* 34, 3 (2001), 56–64. doi:10.1109/2.910894

[32] Vladimir I. Levenshtein. 1965. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics. Doklady* 10 (1965), 707–710. https://api.semanticscholar.org/CorpusID:60827152

[33] Jiawei Li, Yizhe Yang, Yu Bai, Xiaofeng Zhou, Yinghao Li, Huashan Sun, Yuhang Liu, Xingpeng Si, Yuhao Ye, Yixiao Wu, Yiguan Lin, Bin Xu, Bowen Ren, Chong Feng, Yang Gao, and Heyan Huang. 2024. Fundamental capabilities of large language models and their applications in domain scenarios: A survey. In *Proc. Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 11116–11141. doi:10.18653/v1/2024.acl-long.599

[34] Zekun Li, Baolin Peng, Pengcheng He, Michel Galley, Jianfeng Gao, and Xifeng Yan. 2023. Guiding large language models via directional stimulus prompting. In *Proc. Int. Conf. on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA, Article 2735, 27 pages.

[35] Q. Vera Liao, Daniel Gruen, and Sarah Miller. 2020. Questioning the AI: Informing design practices for explainable AI user experiences. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1—-15. doi:10.1145/3313831.3376590

[36] Catherine C. Marshall. 1997. Annotation: From paper books to the digital library. In *Proc. ACM Int. Conf. on Digital Libraries*. ACM, New York, NY, USA, 131––140. doi:10.1145/263690.263806

[37] Catherine C. Marshall. 1998. Toward an ecology of hypertext annotation. In *Proc. ACM Conf. on Hypertext and Hypermedia: Links, Objects, Time and Space—Structure in Hypermedia Systems: Links, Objects, Time and Space—Structure in Hypermedia Systems*. ACM, New York, NY, USA, 40––49. doi:10.1145/276627.276632

[38] Hrim Mehta, Adam Bradley, Mark Hancock, and Christopher Collins. 2017. Metatation: Annotation as implicit interaction to bridge close and distant reading. *ACM Trans. Comput.-Hum. Interact.* 24, 5, Article 35 (Nov. 2017), 41 pages. doi:10.1145/3131609

[39] Meredith Ringel Morris, A.J. Bernheim Brush, and Brian R. Meyers. 2007. Reading revisited: Evaluating the usability of digital display surfaces for active reading tasks. In *IEEE Int. Workshop on Horizontal Interactive Human-Computer Systems*. IEEE, Piscataway, NJ, USA, 79–86. doi:10.1109/TABLETOP.2007.12

[40] Shankar Narayanaswamy. 1996. *Pen and speech recognition in the user interface for mobile multimedia terminals*. University of California, Berkeley, Berkeley, CA, USA.

[41] OpenAI. 2025. OpenAI Node API Library. https://platform.openai.com

[42] Ilia A. Ovsiannikov, Michael A. Arbib, and Thomas H. McNeill. 1999. Annotation technology. *Int. J. Hum.-Comput. Stud.* 50, 4 (April 1999), 329—-362. doi:10.1006/ijhc.1999.0247

[43] Hyerim Park, Malin Eiband, Andre Luckow, and Michael Sedlmair. 2025. Exploring visual prompts: Refining images with scribbles and annotations in generative AI image tools. In *Proc. of the Extended Abstracts of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, New York, NY, USA, Article 257, 10 pages. doi:10.1145/3706599.3719802

[44] Michelle G Paterno. 2002. Responding to student writing. *Kritika Kultura* 1, 2 (2002), 5.

[45] Ken Pfeuffer, Ken Hinckley, Michel Pahud, and Bill Buxton. 2017. Thumb + pen interaction on tablets. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, New York, NY, USA, 3254–3266. doi:10.1145/3025453.3025567

[46] Karl Toby Rosenberg, Rubaiat Habib Kazi, Li-Yi Wei, Haijun Xia, and Ken Perlin. 2024. DrawTalking: Building interactive worlds by sketching and speaking. In *Proc. of the ACM Symposium on User Interface Software and Technology (UIST '24)*. ACM, New York, NY, USA, Article 76, 25 pages. doi:10.1145/3654777.3676334

[47] Orit Shaer, Angelora Cooper, Osnat Mokryn, Andrew L Kun, and Hagit Ben Shoshan. 2024. AI-augmented brainwriting: Investigating the use of LLMs in group ideation. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, New York, NY, USA, Article 1050, 17 pages. doi:10.1145/3613904.3642414

[48] Jingyu Shi, Rahul Jain, Hyungjun Doh, Ryo Suzuki, and Karthik Ramani. 2024. An HCI-centric survey and taxonomy of human-generative-AI interactions. arXiv:2310.07127 [cs.HC] https://arxiv.org/abs/2310.07127

[49] Cvetka Sokolov. 2022. Challenges of written response to student writing: Praise, over-Commenting and appropriation. *AAA: Arbeiten aus Anglistik und Amerikanistik* 47, 1 (2022), pp. 125–152. https://www.jstor.org/stable/27204982

[50] Sangho Suh, Meng Chen, Bryan Min, Toby Jia-Jun Li, and Haijun Xia. 2024. Luminate: Structured generation and exploration of design space with large language models for human-AI co-creation. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, New York, NY, USA, Article 644, 26 pages. doi:10.1145/3613904.3642400

[51] Craig J. Sutherland, Andrew Luxton-Reilly, and Beryl Plimmer. 2016. Freeform digital ink annotations in electronic documents: A systematic mapping study. *Computers & Graphics* 55 (2016), 1–20. doi:10.1016/j.cag.2015.10.014

[52] Stephanie Valencia, Richard Cave, Krystal Kallarackal, Katie Seaver, Michael Terry, and Shaun K. Kane. 2023. "The less I type, the better": How AI language models can enhance or impede communication for AAC users. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, New York, NY, USA, Article 830, 14 pages. doi:10.1145/3544548.3581560

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. Int. Conf. on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA, 6000––6010.

[54] Vercel. 2025. Next.js – The React Framework for the Web. https://nextjs.org

[55] Qian Wan, Xin Feng, Yining Bei, Zhiqi Gao, and Zhicong Lu. 2024. Metamorpheus: Interactive, affective, and creative dream narration through metaphorical visual storytelling. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, New York, NY, USA, Article 166, 16 pages. doi:10.1145/3613904.3642410

[56] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, New York, NY, USA, 143–146. doi:10.1145/1978942.1978963

[57] Joanna Wolfe. 2002. Annotation technologies: A software and research review. *Computers and Composition* 19, 4 (2002), 471–497. doi:10.1016/S8755-4615(02)00144-5

[58] Haoran Wu, Wenxuan Wang, Yuxuan Wan, Wenxiang Jiao, and Michael Lyu. 2023. ChatGPT or Grammarly? Evaluating ChatGPT on grammatical error correction benchmark. arXiv:2303.13648 [cs.CL] https://arxiv.org/abs/2303.13648

[59] Ryan Yen, Jian Zhao, and Daniel Vogel. 2025. Code Shaping: Iterative code editing with free-form AI-interpreted sketching. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, New York, NY, USA, Article 872, 17 pages. doi:10.1145/3706598.3713822

[60] Gokul Yenduri, M. Ramalingam, G. Chemmalar Selvi, Y. Supriya, Gautam Srivastava, Praveen Kumar Reddy Maddikunta, G. Deepti Raj, Rutvij H. Jhaveri, B. Prabadevi, Weizheng Wang, Athanasios V. Vasilakos, and Thippa Reddy Gadekallu. 2024. GPT (Generative Pre-Trained Transformer)— A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. *IEEE Access* 12 (2024), 54608–54649. doi:10.1109/ACCESS.2024.3389497

[61] Zahra Zahedi and Subbarao Kambhampati. 2021. Human-AI symbiosis: A survey of current approaches. arXiv:2103.09990 [cs.AI] https://arxiv.org/abs/2103.09990

[62] Vivian Zamel. 1985. Responding to student writing. *TESOL Quarterly* 19, 1 (1985), 79–101. http://www.jstor.org/stable/3586773

A System Usability Scale

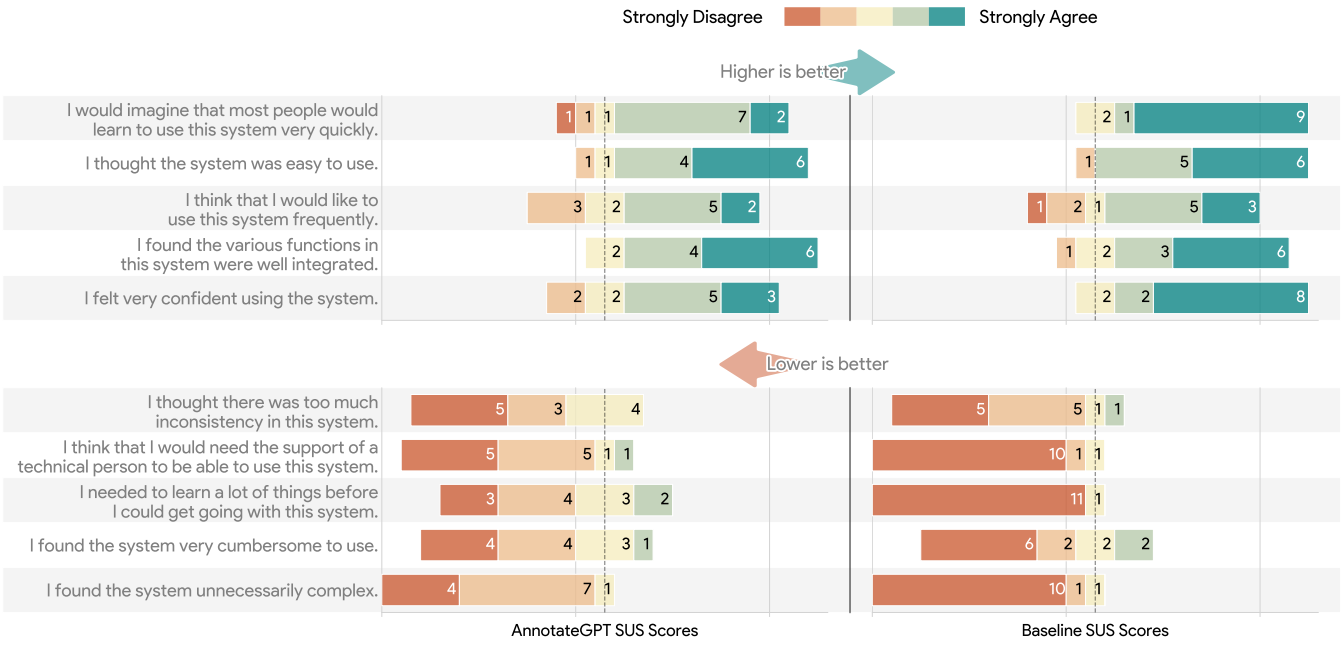


Figure 16: Plot representing the SUS scores across AnnotateGPT and baseline.

B NASA Task Load Index

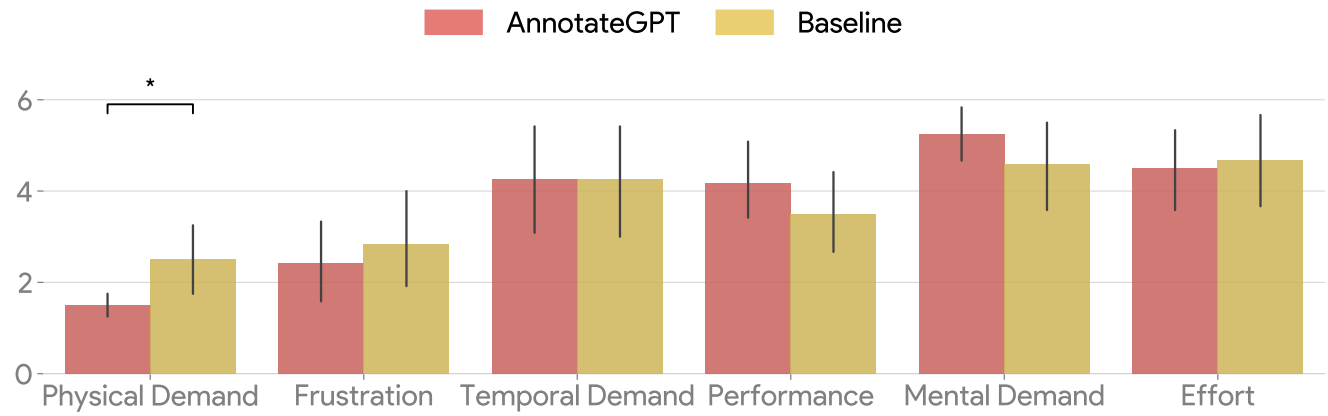


Figure 17: Plot representing the TLX scores across AnnotateGPT and baseline. Asterisk depict significant difference ($p < 0.05$).

C Annotation History

Table 4: The annotation history shown at the last activation when inferring the annotation purpose for each participant.

ID	Last Annotation History Entry
P1	The user has a pattern of marking significant terms and grammatical elements in their annotations, often using red to denote importance, as seen in previous annotations that emphasized key beliefs and grammatical elements like subject pronouns.
P2	The user has previously provided annotations aimed at improving clarity in student writing, recognizing strong arguments, and informing students about structuring their thoughts and engaging with their educational perspectives.
P3	User has a history of using underlining for emphasizing important phrases, red pen strokes for indicating questions, and comments that encourage clarity and conciseness in student writing.
P4	Past annotations have included corrections and highlights for clarity and precision, emphasizing the importance of strong argumentative language and precise wording in student writings.
P5	Previous annotations in the history involve circling significant phrases and marking corrections, emphasizing the importance of clarity and coherent sentence structures. The user often highlights crucial ideas or transitions, indicating a focus on constructive critique and feedback for student improvement.
P6	The user's annotations support both revision by highlighting key educational arguments and targeted feedback by clarifying student performance and future planning.
P7	The user has a track record of highlighting, underlining, and making handwritten notes to improve clarity and coherence in student writing, while also evaluating transitional phrases and main arguments.
P8	Previous annotations demonstrate a consistent approach to improving clarity, removing unnecessary words, and ensuring grammatical accuracy in students' writing, often focusing on seamless flow and concise expression.
P9	Previous annotations reflect an approach that emphasizes critical points in student writing, often highlighting areas for improvement and clarity.
P10	Previous annotations have involved circling phrases to indicate grammatical issues, unclear phrases, or themes relevant to English test marking, indicating a methodical approach to grading.
P11	Previous annotations aimed at enhancing clarity and articulation in student writing, often indicating vagueness or areas needing revision.
P12	The user's past annotations involve marking areas for grammatical corrections, emphasizing clarity, and enhancing logical flow, particularly in English tests, demonstrating a consistent focus on effective communication.

D Annotation Heatmaps

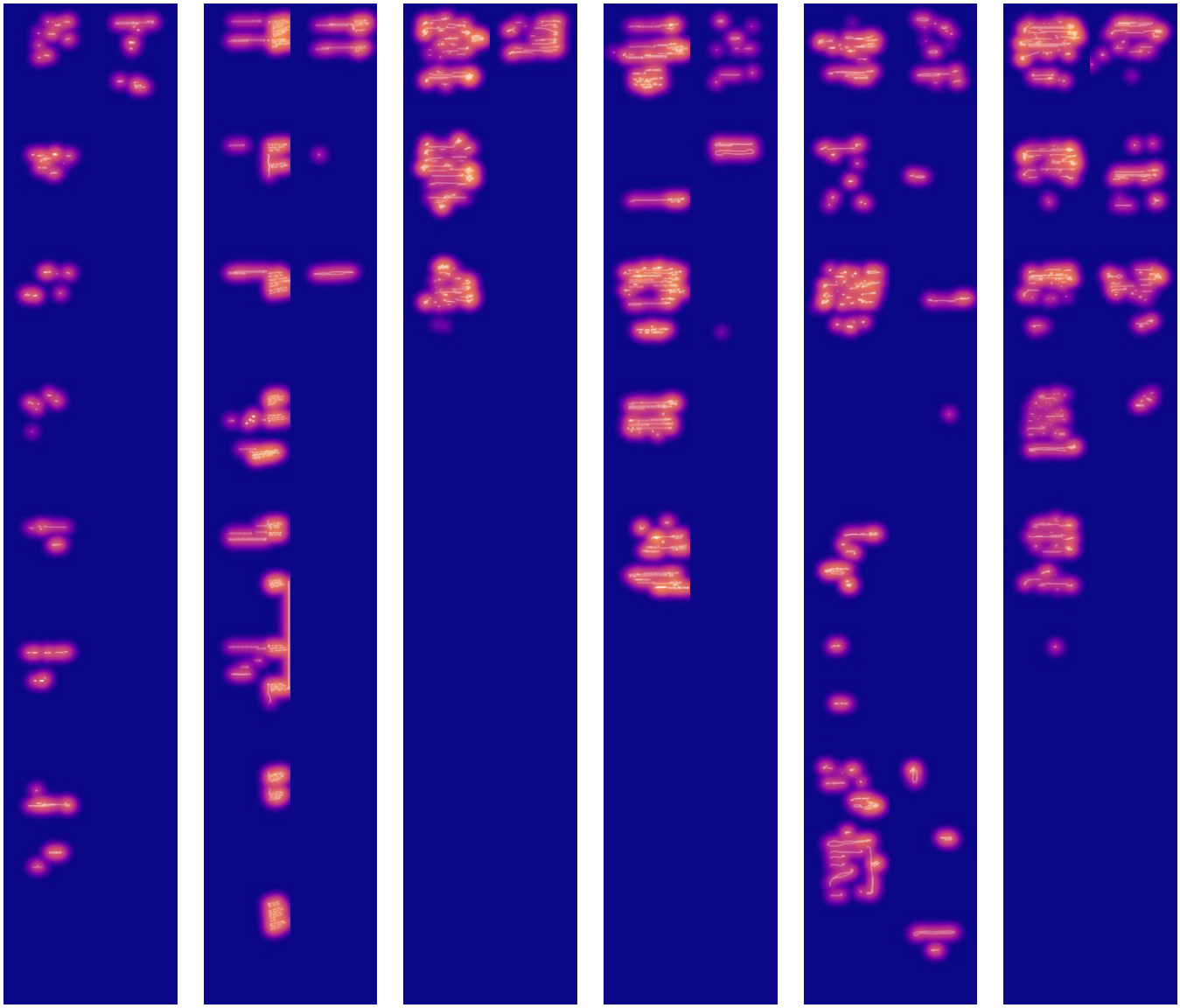


Figure 18: Heatmaps of P1-6 annotations, where the left heatmap of each figure is with the baseline and the right is with AnnotateGPT.

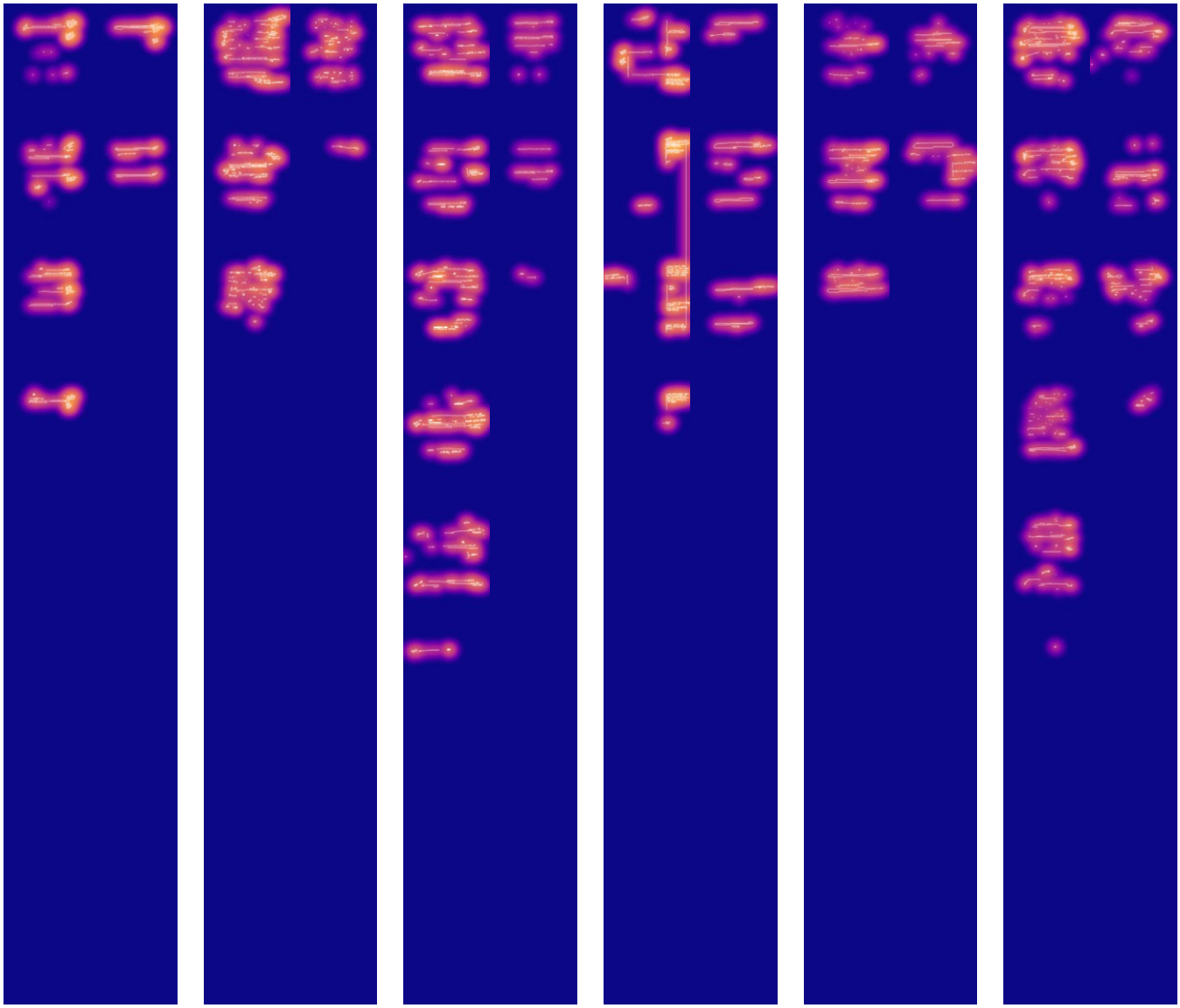


Figure 19: Heatmaps of P7-12 annotations, where the left heatmap of each figure is with the baseline and the right is with AnnotateGPT.

E Prompts for Annotation Classification and Purpose Inference

Table 5: The prompt for annotation classification and purpose inference. Blue text is only added when no stroke annotates more than two words to understand further how the user engages with content.

Prompt Role	Prompt
System	You are an expert in describing annotations and determining their purpose. You will be shown two images of annotations from a document a user has personally annotated. The first image shows the annotation with the document, and the second shows the annotation without the document. Do not give vague answers, such as whether the user is interested in or emphasizing the text; be very specific. Describe your steps first.
User	Types of annotation: - circles or boxes - underlining - highlighting - crossing out - handwritten notes/text - punctuation marks (e.g., commas, periods, question marks, asterisks, etc.), choose which one - arrows - brackets, angle brackets, or braces Here are the steps: 1. Describe the annotation by reviewing the list of annotation types for possibilities. 2. Guess the purpose of the annotation based on the context. 3. Use [four / two] branches of thinking such as backtracking to check for any other possibilities. 4. Look at past annotation history in your knowledge base 5. Summarize your past findings and relate them to the annotation 6. Give [four / two] different guesses of the purpose using different personas and past annotation history. The purposes should have different themes and relate to the context. 7. For each guess, give two levels of detail: specific and broad. When describing with specific, describe the purpose so it is specific to the words of the annotated text. When describing with broad, use umbrella terms without using the annotated text. [*annotation image*] [*annotation image without underlying text*] Context: The user is [*context*] and has [*annotation type and annotated text*].
User	Let's work this out in a step by step way to be sure we have the right answer.

F Prompts for Generating Annotations

Table 6: The prompt for generating annotations by extracting the sentence and its associated feedback and targeted words.

Prompt Role	Prompt
System	You are an expert at annotating documents. The user has annotated the document I have given you. Given the purpose of an annotation, you will find all sentences in the document that could be annotated for the same purpose. Do not change the sentence from the document in any way. Give one sentence in one annotation. Describe your steps first. Do not ask follow-up questions.
User	[*context*]. Read every page and find sentences that could be annotated with: [*purpose*] Here is a step-by-step list for annotating a document: 1. Describe what details in sentences to look for in the document. Be specific. Do not change the original purpose in any way. 2. Explain why you annotated the sentence. 3. Suggest fixes for the sentence by describing the fix without giving the answer. 4. Combine the explanation and suggestion without quoting the sentence using less than 20 words. 5. Do not include any sentences that need no modification. 6. Make a list of sentences for each response using triple asterisks for sentences and double curly braces for the explanation and suggestion. For example: ## Response <number> *** <sentence> *** {{ <explanation and suggestion> }} ... 7. For each sentence, you can optionally target words in the sentence to annotate. If you do, list the words or phrases to look for in the sentence, separated by commas and enclosed by triple quotation marks. For example: ## Response <number> *** <sentence> *** """ <words or phrase to look for (e.g. <word/phrase 1>, <word/phrase 2>)> """ {{ <explanation and suggestion> }} ... Make sure you have all the sentences needed to be annotated in the format above.
User	Walk me through one question at a time in manageable parts step by step, summarizing and analyzing as we go to make sure we have all the sentences needed to be annotated

G Stroke Clustering

The temporal distance between two pen strokes, (s_i, s_{i+1}) is:

$$\Delta t = \begin{cases} \dfrac{t_0(s_{i+1}) - t_f(s_i)}{30} & \text{if } t_0(s_{i+1}) - t_f(s_i) < 30 \text{ seconds}, \\ 1 & \text{otherwise}, \end{cases}$$

where $t_0(s_{i+1})$ is the start time of s_{i+1} and $t_f(s_i)$ is the end time of s_i. The temporal distance has an upper bound of 30 seconds to ensure normalization across all pen stroke pairs. The spatial distance between two pen strokes is the minimum Euclidean distance between two bounding boxes of s_1 and s_2, normalized by page size. Thus, the spatiotemporal distance is defined as:

$$d_{st} = \sqrt{\Delta s^2 + \Delta t^2}$$

Hierarchical agglomerative clustering merges the closest pen strokes, but the challenge lies in not knowing the number of clusters in advance. A heuristic can help identify this number automatically [27]. During iterations, if there is a sharp increase in the closest pairwise distance, it indicates a "forced merge," which combines distant clusters. Thus, identifying the optimal stopping iteration, i^*.

$$i^* = \arg\max_i \left[\frac{d_{i+1} - d_i}{d_i - d_{i-1}} \times (d_{i+1} - d_i) \right]$$

The ratio in this expression compares the increase between consecutive iterations to detect sharp increases. However, if the prior increase is small, even a minor rise can skew the ratio early on. The method prioritizes larger global leaps by using the absolute increase $(d_{i+1} - d_i)$ as a scaling factor to address this.

H Post-Study Interview Questions

1. What types of things did you annotate when grading the test?
2. Which annotation system did you prefer and why?
3. What were your first impressions of the assistant?
4. Why did your annotations not have textual feedback when using the assistant?