ChatGPT for Education Research: Exploring the Potential of Large Language Models for Qualitative Codebook Development

Amanda Barany[1[0000-0003-2239-2271]], Nidhi Nasiar[1[0009-0006-7063-5433]], Chelsea Porter[1[0009-0005-0246-8509]], Andres Felipe Zambrano[1[0000-0003-0692-1209]], Alexandra L. Andres[1[0000-0001-7509-1574]], Dara Bright[2], Mamta Shah[1,3[0000-0002-4932-2831]], Xiner Liu[1], Sabrina Gao[1], Jiayi Zhang[1[0000-0002-7334-4256]], Shruti Mehta[1], Jaeyoon Choi[4], Camille Giordano[1] and Ryan S. Baker[1[0000-0002-3051-3232]]

[1] The University of Pennsylvania, Philadelphia PA 19104, USA
[2] Consortium of DEI Health Educators, Philadelphia PA, USA
[3] Elsevier, Philadelphia PA 19103, USA
[4] University of Wisconsin-Madison, Madison WI 53706, USA
amanda.barany@gmail.com

Abstract. In qualitative data analysis, codebooks offer a systematic framework for establishing shared interpretations of themes and patterns. While the utility of codebooks is well-established in educational research, the manual process of developing and refining codes that emerge bottom-up from data presents a challenge in terms of time, effort, and potential for human error. This paper explores the potentially transformative role that could be played by Large Language Models (LLMs), specifically ChatGPT (GPT-4), in addressing these challenges by automating aspects of the codebook development process. We compare four approaches to codebook development - a fully manual approach, a fully automated approach, and two approaches that leverage ChatGPT within specific steps of the codebook development process. We do so in the context of studying transcripts from math tutoring lessons. The resultant four codebooks were evaluated in terms of whether the codes could reliably be applied to data by human coders, in terms of the human-rated quality of codes and codebooks, and whether different approaches yielded similar or overlapping codes. The results show that approaches that automate early stages of codebook development take less time to complete overall. Hybrid approaches (whether GPT participates early or late in the process) produce codebooks that can be applied more reliably and were rated as better quality by humans. Hybrid approaches and a fully human approach produce similar codebooks; the fully automated approach was an outlier. Findings indicate that ChatGPT can be valuable for improving qualitative codebooks for use in AIED research, but human participation is still essential.

Keywords: Large Language Models, ChatGPT, Inductive Coding, Research Methods.

1 Introduction

When examining qualitative data in education research, the process of "coding", or defining concepts and identifying where they occur in the data, is a key part of the

meaning-making process [32]. Some coding projects are driven by top-down deductive approaches that apply codes from existing codebooks or frameworks [3]. Researchers have also found value in inductive, bottom-up coding techniques that ground codes in the research context (sometimes referred to as thematic analysis [5]). Inductive codes allow meaning to emerge from frequent, dominant, or significant themes in the data [4, 36]. Given these affordances, inductive codebook development has been featured in a variety of educational research contexts in the last decade, from small-scale case studies (e.g., [25]) to large-scale meta-summaries of education research (e.g., [1]).

While widely used in education research, inductive codebook development is not without its challenges. The practices for initially developing inductive codes and then applying them to the dataset are often inconsistent and there are often issues with reliability and fairness [17, 32]. To help address these issues, researchers often adopt three practices: (1) transparent procedures and documentation when creating a preliminary codebook [38], (2) pilot testing codes during codebook refinement [32], and (3) inter-coder reliability checks [30]. However, these efforts can be time-consuming (e.g., [8, 34]). Campbell et al. [7] also note a tradeoff in traditional inductive code development between time spent (efficiency), the utility of the final codebook for representing nuance, and the reliability with which coders can later code the data. As our field now works with increasingly large-scale and complex learning data, we need techniques that can maximize efficiency without limiting code utility or reliability.

A potential solution is the automation of coding processes. Though there have been decades of research attempting this (e.g., [29, 37]), existing tools have been critiqued as producing low-quality codes or codes that miss nuances that humans identify [18] and have been unable to explain the reasoning for their recommendations [26], which are significant limitations for codebook development. Large Language Models (LLMs) such as ChatGPT (GPT-4), however, draw on advanced natural language processing capabilities to process and generate human-like text based on open-ended textual input.

To support inductive coding, ChatGPT could be used in two phases of the codebook development process: (1) to identify preliminary codes from data; and (2) to test and refine codes. While scholars have already used ChatGPT for both codebook development and codebook refinement [16, 19], there has not yet been a systematic study of what benefits it can bring to each phase of this process. In this work, we tried four different ways to create a codebook for the same data: a fully manual approach, a fully automated approach, and two hybrid approaches that leverage ChatGPT, one in codebook development and one in codebook refinement. To understand when and how ChatGPT might best support the codebook development process, we compare time spent on each approach, reliability among human coders in applying the codes, ratings of codebook utility by human coders, and whether approaches yielded similar codes.

2 Automated Tools for Qualitative Data Analysis

Over time, there has been considerable interest in using natural language processing tools to support codebook development [14, 20] and automated coding [6, 15, 22]. However, qualitative coders have largely not adopted these tools, due to concerns that the codes obtained are limited in quality [18]. There has also been considerable interest

in automatically coding data (given a codebook or examples) using a range of natural language processing methods (e.g., [9, 10, 14, 23, 42]), including LLMs (e.g., [11]).

Recent work has explored the use of ChatGPT for deductive coding [35] with some researchers testing the reliability of ChatGPT compared to humans or other automated coding tools [41]. Törnberg [38] proposed a step-by-step process that includes "mutual learning" between researchers and ChatGPT, where researchers iteratively refine the prompts given to ChatGPT to improve its responses and use ChatGPT's feedback to refine deductive codes and apply them reliably. Zhang et al. [44] similarly describe ChatGPT as a "tool" for data synthesis and as a "co-researcher" that can assist, challenge, or supplement coders' interpretations. Zambrano et al. [43] found that ChatGPT's explanations for coding decisions had the potential to improve construct validity, find ambiguity in codebook definitions, and help human coders achieve better interrater reliability. Other researchers have raised concerns that ChatGPT classification can be nonreproducible [30], recommending human review [28].

Two projects have explored the application of ChatGPT for inductive code development using thematic analysis. De Paoli [16] offered a 5-phase process model with LLMs that uses prompts to first identify and refine inductive codes in a dataset, and then pairs a researcher and ChatGPT to review and finalize themes. Gao et al. [19] propose that ChatGPT may have a role at three points during thematic analysis: (1) suggesting codes during open coding, (2) identifying disagreements when researchers refine and test the codes, and (3) suggesting which codes to combine during code finalization. While these studies show how ChatGPT can help develop codes, they do not test how reliably coders could apply those codes to the datasets or compare its utility at different phases of the process. The aim of our research is, therefore, to investigate whether ChatGPT is beneficial in either or both initial codebook development and refinement, in different structures of human/LLM collaboration.

3 Methods

3.1 Data Source

We conduct this research in the context of high-dosage tutoring lessons, in which trained tutors offer personalized, small-group support for mathematics learning (e.g. [13, 24]). Prior research has shown that this type of tutoring benefits students' math learning, achievement, and grades [12]. We obtained transcripts from four 60-minute tutoring sessions conducted by the non-profit organization Saga Education. Sessions were conducted virtually with students in high-poverty schools in an urban region of the northeastern United States from 2022-2023. Students were 9th graders enrolled in Algebra I. Sample data from a tutoring session is shown in Figure 1.

id	speaker_type	text
74	tutor	In total, how many x's do we have now?
75	student	Uh, seven.
76	tutor	Exactly.
77	tutor	Now we have seven.
78	tutor	So what should the new exponent of x be?
79	student	x to the seven.
80	tutor	Exactly.

Fig. 1. A sample of transcript data from a tutoring lesson on the Saga platform.

3.2 Codebook Development

To explore the utility of ChatGPT as a tool for inductive codebook development, four approaches were applied by four different members of the research team to the same three tutoring lesson transcripts. The focus of qualitative coding in all cases was primarily to identify the instructional strategies or techniques used in the tutoring lessons, though other emergent codes were also included. For each of the approaches described below, the researchers worked independently during this stage to avoid biasing those engaging in the other approaches. The total time spent developing the four codebooks was logged by researchers to compare the efficiency of each approach.

The human-only approach applies common practices considered a gold standard for inductive coding in education research [33], while the other three approaches used ChatGPT (GPT-4) to automate some or all of the process. Within the approaches that used ChatGPT, we opted for the web-based chatbot version over the API, anticipating that future researchers might prefer the web version for more straightforward interaction with the chatbot when developing and refining a codebook. Our study design required that authors work exclusively on one codebook to avoid skewing the results due to the order of codebook development (it is expected that the next codebook developed by the same researcher using the same data would have higher quality). Therefore, although codebook development, revision, and refinement is usually a collaborative process [40], for this study each codebook developer (authors 2 to 5) worked independently to avoid biasing other members of the research team.

Human Only (H). For approach H, a researcher used the codebook development process outlined by Weston et al. [40] including stages of code conceptualization, application/review, and refinement. The researcher (author 5) first qualitatively reviewed the dataset to identify common patterns and themes that appeared across the transcripts in an inductive search for tutor/student exchanges that demonstrated specific instructional strategies or techniques. These themes were manually organized into a preliminary codebook that included tentative code names, definitions, and example quotes. The preliminary codebook was then used to reexamine the dataset to determine if revisions to the inclusion and exclusion criteria were necessary or if any new themes emerged. This process was repeated until no further additions or revisions were needed.

Human Code Development, ChatGPT Refinement (HC). For approach HC, a human coder (author 3) again engaged in qualitative review and the creation of a preliminary codebook as outlined by Weston et al. [40]. The researcher then tasked ChatGPT with reexamining and refining the codebook by entering the following prompt:

> *You are a researcher helping develop a qualitative codebook for text data of a math tutor's interactions with students. I will give you the first draft of the codebook, and then the data being coded, in batches. Please help me refine the codes and codebook, focusing on instructional strategies or techniques.*

The preliminary codebook and full dataset were then entered into ChatGPT, with the three class sessions divided into 13 batches of 77-98 lines. This range was selected to

maximize batch sizes (more than 75 lines), while not exceeding the processing limits of the version of ChatGPT we used (which was 4096 tokens corresponding to approximately 100 lines of our dataset). While it would be ideal for subsets to maintain a uniform line count, it was not possible to achieve this without segmenting the data in the middle of a response or explanation that required context from prior lines; breakpoints for batches were selected to minimize this type of context loss.

Every three to four batches of data, the researcher prompted ChatGPT to offer further refinement: *Please give me a refined codebook, with examples, based on all X batches of data so far*. After the final batch entry and refinement, the researcher reviewed the codebook to check for errors or inconsistencies (in response to Mesec's [27] caution that all ChatGPT output must be evaluated by a human prior to dissemination). While the refined codes and definitions were consistent with the researcher's understanding of the data, random checks of the examples showed many example quotes were hallucinations. The researcher replaced them with quotes from the original dataset.

ChatGPT Code Development, Human Refinement (CH). For approach CH, a researcher (author 2) tasked ChatGPT with creating a preliminary codebook before engaging in manual re-examination and refinement, using the process in Weston et al. [40] for that stage. The researcher began with an initial review of the data to understand its structure and context but did not exhaustively analyze the dataset to identify themes. Another researcher (author 4) tested multiple prompts with ChatGPT to identify the one that could most reliably provide codes, definitions, and examples. Prompts were crafted based on best practices from existing prompt engineering frameworks (e.g., [27]), which emphasize offering ChatGPT clear, concise, and specific task descriptions to ensure the model receives the necessary information but is not confused by unnecessary details. The first 100 lines of the dataset were used for prompt development and testing.

Recognizing ChatGPT's challenges in maintaining response consistency (due to the variation of the chatbot's responses) the researcher re-evaluated each prompt across sessions with ChatGPT, using various browsers and computers. After identifying a prompt that produced consistently similar responses—where responses generated by repeated tests of the same prompt had no more than two codes that were different across runs—this test was replicated using two additional sets of 100 lines. Once the prompt's consistency was confirmed across these new sets, it was applied to the entire dataset using the same batches detailed in approach HC. The final prompt read:

Hi ChatGPT, I want to analyze the following interaction between an instructor and some students: [DATA] Please give me a codebook to analyze the instructional methodologies and the sentiment within this interaction.

The result of this process was a codebook of 8-12 themes for each batch. All themes not proposed at least three times across the 13 batches were discarded. The themes that the researcher identified as conceptually similar were grouped together into a single theme. Across the three or more examples of each code, the most straightforward definition was selected, or two were combined to create the preliminary codebook.

After the ChatGPT part of the process had been completed, the researcher (author 2) used Weston et al.'s [40] approach to codebook application/review and refinement described in approach H, repeatedly applying the codes to the dataset to refine them until no further code revisions or additions were necessary.

ChatGPT Only (C). For approach C, a researcher (author 4) used ChatGPT to develop the preliminary codebook using the procedures and prompts detailed in approach CH and used ChatGPT for the review and refinement of the codebook using the procedures and prompts detailed in approach HC. The researcher did not add any concepts or clarifications not originally provided by ChatGPT. The final version of the codebook also contains three example quotes given by ChatGPT for each code. If more than three examples were provided by ChatGPT, the researcher selected the three clearest examples. No hallucinations were obtained in this condition. Beyond this review, the researcher made no additional interventions for the final version of the codebook.

3.3 Coding Procedures

Once the codebooks were finalized, four pairs of researchers (authors 6-13) who were not involved in the development process and were not familiar with the data were randomly assigned to code the tutoring lesson transcripts using one of the four codebooks. The pairs were introduced to the study design and the context of the dataset but were not told which approach produced the codebook they used. Researchers were instructed to independently mark each code as present (1) or absent (0) for each line of data based on the code's name and the inclusion/exclusion criteria provided in the definition. Upon completion of the first round of coding, coders submitted a brief survey in which they rated their codebook on a scale of 1 (lowest) to 5 (highest) for ease of use, clarity, and the mutual exclusivity and exhaustiveness of codes.

After the pairs had independently coded all lines of data, Cohen's kappa (κ) was used to assess the consistency of code applications. Research pairs then met virtually for 1-hour sessions via Zoom, where they were prompted to discuss and resolve coding inconsistencies in the dataset using social moderation techniques (e.g., [21]). Codebook developers did not participate in this process or clarify any aspect of the codebook to avoid biasing the process. Coders were invited to annotate their codebooks based on what was learned from social moderation, then code a fourth lesson transcript consisting of 150 lines of new tutoring data based on their refined understanding of codes. Cohen's κ coefficients were calculated for this second round of coding.

3.4 Codebook Evaluation

To evaluate the utility of each codebook, at the end of the coding process, coders were surveyed to evaluate their perception of their codebook's ease of use, code clarity, mutual exclusivity, and exhaustiveness using a Likert Scale ranging from 1 (lowest) to 5 (highest). Criteria were chosen based on previously published principles for what constitutes a high-quality codebook [4, 5]. The level of agreement between the coders for each construct on each codebook was employed as a proxy for codebook quality. Separate researchers (authors 1 and 14) also evaluated the conceptual overlap across each of the four codebooks. One hundred pairs of codes from two codebooks for the same data point were randomly generated, with each codebook represented in at least 25 pairs and each code represented at least once. The researchers, who were not involved in codebook development, independently coded each pair as representing the same concept (1) or representing different concepts (0). While the coders reached a high percentage of agreement for the first 100 pairs, few instances of conceptual similarity occurred. Fifty-two additional code pairs with potential for conceptual

similarity were purposively sampled based on review of the blinded codebooks. Both researchers then independently coded the additional pairs, obtaining a Cohen's κ coefficient of 0.86. With inter-rater reliability established, the first author reviewed all code pairs to identify every instance of code overlap across the four codebooks.

4 Results

4.1 Time Spent

Analyzing time spent can shed light on the efficiency of each of the four approaches. Table 1 gives the total time spent by the researchers on processes up to and including development of a preliminary codebook, and total time spent on codebook refinement and finalization. Time spent engaging in preliminary development varied by codebook and researcher; approach H took longest (180 minutes) and C the shortest (50 minutes).

In approaches HC and CH, which both used ChatGPT to automate code refinement, more time was spent engaging in codebook refinement than codebook development. Automated code refinement for these approaches also took more time to complete than human code refinement in approaches H or C. This may be because human intervention was still needed to make final decisions or adjustments after ChatGPT had completed the automated refinement of codes. In approach HC, the researcher noticed that ChatGPT had hallucinated example quotes to populate the final codebook and spent time replacing them with genuine quotes. In approach C, the researcher spent time merging similar codes that repeatedly emerged across the repeated prompts. This multi-step refinement may have contributed to the longer time spent. Some of the differences in time spent in each process may also relate to individual variations by the researcher.

Table 1. Records of time spent on codebook development.

Codebook Approach	Preliminary Codebook Development	Codebook Refinement	Total Time Spent
(1) Human	180 minutes	40 minutes	220 minutes
(2) Human → ChatGPT	80 minutes	165 minutes*	245 minutes
(3) ChatGPT → Human	107 minutes	60 minutes	167 minutes
(4) ChatGPT	50 minutes	63 minutes**	113 minutes

* 90 minutes spent using ChatGPT for code refinement, 75 minutes of human codebook revision
** 42 minutes spent using ChatGPT for code refinement, 21 minutes of human codebook revision

4.2 Codebook Utility

Coders' (authors 6-13) Likert-style rankings and qualitative reflections on the codes they applied to the datasets offer preliminary insights into the quality and utility of the codebooks developed using each approach. Table 2 summarizes coders' rankings of each codebook's ease of use, code clarity, mutual exclusivity, and exhaustiveness from 1 (lowest) to 5 (highest). Approach HC and CH, which leveraged combinations of manual and automated techniques to develop and refine codes, were ranked highest for clarity and mutual exclusivity of codes, as well as for the codebooks' ease of use.

Approach H was ranked lowest for code exhaustiveness, suggesting that coders may have noticed themes in their review of data that were not represented in the codebook. Given that the researcher for approach H used only manual techniques when developing the codebook, they may have been more likely to emphasize instructional strategies that emerged in the data – the research focus – and not develop other codes (discussed further below). Approach C was ranked lowest for the mutual exclusivity of codes, suggesting that coders felt these codes had more conceptual similarities. For example, codes in the ChatGPT codebook such as *Direct Instruction* (providing direct information) compared to *Task Assignment* (directing students to a task) used similar terms to describe different concepts, and *Questioning and Check-in* (asking questions to probe understanding) compared to *Metacognition* (encouraging student reflection on processes), which describe phenomena that might sometimes overlap (e.g., a check-in that induces metacognition), making it challenging to categorize them distinctly.

Table 2. Coder ratings of codebook utility.

Codebook Approach	Ease of use	Clarity of codes	Mutual Exclusivity	Exhaustiveness
(1) H	2.5	3	3	2
(2) HC	4.5	3.5	3.5	3
(3) CH	4	4	3.5	2.5
(4) C	3	3	1.5	3

4.3 **Inter-Rater Reliability**

In the final codebooks, approach H (Human) had 9 codes, approach HC (Human → ChatGPT) had 10 codes, approach CH (ChatGPT → Human) had 8 codes, and approach C (ChatGPT) had 11 codes. Table 3 provides an overview of percent agreement and Cohen's κ coefficients across two rounds of paired independent coding to explore whether codes from each codebook can be applied consistently and reliably. For round 1 of coding, two out of nine codes for approach H and two out of ten codes for approach HC (Human → ChatGPT) achieved a κ of 0.6 or above. Coders achieved high first-round agreement for codes related to mentorship enacted by the tutors, such as *Providing Assistance* (0.60), *Checking in or Expressing Concern* (0.71), and *Offering Greetings or Pleasantries* (0.64), and for students asking questions (*Questioning*, 0.83). One error occurred in this process: in two instances, one coder using the approach H codebook interpreted a pair of consecutive codes as a single code and applied them to the data as such; as a result, paired agreement could not be calculated for four codes against the other coder in round 1, who applied them as four independent codes.

Coders using the ChatGPT codebook saw some of the lowest agreement measures for codes in round 1 (average $\kappa = 0.21$, compared to 0.34, 0.38, and 0.26 for other codebooks), which may relate to concerns about code mutual exclusivity they identified in their final reflections. For example, when the pair met to refine their understanding of the codes before round 2 of coding, they noted that two codes (*Clarification and Reiteration* and *Corrections*) both included the term "clarification" in either their name or definition, which muddied coders' understanding of how they differed. Annotations from their discussion highlight efforts to emphasize what makes each code distinct.

For round 2 of coding, average κ coefficients for all four codebooks improved. Coders using the ChatGPT → Human codebook saw universal improvements in their agreement in round 2, with every code reaching a κ value at 0.6 or above (average κ = 0.70). Annotations from their discussion show how terms from the definitions offered concrete examples for how the code might appear in the data (e.g., *Aligning to Prior Knowledge*, "Tutor…using the word 'remember'"). However, this code pair chose to meet more times to continue refining and improving their shared understanding of the codes than the other pairs, which likely explains their improved inter-rater agreement.

Four out of nine codes for approach H and four out of ten codes for approach HC reached a κ of 0.6 or above in round 2. New codes to reach this threshold include *Comfort/consolation* and *Prompting Self-explanation* for approach H and *Clarification/rephrasing* and *Feedback* for HC. New codes in approach H did not have prior measures of agreement due to the coding inconsistency in approach H round 1.

For the ChatGPT codebook (approach C), the code *Friendly Interaction and Encouragement* reached κ of 0.6 in round 2. Some of the codes that were clarified during the pair's round 2 discussion, such as *Direct Instruction,* and *Questioning and Check-in,* saw improvement in inter-rater agreement, but did not reach the threshold for moderate agreement. Other codes saw minimal improvement or decrease in κ values.

In approach HC (4 out of 10 codes) and C (2 out of 11 codes), some κ values decreased when codes were applied to new data, and in three of the four codebooks, one or more codes did not appear in the new dataset – a common challenge when qualitatively coding relatively varied data.

Table 3. Measures of agreement across two rounds of hand coding for each item.

Approach	Code	Cohen's κ coefficient	
		Round 1	Round 2
(1) Human	1. Assistance	**0.60**	**0.62**
	2. Encouragement	0.05	0.26
	3. Checking in/concern	**0.71**	**0.80**
	4. Comfort/consolation	**	**0.61**
	5. Commendation	**	0.16
	6. Prompt self-explanation	**	**0.61**
	7. Relating/casual	0.04	0.31
	8. Scaffolding	**	0.35
	9. User interface issues	0.29	*
	Average κ	0.33	0.47
(2) Human → ChatGPT	1. Clarification/rephrasing	0.04	**0.65**
	2. Connecting to prior knowledge	0.36	*
	3. Direct instruction	0.22	0.17
	4. Engagement checks	0.45	*
	5. Feedback	0.21	**0.62**
	6. Greetings/pleasantries	**0.64**	**0.75**
	7. Guided practice	0.29	0.19
	8. Questioning	**0.83**	**0.87**
	9. Session logistics	0.32	0.15
	10. Software/tool use	0.44	0.38
	Average κ	0.38	0.47

	1. Aligning to Prior Knowledge	0.40	**0.66**
	2. Checking Understanding/Engagement	0.45	**0.60**
	3. Encouragement	0.12	**0.80**
(3) ChatGPT	4. Greeting	0.43	**0.85**
→ Human	5. Guiding Feedback	0.05	**0.66**
	6. Instruction	0.21	**0.66**
	7. Technical and Logistics	0.09	**0.66**
	8. Time Management	0.31	**0.72**
	Average κ	0.26	0.70
	1. Clarification and reiteration	0.01	*
	2. Corrections	-0.02	*
	3. Direct instruction	0.07	0.24
	4. Expressions of frustration/impatience	0.20	*
	5. Friendly interaction and encouragement	0.42	**0.67**
(4) ChatGPT	6. Guided practice	0.37	0.46
	7. Metacognition	0.50	0.49
	8. Student uncertainty	0.20	-0.01
	9. Task assignment	0.30	0.30
	10. Technical problem addressal	0.21	0.28
	11. Questioning and check-in	0.08	0.55
	Average κ	0.21	0.37

* Code did not appear in sample dataset

** Coder treated two codes as a single code; agreement with second coder could not be calculated

4.4 **Conceptual Overlap**

The cross-codebook comparison to identify common themes resulted in 28 total pairs of codes across the four codebooks that were found to represent the same concept. Pairs aligned around nine general categories of overlap as illustrated in Table 4. Themes were identified as conceptual outliers if they appeared in only one codebook.

Four themes with universal representation across codebooks were *Checking In*, *Feedback*, *Guided Practice*, and *Technical Issues*. Three of these codes relate directly to the research focus for each approach: instructional strategies or techniques used in tutoring lessons. The fourth, *Technical Issues*, was common in the datasets, as tutors offered platform or logistical support throughout lessons. While the names and definitions had slight differences across codebooks (e.g., *Checking in/concern* versus *Questioning and check-in*), the high level of overlap suggests that all four approaches can lead to the identification of key themes if they are prevalent in the dataset.

The largest overlap between approaches was between approach HC and approach CH, the two hybrid Human-ChatGPT approaches. Seven categories were found in both codebooks, with only two categories found in HC but not CH, and only one category found in CH but not HC. Approach HC (Human → ChatGPT) had the greatest conceptual similarity to the manual approach H, finding all themes represented in the human codebook as well as three other themes. Approach CH (ChatGPT → Human) also captured most of what was seen in the human codebook (five of six themes), plus three additional themes. As such, it seems that hybrid approaches can capture most or all of what a pure human approach captures, plus additional codes. Approach C (all ChatGPT) had the lowest agreement with the other three approaches.

Overall, approaches CH and C, which used ChatGPT to automate preliminary codebook development, were more likely to generate novel themes that did not appear in any other codebooks. These themes primarily focused on student behaviors and affective states. These categories tended to be less prevalent in the dataset, and less related to the project goal. Approach C (ChatGPT only) also missed the prevalent theme *Greetings/casual*, which characterizes the casual and introductory interactions between tutor and students during sessions. All told, the fully automated approach to codebook development emerged as an outlier in terms of conceptual similarity.

Table 4. Heat map of code categories across codebooks.

Code Categories	H	HC	CH	C	Total
(1) Checking in	X	X	X	X	4
(2) Feedback	X	X	X	X	4
(3) Guided Practice	X	X	X	X	4
(4) Technical Issues	X	X	X	X	4
(5) Greetings/ casual language	X	X	X	0	3
(6) Questioning or response prompting	X	X	0	0	2
(7) Connecting to prior knowledge	0	X	X	0	2
(8) Logistics/Time Management	0	X	X	0	2
(9) Direct Instruction	0	X	0	X	2
*Student Responses	0	0	X	0	1
*Student Uncertainty	0	0	0	X	1
*Frustration or Impatience	0	0	0	X	1

* Themes that emerged as outliers in the codebooks

5 Discussion and Conclusion

In this paper, we explored when and how ChatGPT might support the process of inductive codebook development. We applied a fully manual approach, a fully automated approach, and two hybrid approaches to creating codebooks for math tutoring transcripts. The hybrid approaches involved utilizing ChatGPT for either preliminary codebook development or refinement. For each approach, we compared the time spent, ratings of codebook utility, and inter-rater reliability metrics. Lastly, we assessed whether different methods produced similar or overlapping themes.

Results indicate that automating elements of codebook development has the potential to improve the time efficiency of the process, especially when both preliminary development and final refinement are automated (approach C). However, we found that humans could not be excluded completely from the process; even when codebook refinement was automated by ChatGPT, further human refinement was still needed to address errors or inconsistencies. Thus, it may make the most sense to use automation initially and then transition to manual refinement (e.g., approach CH).

The fully automated codebook was ranked lowest for utility, had the lowest inter-rater reliability measures when applied by coders, and saw the least conceptual overlap with other codebooks. This aligns with Reiss' [29] cautions regarding the consistency and reliability of results when ChatGPT is used without human supervision. While the fully human process was able to identify many codes that were consistent across other

codebooks, it also missed some themes that were represented in the two hybrid approaches (e.g., *Logistics/Time Management*). Overall, the hybrid approaches received the highest utility ratings, achieved comparable or better inter-rater reliability outcomes than other approaches, and had the highest conceptual overlap. These findings indicate that ChatGPT can be useful as both a tool and co-researcher to support "mutual learning" between humans and LLM [38, 44].

There is considerable further work to be done to understand how LLMs and humans can best work together for qualitative coding. As the first comparison of different approaches to using LLMs for inductive codebook development, our study has several limitations that can be addressed in future studies. For example, a tighter comparison could have been achieved by more precise coding instructions and by controlling the amount of time spent working on achieving inter-rater reliability.

This research offers insights into how Large Language Models can be integrated into the inductive codebook development process for qualitative data analysis in education research. Findings suggest that while automated approaches can enhance efficiency, the collaboration between human researchers and ChatGPT is most beneficial for producing high-quality, non-overlapping, and comprehensive codebooks where human coders can obtain reliable results. We hope that this study provides a foundation for further exploration and refinement of methodologies, emphasizing the potential of hybrid approaches for leveraging the strengths of automated and manual processes.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

Acknowledgements. This work was supported by funding from the Learning Engineering Virtual Institute (LEVI) Engagement Hub. All opinions expressed are those of the authors.

References

1. Anderson, J., & Taner, G.: Building the expert teacher prototype: A metasummary of teacher expertise studies in primary and secondary education. *Educational Research Review 38*, 100485 (2023). https://doi.org/10.1016/j.edurev.2022.100485
2. Bakharia, A.: On the equivalence of inductive content analysis and topic modeling. In: Eagan, B., Misfeldt, M., Siebert-Evenstone, A. (eds.) Advances in Quantitative Ethnography: First International Conference, ICQE 2019, Madison, WI, USA, October 20–22, 2019, Proceedings 1 (pp. 291-298). Springer International Publishing (2019).
3. Bingham, A. J., & Witkowsky, P.: Deductive and inductive approaches to qualitative data analysis. In: Vanover, C., Mihas, P., Saldana, J. (eds.) *Analyzing and Interpreting Qualitative Data: After the interview*, 133-146 (2021).
4. Boyatzis, R.: *Transforming qualitative information: Thematic analysis and code development*. Sage, Thousand Oaks, CA (1998).
5. Braun, V., & Clarke, V.: Thematic Analysis. In: Cooper, H., Camic, C.M., Long, D.L., Panter, A.T., Rindskopf, D., & Sher, K.J. (eds.) *APA handbook of research methods in psychology, Vol. 2. Research designs: Quantitative, qualitative, neuropsychological, and biological*, pp. 57-71. American Psychological Association (2012).
6. Cai, Z., Siebert-Evenstone, A., Eagan, B., Shaffer, D. W., Hu, X., & Graesser, A. C.: nCoder+: a semantic tool for improving recall of nCoder coding. In: Eagan, B., Misfeldt, M., Siebert-Evenstone, A. (eds.) *Advances in Quantitative Ethnography. ICQE 2019. Communications in Computer and Information Science, vol 1112*. Springer (2019).

7. Campbell, J. L., Quincy, C., Osserman, J., & Pedersen, O. K.: Coding in-depth semistructured interviews: Problems of unitization and intercoder reliability and agreement. *Sociological Methods & Research 42*(3), 294-320 (2013).
8. Castleberry, A., & Nolen, A.: Thematic analysis of qualitative research data: Is it as easy as it sounds?. *Currents in Pharmacy Teaching and Learning 10*(6), 807-815 (2018).
9. Chen, N. C., Drouhard, M., Kocielnik, R., Suh, J., & Aragon, C. R.: Using machine learning to support qualitative coding in social science: Shifting the focus to ambiguity. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, *8*(2), 1-20 (2018).
10. Cher, P. H., Lee, J. W. Y., & Bello, F.: Machine Learning Techniques to Evaluate Lesson Objectives. In: *International Conference on Artificial Intelligence in Education*, pp. 193-205. Springer International Publishing (2022, July).
11. Cochran, K., Cohn, C., Rouet, J. F., & Hastings, P.: Improving Automated Evaluation of Student Text Responses Using GPT-3.5 for Text Data Augmentation. In: *International Conference on Artificial Intelligence in Education*, pp. 217-228. Cham, Springer Nature Switzerland (2023, June). https://doi.org/10.1007/978-3-031-36272-9_18
12. Cook, P. J.: Not too late: Improving academic outcomes for disadvantaged youth. Northwestern University Institute for Policy Research Working Paper, 15-01 (2015).
13. Cook, P.J., Dodge, K., Farkas, G., Fryer, R.G., Guryan, J., Ludwig, J. and Steinberg, L.: The (surprising) efficacy of academic and behavioral intervention with disadvantaged youth: results from a randomized experiment in Chicago, Working Paper No. 19862. National Bureau of Economic Research (2014). https://doi.org/10.3386/w19862
14. Crowston, K., Allen, E. E., & Heckman, R.: Using natural language processing technology for qualitative data analysis. *Int'l J. of Social Research Methodology*, *15*(6), 523-543 (2012).
15. Crowston, K., Liu, X., & Allen, E. E.: Machine learning and rule-based automated coding of qualitative data. In: *Proceedings of the American Society for Information Science and Technology*, *47*(1), 1-2 (2010). https://doi.org/10.1002/meet.14504701328
16. De Paoli, S.: Performing an Inductive Thematic Analysis of Semi-Structured Interviews With a Large Language Model: An Exploration and Provocation on the Limits of the Approach. *Social Science Computer Review*, 08944393231220483 (2023).
17. Eagan, B. R., Rogers, B., Serlin, R., Ruis, A. R., Arastoopour Irgens, G., & Shaffer, D. W.: Can we rely on IRR? Testing the assumptions of inter-rater reliability. In: *International Conference on Computer Supported Collaborative Learning* (2017, January).
18. Gao, J., Choo, K. T. W., Cao, J., Lee, R. K. W., & Perrault, S.: CoAIcoder: Examining the effectiveness of AI-assisted human-to-human collaboration in qualitative analysis. *ACM Transactions on Computer-Human Interaction*, *31*(1), 1-38 (2023).
19. Gao, J., Guo, Y., Lim, G., Zhan, T., Zhang, Z., Li, T. J. J., & Perrault, S. T.: CollabCoder: A GPT-Powered Workflow for Collaborative Qualitative Analysis. *arXiv preprint arXiv:2304.07366* (2023). https://doi.org/10.48550/arXiv.2304.07366
20. Gauthier, R. P., & Wallace, J. R.: The computational thematic analysis toolkit. In: *Proceedings of the ACM on Human-Computer Interaction*, *6*(GROUP), 1-15 (2022).
21. Herrenkohl, L. R., & Cornelius, L.: Investigating elementary students' scientific and historical argumentation. *Journal of the Learning Sciences 22*(3), 413-461 (2013).
22. Leech, N. L., & Onwuegbuzie, A. J.: Beyond constant comparison qualitative data analysis: Using NVivo. *School Psychology Quarterly*, *26*(1), 70-84 (2011).
23. Liew, J. S. Y., McCracken, N., Zhou, S., & Crowston, K.: Optimizing features in active machine learning for complex qualitative content analysis. In: *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pp. 44-48 (2014).
24. Linzarini, A., Budgen, S., Merkley, R., Gaab, N. R., Siegel, L., Aldersey, H., et al.: Identifying and supporting children with learning disabilities. In: Bugden, S. and Borst, G. (eds.) *Education and the Learning Experience in Reimagining Education: The International Science and Evidence based Education Assessment*. UNESCO MGIEP, New Delhi (2022).

25. Liu, L.: Using generic inductive approach in qualitative educational research: A case study analysis. *Journal of Education and Learning 5*(2), 129-135 (2016).
26. Marathe, M., & Toyama, K.: Semi-automated coding for qualitative research: A user-centered inquiry and initial prototypes. In: *CHI '18: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1-12 (2018, April).
27. Marvin, G., Hellen, N., Jjingo, D., & Nakatumba-Nabende, J.: Prompt Engineering in Large Language Models. In: *International Conference on Data Intelligence and Cognitive Informatics*, pp. 387-402. Springer Nature Singapore (2023, June).
28. Mesec, B.: The language model of artificial inteligence chatGPT - a tool of qualitative analysis of texts. *Authorea Preprint*s (2023).
29. Perrin, A. J.: The CodeRead system: Using natural language processing to automate coding of qualitative data. *Social Science Computer Review*, *19*(2), 213-220 (2001).
30. Reiss, M. V.: Testing the reliability of ChatGPT for text annotation and classification: A cautionary remark. *arXiv preprint arXiv:2304.11085* (2023).
31. Saldaña, J., & Omasta, M.: *Qualitative Research: Analyzing Life*. Sage Publications (2016).
32. Shaffer, D. W., & Ruis, A. R.: How we code. In: *Advances in Quantitative Ethnography: Second International Conference, ICQE 2020, Malibu, CA, USA, February 1-3, 2021, Proceedings 2*, pp. 62-77. Springer International Publishing (2021).
33. Strauss, A., & Corbin, J.: *Basics of qualitative research*. Sage Publications (1990).
34. Sutton, J., & Austin, Z.: Qualitative research: Data collection, analysis, and management. *The Canadian Journal of Hospital Pharmacy 68*(3), 226 (2015).
35. Tai, R. H., Bentley, L. R., Xia, X., Sitt, J. M., Fankhauser, S. C., Chicas-Mosier, A. M., & Monteith, B. G.: An Examination of the Use of Large Language Models to Aid Analysis of Textual Data. *bioRxiv*, 2023-07 (2023). https://doi.org/10.1101/2023.07.17.549361
36. Thomas, D.: A General Inductive Approach for Qualitative Data Analysis. *American Journal of Evaluation 27*(2), 237–246 (2006). https://doi.org/10.1177/1098214005283748
37. Tierney, P. J.: A qualitative analysis framework using natural language processing and graph theory. *Int'l Review of Research in Open and Distributed Learning*, *13*(5), 173-189 (2012).
38. Törnberg, P.: How to Use Large-Language Models for Text Analysis (2023).
39. Tracy, S. J.: Qualitative quality: Eight "big-tent" criteria for excellent qualitative research. *Qualitative Inquiry*, *16*(10), 837-851 (2010).
40. Weston, C., Gandell, T., Beauchamp, J., McAlpine, L., Wiseman, C., & Beauchamp, C.: Analyzing interview data: The development and evolution of a coding system. *Qualitative Sociology 24*, 381-400 (2001). https://doi.org/10.1023/A:1010690908200
41. Xiao, Z., Yuan, X., Liao, Q. V., Abdelghani, R., & Oudeyer, P. Y.: Supporting Qualitative Analysis with Large Language Models: Combining Codebook with GPT-3 for Deductive Coding. In: *Companion Proceedings of the 28th International Conference on Intelligent User Interfaces*, pp. 75-78 (2023, March). https://doi.org/10.1145/3581754.3584136
42. Yang, B., Nam, S., & Huang, Y.: "Why My Essay Received a 4?": A Natural Language Processing Based Argumentative Essay Structure Analysis. In: *International Conference on Artificial Intelligence in Education*, pp. 279-290. Springer Nature Switzerland (2023).
43. Zambrano, A. F., Liu, X., Barany, A., Baker, R. S., Kim, J., & Nasiar, N.: From nCoder to ChatGPT: From automated coding to refining human coding. In: *International Conference on Quantitative Ethnography*, pp. 470-485. Springer Nature Switzerland (2023).
44. Zhang, H., Wu, C., Xie, J., Lyu, Y., Cai, J., & Carroll, J. M.: Redefining qualitative analysis in the AI era: Utilizing ChatGPT for efficient thematic analysis. *arXiv preprint arXiv:2309.10771* (2023). https://doi.org/10.48550/arXiv.2309.10771