**Using Large Language Models for Automated Coding of Self-Regulated Learning Think-Aloud Protocol Data**

Sirui Ren, Ha Nguyen, Matthew L. Bernacki, Linyu Yu, Jeffrey A. Greene

1. **Introduction**

The alarming attrition rate of 40-60% in undergraduate Science–Technology–Engineering–Mathematics (STEM) represents a persistent educational challenge with significant economic implications despite growing workforce demands and institutional investments (Chen, 2013; National Academies of Sciences, Engineering, and Medicine, 2024). Self-regulated learning (SRL)—defined as the proactive cognitive, motivational, and behavioral processes through which learners systematically pursue academic goals using planning, monitoring, and strategic adaptation—has emerged as crucial for student persistence and success in challenging STEM curricula (Greene et al., 2024). Students who effectively employ SRL processes demonstrate substantially higher persistence rates (Park et al., 2019) and achievement (Blackmore et al., 2021). However, what makes SRL particularly complex is its fundamentally dynamic nature, with learners implementing sophisticated IF-THEN decision rules that guide evaluation of situational demands and selection of appropriate strategies. These decisions occur repeatedly, often on a second-to-second timeframe throughout task completion, illustrating the intricate and fluid nature of self-regulation (Winne & Hadwin, 1998). These conditional relationships manifest differently across tasks, academic domains, and sociocultural settings as learners respond to specific affordances and constraints embedded within learning environments (Ben-Eliyahu & Bernacki, 2015). This context-dependent and highly conditional nature of SRL necessitates methodological approaches capable of efficiently capturing its dynamic characteristics as these processes unfold within authentic learning contexts (Greene et al., 2015).

Think-aloud protocols (TAPs; Greene et al., 2018) serve as a powerful methodological tool for capturing evidence of these dynamic SRL processes as they unfold in real time, yet their strengths are counterbalanced by substantial coding resource challenges. When implemented with rigorous methodological controls—such as structured task environments, standardized participant instructions, and systematic coding schemes—TAPs yielded valid, ecologically sensitive data about learners' cognitive and metacognitive behaviors (Greene et al., 2015). Unlike retrospective self-reports, which relied on learners' imperfect memory, TAPs avoided recall bias by documenting verbalized thoughts during task performance (Winne & Perry, 2000). This concurrent approach enabled researchers to code evidence of transient SRL processes, as verbalizations revealed students' moment-to-moment monitoring and adaptive strategy shifts (Binbasaran et al., 2015). However, the richness of TAP data creates substantial resource demands that limit scalability for large-scale research. The coding process requires labor-intensive and time-consumibg steps including transcription, segmentation, and classification using theory-aligned schemes (Greene et al., 2013).

Recent methodological investigations demonstrated that Large Language Models (LLMs) can substantially reduce manual coding workload and maintain high reliability in educational research (Chernikova et al., 2024; Sailer et al., 2024; Tai et al., 2024), positioning these computational systems as promising tools for addressing the resource challenges of coding TAP data. To optimize LLM performance for specifc tasks, researchers have developed diverse prompt engineering strategies including structural reasoning techniques like chain-of-thought prompting (Kojima et al., 2022), example-based approaches such as few-shot learning (Zhao et al., 2021), role specification through expert persona adoption (Gao, 2023), and strategic code generation techniques (Weng, 2023). Experimental evidence demonstrates that the effectiveness of these prompting strategies varies considerably based on model architecture, training corpus characteristics, and task complexity (Corlatescu et al., 2024; Demszky et al., 2023), suggesting that tailored approaches may be necessary for the specific challenges of applying SRL codes to TAP data.

Despite these promising advances, current prompt engineering approaches face two critical limitations that may undermine their effectiveness for SRL coding. First, existing strategies have primarily emerged from computer science research (e.g., Brown et al., 2023; Kojima et al., 2022) without sufficient incorporation of established learning theories. This creates a disconnect where the technical methods for instructing LLMs have evolved separately from the theoretical frameworks needed to accurately identify and classify SRL processes. Without theoretical foundations explicitly incorporated into prompting strategies, LLMs may apply classifications based solely on surface-level linguistic patterns, potentially compromising the validity of SRL coding and limiting the insights that can be derived from automated approaches. Second, although LLMs demonstrate potential for automating coding tasks, current approaches are not equipped to handle the context-dependent nature that is essential to SRL processes. For example, Winne and Hadwin's (1998) model highlights that the same verbalization (e.g., "I need to read this again") can serve different regulatory functions depending on its phase and contextual conditions. This context-dependency extends further across academic domains, as research has consistently demonstrated that metacognitive monitoring and strategy use manifest differently due to variations in task structures and epistemic demands across tasks in different domains (Greene et al., 2015; Veenman et al., 2006).

To address these limitations, our approach examines how LLMs can identify self-regulatory processes that manifest at different levels of context (Figure 1). We conceptualize these levels as concentric circles of contextual influence. The innermost core represents the fundamental cognitive task level that remains consistent across STEM courses—including essential problem-solving processes, analytical reasoning, and core SRL strategies that students employ regardless of domain. For example, students across both mathematics and biology courses engage in similar core metacognitive and cognitive processes such as identifying relevant information, applying mathematical operations, and evaluating their solutions. The middle circle encompasses course-specific cognitive and metacognitive demands, reflecting how learning objectives differ across domains. In mathematics, this involves conceptual understanding and procedural fluency related to geometry and algebra, whereas in biology, this involves applying mathematical problem-solving to concrete phenomena like biodiversity and heredity case studies. These differences are significant because the learning materials that are targets for cognition and metacognition differ substantially between contexts. The outer circle represents the broader instructional design elements—including disciplinary-specific terminology (e.g., "associative property" in mathematics versus "genetic inheritance" in biology) and learning environments (e.g., high-structure active learning, flipped classroom, traditional lectures) that create different emergent learning contexts. Importantly, our study examines university course lessons with different instructional designs from different domains that share only partially overlapping task goals (e.g., completing homework assignments, solving problems during class) and design features (e.g., watching videos).

Building on this multi-level conceptual framework, the present study evaluates the capability of theory-driven prompt engineering approaches to accurately classify frequent SRL processes in TAPs across both biology and mathematics contexts. To communicate these layered contextual nuances in our LLM prompts, we drew from a comprehensive TAP codebook that has evolved through multiple research iterations across diverse learning contexts (see Bernacki et al., 2024 for the codebook details; codebook descended from Greene & Azevedo, 2009). This codebook has undergone systematic refinement as it was applied to each new task across domains, with each implementation contributing task-specific micro-level SRL codes that complement the core framework. Through this investigation, we address three research aims: (1) We examine how core task features implemented within different STEM course designs (specifically flipped mathematics and high-structure biology) influence LLM coding accuracy for six key SRL behaviors in student verbalizations, seeking to identify how task-specific elements impact automated classification of SRL processes. (2) We investigate how the effectiveness of different prompt engineering strategies varies across these task implementations and SRL codes, comparing zero-shot versus few-shot approaches and assessing whether including varying levels of contextual information (no context, task-aligned, and course-aligned) meaningfully impacts classification accuracy. (3) We analyze how the inherent nature of different SRL processes influences LLM coding accuracy across tasks, particularly comparing processes with standardized verbal expressions versus those more dependent on task-specific language. Through this systematic investigation, we aim to advance methodological approaches for SRL assessment that combine theoretical rigor with computational efficiency, ultimately enhancing our capacity to study SRL at scale across diverse educational contexts.

A diagram of a diagram

AI-generated content may be incorrect.

**Figure 1.** The nested structure of context levels in SRL

**2. Literature Review**

*2.1 Self-Regulated Learning (SRL) and Think-Aloud Protocols (TAPs)*

*2.1.1 Theoretical foundations of SRL*

SRL represents the complex interplay between cognitive, metacognitive, and motivational processes that enable learners to actively manage their learning (Winne & Hadwin, 1998; Zimmerman, 2000). At the core of SRL are cognitive and metacognitive processes that enable learners to effectively direct their learning experiences (Winne, 1995). Cognitive processes involve the actual manipulation of information and the application of learning strategies (elaboration, organization, and rehearsal). Metacognitive processes pertain to thinking about one's own thinking and learning (planning, monitoring, control, evaluation). Meta-analyses have consistently shown that effective SRL interventions substantially improve learning outcomes across educational domains and settings (Dignath et al., 2008; Shao et al., 2023; Theobald, 2021). SRL enhances students' metacognitive strategy use (Guntur & Purnomo, 2024), improves motivation and self-efficacy, and strengthens academic performance across STEM domains (Dignath et al., 2008).

SRL theory has evolved to recognize learning regulation as inherently situated within specific contexts and highly sensitive to task demands. Early foundational work by Zimmerman (2000) conceptualized SRL as cyclical, comprising forethought, performance, and self-reflection phases. In this model, learners proactively adapt their thoughts and behaviors based on environmental conditions and task requirements. This social-cognitive perspective emphasizes how learners' regulatory processes are shaped by interactions between personal, behavioral, and environmental factors specific to each learning situation.

The Metacognitive and Affective Model of Self-Regulated Learning (MASRL; Efklides, 2011, 2018) provides a detailed account of how regulatory processes operate at multiple levels during task engagement. At the *Person level*, learners bring general knowledge and beliefs about learning strategies that must be adapted to specific task demands—this includes domain knowledge, motivational beliefs, and metacognitive knowledge about strategies that exists prior to engaging with specific tasks. The *Task × Person* level captures how cognitive processing, metacognitive experiences, and affect interact during actual task completion. This interaction is critical because it represents the dynamic process through which learners experience subjective feelings (e.g., feeling of difficulty), make judgments (e.g., judgment of learning), and form online task-specific estimates of their performance that directly influence strategy selection and effort allocation. The MASRL model's distinction between \*Person-level\* factors and *Task × Person* interactions explains why identical learning behaviors may serve different regulatory functions depending on a learner's current metacognitive experiences and affective states.

Winne and Hadwin's (1998) information processing model elaborates on internal SRL processes by emphasizing how learners' regulatory decisions are governed by loosely sequential conditions-operations-products-evaluations-standards (COPES) cycles that are inherently context-dependent. Within this framework, conditions represent task parameters and cognitive resources available to the learner; operations are the cognitive processes and strategies applied; products are the outcomes of operations; evaluations involve comparing products against standards; and standards are criteria for determining success. These COPES elements interact across four distinct phases: task definition, goal setting and planning, enacting strategies, and adaptation. This model specifically distinguishes task definition as a separate phase from goal setting—a critical theoretical distinction that highlights how learners' initial interpretation of task requirements fundamentally shapes their subsequent regulatory decisions. The recursive nature of these phases, with monitoring and control processes operating throughout, explains why similar verbalizations may represent different regulatory processes depending on their phase-specific context.

*2.1.3 Task-specific manifestations of SRL processes*

Research on SRL has consistently demonstrated that although fundamental regulatory mechanisms remain consistent across educational contexts, the specific manifestations of these processes vary substantially based on task requirements and instructional designs (Alexander et al., 2011; Greene et al., 2015). This task-specific nature of SRL has profound implications for both research methodology and educational practice, especially when employing automated analysis tools across different contexts.

Early research by Veenman et al. (1997) established that metacognitive skills exhibited both general (task-independent) and specific (task-dependent) components. Their findings indicated that approximately 60% of metacognitive variance could be attributed to a general metacognitive factor, with the remaining 40% explained by task-specific manifestations, documenting these differences across mathematics problem-solving tasks versus text-based reading comprehension tasks.

In STEM education specifically, Dinsmore et al. (2008) further elaborated on these task-specific distinctions, showing how algebra equation tasks with algorithmic solution procedures elicited verbalizations focused on procedural accuracy, whereas open-ended scientific inquiry tasks involving experimental design prompted verbalizations centered on hypothesis evaluation and experimental validity. These differences affected not only which SRL processes learners deployed but also how these processes manifested in their verbalizations.

Building on this evidence from traditional learning environments, Greene and Azevedo (2009) examined hypermedia learning contexts, identifying how identical metacognitive monitoring processes manifested through markedly different verbal indicators when students completed biology conceptual tasks versus history comprehension tasks—the same monitoring function appeared as statements about biological concepts in science contexts versus statements about historical accuracy in humanities contexts. Additionally, Azevedo et al. (2005) documented how hypermedia learning tasks generated distinctive regulatory verbalizations incorporating navigation strategies and information evaluation processes that were absent in traditional textbook learning tasks.

These variations in SRL manifestations across different tasks directly impact measurement validity and instructional approaches. Veenman and Beishuizen (2004) demonstrated this impact by comparing student behaviors when studying complex biology texts under strict time constraints versus simpler texts with ample time. Their results revealed qualitatively different metacognitive strategy deployment—the more challenging condition elicited more selective note-taking and targeted summarization, in contrast to the less constrained condition that generated more comprehensive elaboration strategies. Such findings highlight how task parameters systematically influence both the frequency and quality of regulatory processes employed.

The distinction between the functional purpose of a regulatory process and its surface-level linguistic expression creates a significant challenge for automated coding approaches. Greene and Azevedo (2009) emphasized this distinction in their coding framework, noting that accurate classification of SRL processes required attention to both the underlying regulatory function (which remains consistent across contexts) and the task-specific linguistic expressions through which that function is communicated (which varies by learning environment). This nuanced relationship between general SRL mechanisms and their task-specific manifestations becomes particularly important when developing \*prompt engineering\* approaches for LLMs that must identify comparable regulatory processes across diverse educational settings despite variations in their verbal expression.

*2.1.4 TAPs as a methodology*

TAPs have emerged as a particularly valuable method for capturing the dynamic and context-sensitive nature of SRL processes. This methodology involves participants verbalizing their thoughts during task completion, providing researchers with direct access to ongoing cognitive and metacognitive processes that would otherwise remain unobservable (Ericsson & Simon, 1993; Fox et al., 2011). TAPs are grounded in information processing theory, which posits that only information in working memory can be verbalized during concurrent think-aloud procedures (Ericsson & Simon, 1993). This theoretical foundation makes TAPs particularly suitable for SRL research because they capture evidence of regulatory processes as they occur rather than relying on memory-based reconstructions. The methodology operates on the principle that verbalizations during task performance provide insights into the cognitive and metacognitive operations learners employ, revealing the sequencing, adaptations, and contextual triggers that shape regulatory decisions in authentic learning situations.

Specifically, TAPs capture real-time processing, eliminating the recall biases inherent in retrospective measures (Greene et al., 2018; Winne & Perry, 2000). Furthermore, they provide rich contextual data about how regulatory processes interact with specific task features and environmental conditions. Additionally, they reveal the sequential patterns of regulatory decisions as they unfold naturally during learning activities. These characteristics make TAPs especially well-suited for examining the dynamic, context-dependent nature of SRL emphasized in theoretical models (Ackerman & Thompson, 2017; Greene et al., 2018).

Despite these strengths, TAPs present substantial coding challenges. The verbal protocols generated through this methodology typically require labor-intensive processing, including transcription, segmentation, and classification using theory-aligned coding schemes (Greene et al., 2013). This process necessitates multiple trained coders who must establish reliability before analyzing the full dataset. The time-consuming nature of traditional TAP coding creates a considerable bottleneck for SRL research, limiting sample sizes and hindering researchers' ability to examine regulatory processes across diverse educational contexts (Azevedo et al., 2013; Greene et al., 2018; Messick, 2018). These limitations have motivated interest in more efficient approaches to TAP coding that maintain methodological rigor and increase scalability.

*2.2 Automated Coding of Qualitative Data using LLMs*

Recent advances in LLMs offer promising approaches for addressing the coding challenges associated with TAP data. These models have demonstrated capabilities for automating various coding tasks across educational research domains (Chernikova et al., 2024; Sailer et al., 2024; Tai et al., 2024). LLMs can process natural language data at scale and maintain reasonable levels of accuracy compared to human coders, potentially transforming labor-intensive manual coding workflows (Demszky et al., 2023; Labutov et al., 2022; Syed et al., 2023).

The task-specific nature of SRL processes described earlier presents both unique challenges and opportunities for LLM-based coding approaches. Traditional manual coding methods required human coders to maintain extensive knowledge of both theoretical SRL frameworks and task-specific contexts to accurately interpret learners' verbalizations (Binbasaran & Greene, 2015; Messick, 2018). This dual expertise requirement contributed considerably to the labor-intensive nature of SRL research. LLMs address this challenge through their processing capacity for large volumes of contextual information, pattern detection capabilities, and classification rule application—functionalities that align with the multi-layered interpretative demands of SRL coding across diverse task contexts (Zhao et al., 2023). However, realizing this potential requires careful consideration of how LLM architectures and \*prompt engineering\* strategies can be configured to identify the functional equivalence of regulatory processes despite their varying linguistic manifestations.

Modern LLMs like GPT models (OpenAI, 2024), Claude models (Anthropic, 2024), and Gemini models (Google et al., 2024) are built upon the Transformer architecture, first introduced by Vaswani et al. (2017) in their seminal paper "Attention Is All You Need." This architecture represents a fundamental shift from previous sequential processing approaches to a parallelized attention-based design that has transformed natural language processing. The key innovation of the Transformer is its attention mechanism, which computes weighted relationships between all elements in a sequence simultaneously, enabling the system to focus on relevant information regardless of positional distance (Vaswani et al., 2017). Unlike previous recurrent neural networks that processed words sequentially, transformers compute attention scores for all word pairs in parallel, creating a rich representational capacity that captures syntactic dependencies, semantic relationships, and contextual nuances (Devlin et al., 2019; Lin et al., 2022).

This architectural design is particularly applicable to SRL research because of the complex, context-dependent nature of regulatory processes (Greene et al., 2015; Winne & Hadwin, 1998). For example, determining whether a student statement like "I need to try a different approach" represents metacognitive monitoring, strategy adaptation, or motivation regulation depends critically on surrounding context that may span across multiple utterances in TAP data (Azevedo et al., 2018; Bernacki et al., 2024). The self-attention mechanism in transformer models processes these long-range dependencies effectively (Khan et al., 2022; Vaswani et al., 2017), as it calculates relevance weights for all parts of the dialogue when processing any single utterance. For instance, a comment about adjusting strategy can only be properly classified as metacognitive regulation by considering earlier expressions of goal-setting or task definition that may appear several statements earlier in the transcript (Zhao et al., 2023). These technical capabilities address the core challenge in SRL coding: differentiating between superficially similar statements that serve different regulatory functions based on their broader context.

We selected models with robust transformer architectures for this study based on their documented performance in complex reasoning tasks that require contextual understanding (Anthropic, 2023; Brown et al., 2020; OpenAI, 2023). These models incorporate extensive parameter counts (typically ranging from tens to hundreds of billions of parameters) and diverse pre-training datasets that include scientific literature, educational content, and analytical writing. This pre-training enables pattern recognition in language that corresponds to theoretical constructs in educational research, even without specific training on SRL coding schemas (Chowdhery et al., 2022; Wei et al., 2022).

The effectiveness of these models for coding depends heavily on prompt engineering—the systematic design of input instructions that optimize performance for specific analytical tasks (Liu et al., 2023; White et al., 2023). Several promising approaches have emerged that align with the specific needs of SRL coding: Chain-of-thought prompting guides LLMs through step-by-step analytical processes that mirror expert coding decisions, breaking down the complex task of identifying regulatory processes into smaller reasoning steps (Kojima et al., 2022; Wei et al., 2022). This approach is particularly valuable for SRL coding as it can explicitly incorporate theoretical distinctions between cognitive and metacognitive processes. Few-shot learning provides concrete examples of desired coding decisions, demonstrating how theoretical constructs manifest in actual student verbalizations (Brown et al., 2020; Min et al., 2022). This technique leverages the model's ability to recognize patterns across similar examples, potentially addressing the challenge of identifying SRL processes that manifest differently across learning contexts. Role specification through expert persona adoption positions the model to emulate the knowledge and analytical approach of SRL researchers (Shanahan, 2022; White et al., 2023).

Importantly, the relationship between task specificity and coding performance is complex, particularly in the context of SRL. Recent empirical work by Zhang et al. (2024) illustrates both the potential and limitations of automating SRL detection in educational settings using LLMs. They experimented with two specific embedding approaches for coding SRL think-aloud transcripts: the Universal Sentence Encoder (USE) and OpenAI's text-embedding-3-small model. When comparing these embedding approaches for coding SRL behaviors in student verbalizations from different domains (chemistry to logic), they found that although these approaches effectively detected general SRL behaviors, they encountered difficulties with task-specific vocabulary. For instance, formal logic terms like "associativity" or "DeMorgan rule" created challenges for models trained on chemistry data. Notably, the "Realizing Errors" category showed the strongest transfer effects due to its consistently agnostic indicators like "wrong" or "incorrect." These findings highlight a critical consideration when analyzing SRL behaviors: regulatory processes manifest differently across domains due to variations in task structures and epistemic demands (Greene et al., 2015; Veenman et al., 2006). This evidence underscores the importance of developing approaches that can account for task-specific characteristics and maintain theoretical alignment.

The convergence of research on task-specific SRL manifestations and LLM capabilities raises several important questions that remain underexplored in current literature. First, the extent to which different prompt engineering strategies can address the task-specific nature of SRL processes has not been systematically investigated. Second, the differential performance across various types of SRL processes—some with consistent linguistic markers versus others with more task-dependent expressions—requires further examination. Third, the interaction between contextual information provided in prompts and the accuracy in identifying functionally equivalent SRL processes across different task implementations demands empirical investigation. Addressing these questions would substantially advance our understanding of how automated coding approaches can be optimized to handle the complex, context-dependent nature of SRL and maintain theoretical integrity.

*2.3 Purpose and Research Questions*

The literature review highlights several critical gaps in current approaches to automated SRL coding. First, although LLMs show promise for thematic coding tasks, their application to the complex, context-dependent nature of SRL processes remains underexplored. Second, existing prompt engineering strategies have developed largely without incorporation of established learning theories, creating a disconnect between the technical methods for instructing LLMs and the theoretical knowledge needed to accurately classify SRL processes. Third, the interaction between task-specific elements and SRL process types has not been systematically investigated within automated coding approaches.

This study addresses these gaps by examining how theory-driven prompt engineering approaches can enhance the accuracy of SRL process coding in TAPs across different learning contexts. Through systematic evaluation of these factors, we aim to develop more effective approaches for automated coding of SRL processes that maintain theoretical rigor and increase scalability.

**RQ1: How do core task features implemented within different STEM course designs influence LLM coding accuracy for SRL processes?**

This question examines how the distinctive characteristics of flipped mathematics and high-structure biology learning contexts affect coding accuracy for six key SRL behaviors in student verbalizations. By comparing performance across these different learning contexts, we can identify how task-specific elements influence automated coding of SRL processes.

**RQ2: How does the effectiveness of different prompt engineering strategies vary across task implementations and SRL codes?**

This question investigates: (a) the comparative performance of zero-shot versus few-shot approaches across different task structures and SRL codes, and (b) whether including varying levels of contextual information (no context, task-aligned, and course-aligned) meaningfully impacts classification accuracy. These comparisons help identify optimal prompting approaches for different educational contexts.

**RQ3: How does the nature of different SRL processes influence LLM coding accuracy across tasks?**

This question explores whether coding accuracy differs between SRL processes with consistent linguistic patterns across tasks (e.g., expressions of understanding or learning progress) compared to processes that rely more heavily on task-specific terminology (e.g., forming conclusions about task-specific concepts). We examine whether certain combinations of prompting strategies can optimize coding accuracy for different types of regulatory behaviors, particularly comparing processes with standardized verbal expressions versus those more dependent on task-specific language.

**3. Method**

*3.1 Participants and Data Collection*

*3.1.1 Research Setting and Data Collection Procedures*

The data for this study was obtained from the TUSSLER project, an NSF-funded research initiative examining SRL processes in undergraduate STEM courses. Participants included 49 undergraduate students enrolled in Introductory Biology and 49 undergraduate students in Pre-calculus courses. Each participant completed 90-minute learning sessions focusing on either Biodiversity (Biology) or Ellipse (Mathematics) content in a controlled laboratory setting. During these sessions, participants verbalized their thoughts and actions when engaging in structured learning activities, including video lectures, homework problems, and quizzes. These activities reflected the highly structured instructional designs that faculty implemented in their actual courses (see Figure 2).

*A diagram of a diagram

AI-generated content may be incorrect.*

**Figure 2. Structured activities for Biodiversity (Biology) and Ellipse (Mathematics) in a laboratory setting**

*3.1.2 Codebook Development and Human Coding Procedures*

Participants' verbalizations were audio- and video-recorded during the sessions. The audio recordings were transcribed by Rev.com and subsequently coded by research assistants using NVivo, following an SRL coding framework based on Information Processing Theory (Winne & Hadwin, 1998, 2008) and macro and microTAP coding framework (Greene & Azevedo, 2009). The codebook included over 50 micro-level codes nested within five macro-level categories: Task Definition and Planning (TASK), Monitoring, Domain-General Strategies (DGS), Domain-Specific Strategies (DSS), and Assessment Strategies (ASSESS). Each code incorporated detailed operational definitions, sample excerpts, and distinguishing features from similar codes, and had undergone refinement through several iterations incorporating coder feedback and adjustments. Across participants, a total of 6,591 verbalizations were coded for the biology task and 3,800 for the mathematics task.

Coding reliability was assessed using percentage agreement. For the biology task, reliability was calculated by phase, yielding reliability scores of .74 for the pre-class phase, .89 for the in-class phase, and .81 for the after-class phase. Mathematics task coding reliability was calculated as a whole, with an overall agreement of .74. According to Hallgren (2012), these reliability measures indicated substantial agreement. After establishing reliability, the final coding files were designated as the ground truth for evaluating the performance of computational approaches to coding.

**A diagram of a process

AI-generated content may be incorrect.**

**Figure 3. Research Design and Procedures**

*3.2 Research Design*

*3.2.1 Experimental Design and Variables*

The study employed a factorial design to evaluate multiple dimensions of prompt engineering for automated coding of TAPs data. Specifically, we examined a 6 × 2 × 3 × 6 factorial design, representing TAP codes × tasks × LLMs × prompt conditions. This design enabled systematic evaluation of how different prompt engineering strategies affected coding accuracy across different SRL processes and educational contexts.

The six micro-level codes selected for this study represented distinct cognitive and metacognitive processes essential to SRL theory(**see Table 1**), as described in the Literature Review section. These codes were chosen based on their theoretical significance and prevalence in our dataset. At the macro level of Task Definition and Planning, the micro-level Setting Goals for sub-target (SG) code identified verbalizations where students articulated specific targets for their learning process. Within the macro category of Strategies, two distinct types of cognitive processes were coded: Mathematical Problem Solving (MPS), which captured instances where students applied mathematical operations or procedures to work toward solutions, and Forming New Conclusions (FNC), which identified where students drew inferences or generated new understanding from available information. The macro category of Monitoring contained micro-level codes that distinguished between positive judgments of learning (JOU+), indicating students' perceived understanding, and negative judgments of learning (JOU-), signaling perceived confusion or knowledge gaps—these verbalizations provided evidence of students' real-time assessment of their comprehension. Finally, within the macro category of Assessment, the Ruling Out Answers (ROA) micro-code identified instances where students systematically eliminated incorrect solutions, representing a critical evaluation process in problem-solving activities.

For each code and task, we randomly sampled 50 utterances (25 positive examples of the code, 25 negative examples) from the human-coded TAPs dataset, yielding 600 utterances for evaluating binary classification accuracy. This sampling approach ensured balanced representation of each code, enabling fair comparisons across the different experimental conditions.

**Table 1: Selected TAP codes definition and examples**

|  |  |  |
| --- | --- | --- |
| **Macro TAP code** | **Micro TAP code** | **Definition** |
| Task Definition/Planning | SG | Sub-Goal:target (SG): Learner articulates a specific sub-goal that is relevant to the task. Must immediately carry out some action relevant to the sub-goal (i.e., can't drop the goal immediately after verbalizing). |
| Domain-specific Strategy | MPS | Mathematical Problem-Solving (MPS): Student actively working through a mathematical problem, showing steps or calculations. |
| Assessment | ROA | Ruling Out Answers (ROA): Reviewing answer choices on a multiple-choice question and systematically ruling out answer choices in order to narrow down to the answer they will select. |
| Domain General Strategy | FNC | Forming New Conclusion (FNC): Putting together two pieces of information and drawing a new conclusion that extends beyond what is presented in the learning environment. |
| Monitoring | JOU+ | Judgment of Understanding (JOU): Learner recognizes that they do (JOU+) or do not (JOU-) understand content related to the learning task. |
| Monitoring | JOU- | Judgment of Understanding (JOU): Learner recognizes that they do (JOU+) or do not (JOU-) understand content related to the learning task. |

*3.2.2 LLM Implementation and Prompt Engineering Conditions*

The study investigated three contemporary LLMs released in 2024: OpenAI GPT-4o, Anthropic Claude 3.5 Sonnet, and Google Gemini Pro. These models were selected based on their demonstrated capabilities in natural language understanding and generation tasks (Huang et al., 2024; OpenAI, 2024; Anthropic, 2024; Google et al., 2024). To implement deductive coding of TAPs data using these LLMs, we developed a systematic \*prompt engineering\* approach grounded in recent instruction techniques (Brown et al., 2023; Wei et al., 2022). Each prompt followed a standardized structure that included explicit role definition, detailed task framing, and structured reasoning steps (Liu et al., 2023). The base prompt (illustrated in Figure 3) established the model's role (Deshpande et al., 2023) as an educational psychology expert and provided specific TAP code definitions, followed by a five-step reasoning framework (Lee et al., 2024) that guided the coding process through: 1. Identifying key utterance elements; 2. Assessing alignment with code definitions and examples (Hou et al., 2024); 3. Considering previous utterances and task contextual information (Rao et al., 2023); 4. Evaluating classification confidence; 5. Making final binary classifications.

A close-up of a text

AI-generated content may be incorrect.

**Figure 4. Base Prompt Structure. All prompts begin with a core structure containing the structured elements**

We implemented six distinct prompt conditions (**see Table 2**) that systematically varied component inclusion based on established prompt engineering. We implemented six distinct prompt conditions based on established prompt engineering principles. The Zero-shot with Chain-of-Thought reasoning and Coding Rubric (ZS-CoT-CR) condition included only basic reasoning structure and specific coding criteria without additional context or examples. Zero-shot with Task-Aligned context (ZS-CoT-CR-TA) enhanced the basic prompt by incorporating information about the specific learning task such as Biodiversity for biology or Ellipse for mathematics. Zero-shot with Course-Aligned context (ZS-CoT-CR-CA) further enhanced the prompt with both task information and broader course-specific design elements, including instructional approaches like high-structure active learning for biology and flipped classroom for mathematics. Few-shot with basic structure (FS-CoT-CR) supplemented the basic zero-shot prompt with one positive and one negative example for the targeted SRL code. Few-shot with Task-Aligned context (FS-CoT-CR-TA) combined examples with task-specific contextual information. The most comprehensive condition, Few-shot with Course-Aligned context (FS-CoT-CR-CA), included examples, chain-of-thought reasoning, coding criteria, and both task and course-aligned contextual information. Our preliminary analyses (detailed in the Appendix) demonstrated that a five-utterance contextual window achieved optimal performance when compared with alternative window sizes (0, 1, 3, 5, and 10 utterances). This finding indicated that providing the LLMs with the 5 most relevant previous TAP utterances from the student produced the most accurate coding results at the same time balancing efficiency and context retention.

**Table 2. *Prompt conditions and components***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Chain of thought (CoT) | Rubric (CR) | Few-shot (FS) | Task Aligned (TA) | Course-Aligned (CA) |
| ZS-CoT-CR: Basic prompt including chain-of-thought reasoning and coding criteria. | x | x |  |  |  |
| ZS-CoT-CR-TA: Enhanced the basic prompt with task-specific contextual information. | x | x |  | x |  |
| ZS-CoT-CR-CA: Further enhanced with both task and course-specific design with broader instructional features | x | x |  | x | x |
| FS-CoT-CR: Basic prompt supplemented with one positive and one negative example. | x | x | x |  |  |
| FS-CoT-CR-TA: Few-shot prompt enhanced with task-specific contextual information. | x | x | x | x |  |
| FS-CoT-CR-CA: Comprehensive prompt including examples and both task and course-specific design with broader instructional features | x | x | x | x | x |

*3.2.3 Analytical Strategies*

For performance evaluation, we used accuracy as the primary metric with complementary analysis of confusion matrices to examine classification patterns (detailed precision, recall, and F1 scores appear in the Appendix). Confusion matrices were particularly valuable for this analysis as they provided a complete picture of classification performance beyond overall accuracy, revealing specific patterns of true positives, true negatives, false positives, and false negatives across different conditions. This detailed error analysis helped identify whether LLMs exhibited systematic biases toward certain types of classifications and whether these patterns varied across different SRL processes or tasks.

The analytical approach aligned with the research questions presented earlier: To address RQ1 (task effects), we conducted independent samples t-tests examining task differences (Mathematics vs. Biology) in classification accuracy, one-way ANOVAs examining differences across TAP codes (SG, MPS, FNC, JOU+, JOU-, ROA), and pairwise comparisons between LLMs (GPT-4o vs. Claude 3.5 Sonnet vs. Gemini Pro). For RQ2 (prompt engineering effects), we performed independent samples t-tests comparing zero-shot versus few-shot performance for each TAP code within each task and conducted one-way ANOVAs to assess the impact of three contextual conditions (no context, task-aligned, course-aligned) on coding accuracy. For RQ3 (code-specific patterns), we conducted independent samples t-tests examining task differences for each individual TAP code and analyzed interaction patterns between prompt conditions, tasks, and TAP codes.

**4. Results**

*4.1 Overall model performance metrics*

The performance of LLMs in classifying SRL behaviors exhibited moderate to high accuracy across conditions, ranging from 49% to 90% accuracy. Analysis of the classification performance revealed several notable patterns in accuracy across different conditions and tasks within the mathematics and biology.

*4.2 Tasks Effect , TAPs and LLMs effect (RQ1)*

A substantial main effect of tasks emerged in the analysis, with mathematics tasks demonstrating significantly higher classification accuracy (*M* = 0.78, *SD* = 0.41) compared to biology tasks (*M* = 0.56, *SD* = 0.50), *t* (10798) = 24.73, *p* < .001)(**Figure 5**). This marked difference suggests that SRL behaviors may be more distinctly identifiable in mathematical problem-solving contexts, possibly due to the more structured nature of mathematical reasoning processes.

Significant variations were observed across different TAP codes (**Figure 6**), *F*(5, 10794) = 19.53, *p* < .001, *η²* = 0.009,MPS demonstrated the highest accuracy (*M* = 0.73, *SD* = 0.44) and followed by ROA (*M* = 0.72, *SD* = 0.45), whereas SG showed lower accuracy (*M* = 0.61, *SD* = 0.49). Notably, JOU+ showed relatively lower accuracy compared to JOU- (JOU-: *M* = 0.69, *SD* = 0.46, JOU+: *M* = 0.66, *SD* = 0.47), suggesting potential differences in the models' ability to distinguish between different valence of metacognitive judgments.

A graph showing a number of tasks

AI-generated content may be incorrect.

**Figure 5. Main effect of tasks on LLMs’ coding accuracy**

A graph of a bar chart

AI-generated content may be incorrect.

**Figure 6. Main effect of TAP code types on LLMs’ coding accuracy (ranked by accuracy of the TAP code)**

LLMs comparison in accuracy showed GPT-4o (*M* = .69) and Claude (*M* = .70) performed similarly (*t* = 0.823, *p* = .411), while Gemini Pro showed significantly lower accuracy (*M* = .62, *t* = -5.111, *p* < .001; **Figure 7**). Analysis of the confusion matrices revealed distinct classification patterns across LLMs (**Figure 8**). GPT-4o and Claude 3.5 demonstrated similar conservative prediction patterns, with higher true negative rates (40.5% and 41.9% respectively) compared to Gemini Pro (32.3%). In comparison, Gemini Pro showed a notably higher false positive rate (17.7%) compared to both GPT-4o (9.5%) and Claude 3.5 (8.1%).

A graph of a number of red rectangular bars

AI-generated content may be incorrect.

**Figure 7. LLMs main effect in coding accuracy**

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 8. Confusion matrix in LLMs coding performance**

*4.3 Effects of prompt engineering strategies(RQ2)*

*4.3.1 Zero-Shot vs Few-Shot performance across tasks*

The impact of few-shot learning varied by tasks and TAP code (**Figure 9**). In mathematics TAPs, few-shot prompting showed modest improvements for most codes, with JOU- demonstrating the only statistically significant enhancement (*ΔM* = 0.051, *t*(898) = -2.18, *p* = .030). Interestingly, SG in mathematics showed a significant negative effect with few-shot prompting (*ΔM* = -0.080, *t*(898)= 2.66, *p* = .008). In biology tasks, few-shot effects were generally minimal and non-significant across all TAP codes, with slight improvements in FNC (*ΔM* = 0.047, *t* (898)= -1.41, *p* = .160) being the largest observed difference. This pattern suggests that the utility of few-shot learning may be task-dependent, with greater potential benefits in mathematical contexts.

A comparison of a bar graph

AI-generated content may be incorrect.

**Figure 9. FS vs ZS prompt conditions across TAPs code and tasks in LLMs coding performance**

*4.3.2. Context effects across tasks*

Analysis of context effects (None, Task-aligned, and Course-Aligned) revealed no statistically significant differences across both domains and all TAP codes **(Figure 9**). In the Biology tasks, context effects were consistently non-significant across all TAP codes. SG showed no meaningful difference between context conditions, *F*(2, 297) = 0.08, *p* = .92. Similarly, ROA demonstrated negligible variation across context types, *F*(2, 297) = 0.05, *p* = .95. The pattern continued with JOU+, which showed no significant context effect, *F*(2, 297) = 0.05, *p* = .95. JOU- also revealed no meaningful differences between context conditions, *F*(2, 297) = 0.23, *p* = .79. MPS maintained consistent performance regardless of context, *F*(2, 297) = 0.03, *p* = .97. Finally, FNC showed no significant variation across context types, *F*(2, 297) = 0.06, *p* = .94.

In the Mathematics task, though overall performance was higher, context effects remained non-significant across all TAP codes. SG showed no significant difference between context conditions, *F* (2, 297) = 0.70, *p* = .50. ROA demonstrated the largest non-significant variation, but still fell well short of statistical significance, F(2, 297) = 1.03, *p* = .36. JOU+ revealed no meaningful context effect, *F*(2, 297) = 0.24, *p* = .79. Similarly, JOU- showed remarkably consistent performance across contexts, *F*(2, 297) = 0.12, *p* = .89. MPS maintained stable performance regardless of context type, *F*(2, 297) = 0.74, *p* = .48. Finally, FNC demonstrated no significant variation based on context level, *F*(2, 297) = 0.42, *p* = .66.

A graph of different colored bars

AI-generated content may be incorrect.

**Figure 10. Different context level prompting across TAPs code and tasks in LLMs coding performance**

*4.4 TAP Code-Specific Performance (RQ3)*

*4.4.1 Task and TAP Code Interactions*

The analysis revealed significant task differences across all TAP codes, with consistently higher accuracy in mathematics compared to biology (Figure 10). The most pronounced difference was observed in JOU-, where mathematics tasks performance (*M* = 0.85, *SD*= 0.35) was markedly higher than biology (*M* = 0.53, *SD* = 0.50), *t*(1798) = -15.97, *p* < .001). Similarly, the ROA code showed a pronounced task difference with mathematics (*M* = 0.86, *SD* = 0.34) outperforming biology (*M* = 0.57, *SD* = 0.50), *t*(1798) = -14.82, *p*< .001). The JOU+ code exhibited a smaller but significant difference between mathematics (*M* = 0.763, *SD*= 0.43) and biology (*M*= 0.56, *SD* = 0.50), *t*(1798) = -9.48, *p* < .001). This pattern held for SG (mathematics: *M* = 0.71, *SD* = 0.45; biology: *M* = 0.52, *SD* = 0.50; *t*(1798) = -8.80, *p* < .001), FNC (mathematics: *M*= 0.69, *SD* = 0.46; biology: *M* = 0.55, *SD* = 0.50; *t*(1798) = -6.28, *p* < .001). Interestingly, MPS showed the smallest difference across mathematics and biology (mathematics: *M* = 0.80, *SD* = 0.40; biology: *M* = 0.66, *SD* = 0.47; *t*(1798) = -6.49, *p* < .001).

A graph of numbers and lines

AI-generated content may be incorrect.

**Figure 11. Task differences across TAP codes in LLMs coding performance**

*4.4.2 Prompt, Task and TAP Code Interactions*

Analysis of individual TAP codes revealed distinct classification patterns across prompt conditions and tasks (Figure 11). In Mathematics, ROA demonstrated consistently high performance across all conditions (ranging from 0.83 to 0.90), with peak accuracy under FS-CA prompting (0.90). MPS also showed strong and stable performance in Mathematics (0.74-0.83), particularly excelling with ZS prompts. Notably, FNC remained the most challenging code in Mathematics (0.66-0.71), showing minimal variation across prompt conditions.

In Biology, MPS emerged as the most reliably classified code (0.65-0.68), maintaining relatively stable performance across all prompt conditions. SG proved most challenging in Biology (0.49-0.52), with particularly low performance under FS conditions.

Further analysis revealed that task differences persisted across various prompt conditions, though the magnitude varied by TAP code and condition. The most pronounced differences were consistently observed in JOU- and ROA codes, regardless of prompt engineering approaches. Notably, the MPS code demonstrated the highest overall performance in Biology tasks (M = 0.662), while maintaining strong performance in Mathematics (M = 0.797). In contrast, SG showed the lowest performance in Biology contexts (M = 0.516). These findings suggest that the identification of certain TAP codes is inherently more robust in mathematical problem-solving contexts compared to biological reasoning tasks, possibly due to the more structured nature of mathematical thinking processes or the more explicit manifestation of specific TAP behaviors within mathematical reasoning. The consistent pattern across all codes indicates a substantive domain effect that transcends specific prompt engineering strategies.

A screenshot of a graph

AI-generated content may be incorrect.

**Figure 12. LLMs coding accrucy across prompt condition, tasks and TAP codes**

**5. Discussion**

*5.1 Synthesis of Findings*

Our comprehensive analysis of LLMs' performance in coding SRL processes in think-aloud protocols reveals several significant patterns that extend current understanding of automated coding in educational contexts. The results demonstrated a consistent task effect across all SRL codes, with mathematical problem-solving tasks yielding higher classification accuracy (M = 0.78) compared to biology-focused tasks (M = 0.56). This systematic difference aligns with Greene et al.'s (2015) observation that task structures fundamentally shape how SRL processes manifest in student verbalizations, creating distinct classification challenges across different learning contexts.

The differential performance between GPT-4o and Claude 3.5 Sonnet (M = 0.69 and M = 0.70 respectively) compared to Gemini Pro (M = 0.62) extends recent findings by Huang et al. (2024), who documented similar performance patterns in complex reasoning tasks. Our confusion matrix analysis revealed distinct error patterns that contribute to the emerging literature on LLM behavior in educational applications (cf. Tai et al., 2024; Zhao et al., 2023). The more conservative prediction patterns exhibited by GPT-4o and Claude 3.5 Sonnet compared to Gemini Pro's higher false positive rate suggests that architectural differences influence not just overall accuracy but the specific mechanisms of classification errors—a distinction with important implications for how researchers select LLMs for specific educational coding tasks.

A particularly intriguing finding emerged in the relative stability of MPS classification across tasks, showing the smallest cross-task performance gap (ΔM = 0.13) among all codes examined. This stability is notable in light of Veenman et al.'s (2006) research on metacognitive skillfulness, which documented considerable variation in how metacognitive skills manifest across different task types. Our findings suggest that certain cognitive processes may be more consistent across contexts than previously recognized, aligning with Zimmerman's (2000) social-cognitive framework that emphasized how foundational regulatory processes maintain structural consistency despite contextual variations. In contrast, metacognitive processes like JOU- showed dramatic task-dependent variations (ΔM = 0.32), supporting Efklides' (2011) MASRL model premise that metacognitive experiences are highly sensitive to \*Task × Person\* interactions.

The differential effectiveness of prompt engineering strategies across tasks extends recent theoretical work by Liu et al. (2023) and Wei et al. (2022) on the context-sensitivity of LLM performance. Few-shot learning demonstrated modest but significant improvements primarily in mathematics-focused tasks, particularly for monitoring processes (JOU+, JOU-), but showed minimal impact in biology-focused tasks. This pattern aligns with Brown et al.'s (2020) demonstration that few-shot learning benefits are contingent on the structural alignment between provided examples and target tasks. Surprisingly, the addition of contextual information (task-aligned and course-aligned) showed no significant impact on classification accuracy across both tasks. This finding adds nuance to Rao et al.'s (2023) work on contextual prompting, suggesting that although contextual information can be valuable for certain applications, its benefits may be less pronounced for specific SRL coding tasks than previously hypothesized.

*5.2 Task-Specific Effects on LLM Performance: Differentiating Cognitive and Metacognitive Processes*

Building on the observed task effects, our analysis revealed distinct patterns in how cognitive versus metacognitive processes were classified across learning contexts. These findings provide deeper insights into why mathematical problem-solving tasks yielded more accurate classifications than biology tasks.

In mathematics tasks, where problem-solving follows more structured procedures (Nathan & Koedinger, 2000), LLMs demonstrated robust classification performance across multiple SRL behaviors. Mathematical problem-solving discourse tends to follow standardized patterns with precise procedural language (e.g., "So C and D are the only ones that work"), creating verbalizations that are particularly amenable to computational analysis (Demszky et al., 2023; Huang et al., 2024).

In contrast, biology tasks presented greater challenges for automated classification, as students frequently combined multiple cognitive processes within single statements. Utterances like "No this doesn't seem right because natural selection would take longer" integrated problem-solving assessment with conceptual application, creating multi-layered verbalizations that aligned with Greene and Azevedo's (2009) observation that problem-solving in concept-rich contexts involves more complex verbal expressions that reflect multiple regulatory processes simultaneously.

An unexpected and particularly insightful finding emerged in the relative stability of Mathematical Cognitive Strategy (MPS) classification across both contexts. Despite the overall performance differences between tasks, MPS showed the smallest cross-task performance gap (ΔM = 0.13) among all codes analyzed. This stability contrasted sharply with metacognitive codes like JOU- that showed dramatic task-dependent variations (ΔM = 0.32).

This pattern can be explained by examining the actual utterances in our dataset. In both tasks, students demonstrated similar problem-solving linguistic patterns focused on systematic calculation and step-by-step reasoning. For example, in mathematics, students verbalized calculations with explicit steps: "I'm going to go ahead and pull the nine out. So y squared plus two y equals negative nine." Similarly in biology, despite different content, students used parallel computational structures: "P is .6 and Q is .4. So P-squared is .6 square plus 2 times .6 times .4 plus .4 squared equals one."

These findings align with Anderson's (1982) Theory of Cognitive Architecture *ACT*\* theory of skill acquisition, which distinguishes between procedural knowledge (how to execute cognitive operations) and declarative knowledge (factual information specific to different fields). The computational patterns observed in our data support this theoretical distinction, explaining why LLMs maintained relatively consistent performance for problem-solving processes despite surface-level differences in content vocabulary.

This stability in core cognitive process classification stands in marked contrast to the task-dependent variation observed in metacognitive processes. Our findings extend Winne and Hadwin's (1998) COPES model by illustrating how task characteristics (conditions) influence both the cognitive strategies employed (operations) and their verbal expression (products). In mathematics tasks, conditions focus students' operations on procedural steps with explicit computational markers, whereas biology tasks' conditions lead to operations that blend conceptual understanding with mathematical reasoning, creating more complex verbalization patterns.

Prior research by Lee et al. (2024) on prompt engineering strategies suggested that task-specific strategies like MPS would require heavily contextualized prompts with specialized terminology. However, our dataset revealed that despite surface-level differences in content vocabulary (e.g., "ellipse," "vertices" in mathematics versus "alleles," "frequency" in biology), the underlying problem-solving discourse patterns remained remarkably consistent. This insight challenges assumptions about the necessity of task-specific \*prompt engineering\* for all types of SRL behaviors and suggests that some cognitive processes may have more universal verbal expressions than previously theorized.

*5.4 Tasks-Specific Metacognitive Expression*

The most pronounced task-dependent performance differences emerged in metacognitive monitoring processes, particularly in JOU-. Here, LLM classification achieved notably higher accuracy in mathematics (M = 0.85) compared to biology (M = 0.53). This substantial difference (ΔM = 0.32) illustrates the task-specific nature of metacognitive verbalization patterns, supporting a key theoretical premise in Efklides' (2011) MASRL model that metacognitive experiences are highly sensitive to \*Task × Person\* interactions. As Efklides emphasized, subjective experiences like \*feeling of difficulty\* emerge from the interaction between person-level factors and specific task characteristics, explaining why similar metacognitive functions manifest differently across contexts.

Veenman et al. (2006) demonstrated through their empirical research that metacognitive processes are task-evoked rather than domain-general, requiring "grain-specific analyses" (p. 7) in metacognitive research. Our findings provide empirical support for this theoretical position, demonstrating how the same metacognitive function (JOU-) manifests through distinctly different verbalization patterns across tasks. This variation created significant challenges for consistent classification of metacognitive processes across different task contexts.

Analysis of confusion matrices reveals the underlying patterns driving these differences. Taking the Claude 3.5 Sonnet model with the FS-CoT-CR condition as an example, mathematics classification showed balanced outcomes (True Positive: 24, True Negative: 22) with minimal errors (False Negative: 1, False Positive: 3). Students' mathematics utterances contained explicit phrases that signaled uncertainty about computational steps, such as "I'm confused," "I have no idea," "I don't know," or "Not sure." These unambiguous lexical markers align with what Efklides (2011) described as "\*feeling of difficulty\*" indicators—clear metacognitive experiences that emerge during mathematical problem-solving.

In contrast, biology classifications showed substantial asymmetry (True Positive: 5, True Negative: 21, False Negative: 20, False Positive: 4). This pattern indicates a systematic difficulty in identifying JOU- in biology contexts. Students' monitoring processes in biology tasks manifested through nuanced expressions combining task-specific knowledge with varying degrees of uncertainty. Utterances like "I'm guessing from the example they have here, that P is the dominant allele" and "microevolution would be change, like a small amount of change" intertwined tentative reasoning with task-specific terminology, requiring simultaneous analysis of conceptual coherence and metacognitive stance. This complexity mirrors Veenman et al.'s (2006) observation that tasks in ill-structured contexts necessitate "conditional logic and analogical reasoning" (p. 8), which obscure explicit uncertainty signals.

The high false-negative rate (FN=20) in biology underscores LLMs' difficulty in parsing intertwined conceptual reasoning and metacognitive monitoring, particularly when task-specific terminology masks underlying uncertainty markers. These results enhance our understanding of how metacognitive processes manifest differently across well-structured (mathematics) and ill-structured (biology) task environments due to variations in task demands and epistemic grounding—a distinction with significant implications for both SRL assessment methods and prompt engineering approaches.

*6. Practical Implications*

Our findings have important implications for both the technical implementation of LLMs in educational research and broader SRL assessment practices. The differential performance among LLM architectures (GPT-4o, Claude 3.5 Sonnet, and Gemini Pro) reflects fundamental differences in model training and design. GPT-4o and Claude 3.5 Sonnet demonstrated similar overall accuracy, and their confusion matrices revealed distinct error patterns from Gemini Pro

These performance patterns align with recent technical analyses by Brown et al. (2023) and Lin et al. (2022), who documented similar error distribution differences across LLM architectures. The more conservative prediction patterns in GPT-4o and Claude 3.5 Sonnet likely result from training data differences and instruction-tuning approaches that influence the precision-recall tradeoff in classification tasks (Bommasani et al., 2022; Zhao et al., 2023; Katz et al., 2023). Such variations in model behavior are well-documented in LLM evaluation studies, where architectural and training differences lead to distinct error patterns even when overall accuracy is similar (Singhal et al., 2023; Schaeffer et al., 2023). These architectural differences carry important implications for researchers selecting LLMs for educational coding tasks, highlighting the need to consider not just overall accuracy metrics but also the specific types of classification errors most acceptable for particular research questions.

For educational researchers interested in implementing LLMs for SRL coding, our findings suggest several practical strategies. First, the consistent performance of MPS classification across tasks indicates that cognitive process codes with clear procedural markers may be particularly amenable to automated coding approaches, even without extensive task-specific contextualization. Second, the pronounced task differences in metacognitive process classification highlight the importance of developing specialized prompt strategies for processes that manifest differently across learning contexts. Third, the modest but significant improvement from \*few-shot\* prompting in specific contexts suggests that carefully selected examples can enhance performance for particular SRL codes and tasks.

These findings extend recent work by Chernikova et al. (2024) and Sailer et al. (2024) on optimizing LLMs for educational research applications. They highlight the importance of developing theory-informed prompt engineering strategies that account for both the general characteristics of SRL processes and their task-specific manifestations. By understanding these patterns, researchers can develop more effective protocols for combining human expertise with LLMs in ways that leverage their complementary strengths for SRL assessment.

*7. Future Direction and Conclusion*

Although our study demonstrates the potential of LLMs for coding SRL processes across different tasks, we identified key limitations and promising future directions. A significant limitation is current LLMs' struggle with long-range dependencies and lack of explicit training on cognitive and metacognitive processes. To address this, we propose future research exploring a hierarchical prompting framework that aligns with SRL theoretical models. This approach would first identify macro-level regulatory categories (e.g., monitoring, strategy use, planning) before progressing to specific micro codes like JOU+/JOU-, MPS, potentially improving classification accuracy by managing contextual information more effectively.

This hierarchical approach aligns with theoretical frameworks like Winne and Hadwin's (1998) COPES model and Greene and Azevedo's (2009) TAP coding framework, which conceptualize SRL as occurring through structured, nested processes. Following initial categorization, subsequent prompting stages would progressively refine the classification using nested decision trees aligned with these SRL theoretical frameworks. Each stage would incorporate both bottom-up linguistic analysis and top-down theoretical constraints, potentially addressing the complex challenge of identifying functionally equivalent regulatory processes across different task manifestations.

Future research should also explore multimodal approaches that integrate verbal protocols with other data streams. As suggested by Järvelä et al. (2019) and Molenaar and Järvelä (2022), combining TAPs with eye-tracking, physiological measures, and log data could provide complementary indicators of SRL processes that are less sensitive to verbalization differences across tasks. These multimodal approaches could potentially address some of the task-specific challenges identified in our analysis, particularly for metacognitive processes that showed substantial cross-task variability.

Our findings demonstrate that LLM-based approaches can reliably code certain SRL processes (particularly MPS) across different STEM tasks, suggesting their potential as scalable tools for coding TAPs. However, the varying performance across different SRL codes highlights the need for continued refinement of prompting strategies that account for both the theoretical structure of SRL and the contextual nature of its manifestation.

**Acknowledgements**

**References**

**Appendix**

**Prompt example (MPS: FS-CoT-CR-TA-CA)**

As an expert in educational psychology, analyze this student utterance for the self-regulated learning process: MPS.

MPS Definition:

Mathematical Problem-Solving (MPS): Student actively working through a mathematical problem, showing steps or calculations.

Rubrics:

Components for MPS:

A. Problem Elements: Presence of relevant mathematical concepts, formulas, or calculations.

B. Active Engagement: Evidence of student actively solving or attempting to solve a mathematical problem.

Scoring for MPS:

- Satisfied (1): BOTH components A AND B are present.

- Non\_Satisfied (0): ONE OR NONE of the components are present.

Reasoning steps:

1. Identify key elements in the utterance.

2. Assess how these elements relate to the definition of the TAP code.

3. Consider the context provided and make reasonable inferences.

4. Evaluate your confidence in the classification.

5. Make a classification decision.

Context:

1. [2024-10-31 18:05:00] I know that's Q-squared

2. [2024-10-31 18:05:07] What proportion is born homozygous recessive

3. [2024-10-31 18:05:13] That's going to be very small

4. [2024-10-31 18:05:19] P + Q add up to one

5. [2024-10-31 18:05:50] so that's going to be long tails 280 plus 180

Utterance to analyze: 180 long tail, these also have a long tail plus 240, so total is 420. Out of 420 plus 80 out of 500,

Analyze the following utterance:

"180 long tail, these also have a long tail plus 240, so total is 420. Out of 420 plus 80 out of 500,"

Provide a brief analysis and classify as 1 (presence of the TAP code) or 0 (absence of the TAP code).

End your response with:

Confidence Level: [not sure/confident]

Classification: [1 or 0]

**Appendix X**

Performance Metrics by Model, TAP Code, and Domain

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **TAP Code** | **Domain** | **Accuracy** | **Precision** | **Recall** | **F1** | **n** |
| **Claude 3.5** | SG | Biology | 0.553 | 0.593 | 0.34 | 0.432 | 300 |
| Math | 0.83 | 0.837 | 0.82 | 0.828 | 300 |
| ROA | Biology | 0.58 | 1 | 0.16 | 0.276 | 300 |
| Math | 0.853 | 1 | 0.707 | 0.828 | 300 |
| JOU+ | Biology | 0.583 | 0.617 | 0.44 | 0.514 | 300 |
| Math | 0.863 | 0.819 | 0.933 | 0.872 | 300 |
| JOU- | Biology | 0.543 | 0.576 | 0.327 | 0.417 | 300 |
| Math | 0.91 | 0.855 | 0.987 | 0.916 | 300 |
| MPS | Biology | 0.687 | 0.792 | 0.507 | 0.618 | 300 |
| Math | 0.847 | 0.813 | 0.9 | 0.854 | 300 |
| FNC | Biology | 0.573 | 0.662 | 0.3 | 0.413 | 300 |
| Math | 0.697 | 0.766 | 0.567 | 0.651 | 300 |
| **GPT-4o** | SG | Biology | 0.543 | 0.587 | 0.293 | 0.391 | 300 |
| Math | 0.69 | 0.761 | 0.553 | 0.641 | 300 |
| ROA | Biology | 0.573 | 1 | 0.147 | 0.256 | 300 |
| Math | 0.897 | 1 | 0.793 | 0.885 | 300 |
| JOU+ | Biology | 0.54 | 0.539 | 0.547 | 0.543 | 300 |
| Math | 0.827 | 0.761 | 0.953 | 0.846 | 300 |
| JOU- | Biology | 0.563 | 0.58 | 0.46 | 0.513 | 300 |
| Math | 0.883 | 0.825 | 0.973 | 0.893 | 300 |
| MPS | Biology | 0.677 | 0.798 | 0.473 | 0.594 | 300 |
| Math | 0.807 | 0.791 | 0.833 | 0.812 | 300 |
| FNC | Biology | 0.537 | 0.641 | 0.167 | 0.265 | 300 |
| Math | 0.69 | 0.777 | 0.533 | 0.632 | 300 |
| **Gemini Pro** | SG | Biology | 0.45 | 0.426 | 0.287 | 0.343 | 300 |
| Math | 0.62 | 0.667 | 0.48 | 0.558 | 300 |
| ROA | Biology | 0.547 | 0.621 | 0.24 | 0.346 | 300 |
| Math | 0.843 | 0.926 | 0.747 | 0.827 | 300 |
| JOU+ | Biology | 0.547 | 0.534 | 0.727 | 0.616 | 300 |
| Math | 0.6 | 0.565 | 0.867 | 0.684 | 300 |
| JOU- | Biology | 0.48 | 0.484 | 0.593 | 0.533 | 300 |
| Math | 0.77 | 0.706 | 0.927 | 0.801 | 300 |
| MPS | Biology | 0.623 | 0.683 | 0.46 | 0.55 | 300 |
| Math | 0.737 | 0.703 | 0.82 | 0.757 | 300 |
| FNC | Biology | 0.533 | 0.556 | 0.333 | 0.417 | 300 |
| Math | 0.683 | 0.701 | 0.64 | 0.669 | 300 |

Accuracy: Proportion of all predictions (both positive and negative) that were correct.

Precision: Proportion of positive predictions that were actually positive (true positives divided by all positive predictions).

Recall: Proportion of actual positives that were correctly identified (true positives divided by all actual positives).

F1: Harmonic mean of precision and recall (2 × precision × recall) / (precision + recall), providing a balanced measure when classes are imbalanced.

n: Number of sampled utterances evaluated per condition.

**Appendix**

**TAP code definition, rubrics, and examples updated from codebook**

|  |  |  |
| --- | --- | --- |
| **TAP Code Definition** | **Rubrics** | **Examples** |
| Mathematical Problem-Solving (MPS): Student actively working through a mathematical problem, showing steps or calculations. | **Components for MPS:**   A. Problem Elements: Presence of relevant mathematical concepts, formulas, or calculations.   B. Active Engagement: Evidence of student actively solving or attempting to solve a mathematical problem.   **Scoring for MPS:**   - Satisfied (1): BOTH components A AND B are present.   - Non\_Satisfied (0): ONE OR NONE of the components are present. | **Positive example for MPS:**   "To find the volume, I need to multiply length times width times height. So that's 5 × 3 × 4..."   Classification: 1 (MPS)     **Negative example for MPS:**   "I'm writing down this formula for future reference."   Classification: 0 (Not MPS)   Explanation: This is Taking Notes (TN), not active problem-solving. |
| Forming New Conclusion (FNC): Putting together two pieces of information and drawing a new conclusion that extends beyond what is presented in the learning environment. | **Components for FNC:** A. Integration of Information: Combining at least two pieces of information (from current representation, previous representation, or prior knowledge). B. Novel Conclusion: Drawing a conclusion that goes beyond the explicitly presented information. Scoring for FNC: - Satisfied (1): BOTH components A AND B are present. - Non\_Satisfied (0): ONE OR NONE of the components are present. | **Positive example for FNC:** "My guess is taxes would probably go up in order for the government to pay for all of the healthcare." Classification: 1 (FNC)   **Negative example for FNC:** "The text says that DNA is the genetic material in cells." Classification: 0 (Not FNC) Explanation: This is simple recall or reading, not forming a new conclusion. |
| Judgment of Understanding (JOU): Learner recognizes that they do (JOU+) or do not (JOU-) understand content related to the learning task. | **Components for JOU:** A. Expression of Understanding: Clear statement about comprehension or lack thereof. B. Content Relevance: The understanding (or lack of) is related to the learning task or content. Scoring for JOU: - JOU+ (1): Both components A and B are present, indicating understanding. - JOU- (1): Both components A and B are present, indicating lack of understanding. - Non\_JOU (0): One or none of the components are present. | **Positive example for JOU+:** "That makes sense." Classification: 1 (JOU+)  Negative example for JOU+ (actually JOL+): "I think I'll be able to recall this information during the test next week." Classification: 0 (Not JOU+) Explanation: This is a Judgment of Learning (JOL+), not current understanding. |
| Sub-Goal:[target] (SG): Learner articulates a specific sub-goal that is relevant to the task. Must immediately carry out some action relevant to the sub-goal (i.e., can't drop the goal immediately after verbalizing). | **Components for SG:target:** A. Articulation: Clear statement of a specific sub-goal. B. Task Relevance: The sub-goal is relevant to the current learning task. C. Immediate Action: Indication of carrying out an action related to the stated sub-goal. D. Target Identification: Specifies the target of the goal (e.g., GRQ, specific question, concept). Scoring for SG:target: - Satisfied (1): ALL components A, B, C, and D are present. - Non\_Satisfied (0): ONE OR MORE of the components are missing. | **Positive example for SG:target:** "I'm going to go back to the e-text, scroll down, and make sure that "Q" would equal the recessive allele." (SG:GRQ\_Q4) Classification: 1 (SG:target)   Negative example for SG:target: "I'm going to download the GRQs so I can type my answers." Classification: 0 (Not SG:target) Explanation: While it articulates a goal (A), it's not a learning-related subgoal (B), and doesn't demonstrate a specific learning objective or target (D). |
| Ruling Out Answers (ROA): Reviewing answer choices on a multiple-choice question and systematically ruling out answer choices in order to narrow down to the answer they will select. | **Components for Ruling Out Answers:** A. Multiple Choice Context: Occurs in the context of answering a multiple-choice question. B. Review: Demonstrates a methodical assessment and consideration of answer options. C. Elimination Process: Actively rules out one or more answer choices. D. Reasoning: Provides some rationale for ruling out choices (even if brief or implied). Scoring for Ruling Out Answers: - Satisfied (1): ALL components A, B, C, and D are present. - Non\_Satisfied (0): ONE OR MORE of the components are missing. | **Positive example for Ruling Out Answers:** "Well, I can rule out A and D right away..." Classification: 1 (Ruling Out Answers)   Negative example for Ruling Out Answers: "My guess is taxes would probably go up in order for the government to pay for all of the healthcare." Classification: 0 (Not Ruling Out Answers) Explanation: It's not in a multiple-choice context (A) and it doesn't show review (B), and an elimination process (C), and the reasoning isn't based on ruling out other options (D). |

**Appendix**

**Robust check results for the optimal utterances window (Based on results from Claude 3.5 in ZS-CT-COT)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TAP Code** | **Utterance Window** | **Count** | **Accuracy** | **Std** |
| **FNC** | 0 | 20 | 0.45 | 0.51 |
| **FNC** | 1 | 20 | 0.6 | 0.50 |
| **FNC** | 3 | 20 | 0.6 | 0.50 |
| **FNC** | 5 | 20 | 0.65 | 0.49 |
| **FNC** | 10 | 20 | 0.5 | 0.51 |
| **MPS** | 0 | 20 | 0.6 | 0.50 |
| **MPS** | 1 | 20 | 0.75 | 0.44 |
| **MPS** | 3 | 20 | 0.55 | 0.51 |
| **MPS** | 5 | 20 | 0.8 | 0.41 |
| **MPS** | 10 | 20 | 0.8 | 0.41 |

**Appendix X**

A diagram of a science lesson

AI-generated content may be incorrect.

Figure 1. The laboratory task design for Biology TAPs from Bernacki et al., (2024)

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure 2. The laboratory task design for Math TAPs from Yu et al., (under review)

**Table Biology TAPs from Bernacki et al., (2024)**

A screenshot of a data sheet

AI-generated content may be incorrect.