Acknowledgement

This version of the contribution has been accepted for publication, after peer review (when applicable) but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: http://dx.doi.org/[TO BE DETERMINED].

Citation

Please cite this manuscript as:

Zhang, J., Borchers, C., & Barany, A. (2024, November). Studying the Interplay of Self-Regulated Learning Cycles and Scaffolding Through Ordered Network Analysis Across Three Tutoring Systems. *Proceedings of the International Conference on Quantitative Ethnography*. Cham: Springer Nature Switzerland.

Studying the Interplay of Self-Regulated Learning Cycles and Scaffolding Through Ordered Network Analysis Across Three Tutoring Systems

Jiayi Zhang*(✉)1[0000-0002-7334-4256], Conrad Borchers*(✉)2[0000-0003-3437-8979], and Amanda Barany1[0000-0003-2239-2271]

[1] University of Pennsylvania, Philadelphia, PA 19104, USA
joycez@upenn.edu
[2] Carnegie Mellon University, Pittsburgh, PA 15213, USA
cborcher@cs.cmu.edu

Abstract. Self-regulated learning (SRL) is essential for learning across various contexts and domains. While technology-based learning environments can support SRL, comparisons of SRL processes across learning platforms and domains are scarce. As most prior research has investigated SRL patterns across learner performance levels, methods are lacking to investigate if adaptive support adequately supports learners' temporal SRL during problem solving. This study leverages ordered network analysis (ONA) to investigate SRL processes in terms of processing information, making plans, enacting plans, and realizing errors across platform designs and domains. We analyzed think-aloud data from fifteen students working in three intelligent tutoring systems with high and low degrees of scaffolding spanning the domains of chemistry and formal logic. Students engaged in more SRL transitions in less scaffolded, open-ended platforms and when solving logic problems. Conversely, highly scaffolded environments allowed learners to enact problem-solving operations without prior planning more easily. Future research may investigate the degree to which such active learning without planning is desirable, as it might reduce learning differences predicated on SRL, but also fewer learning opportunities to plan. Our results suggest ONA is a useful methodology for studying the interplay of SRL and scaffolding during tutored problem solving.

Keywords: Self-Regulated Learning, Ordered Network Analysis, Intelligent Tutoring Systems, Scaffolding, Multimodal Learning Analytics, Behavioral Log Data

1 Introduction

Self-regulated learning (SRL) involves learners monitoring and regulating their behaviors and strategies to pursue goals [33]. A range of cognitive, affective, metacognitive, and motivational processes are involved in SRL. Seminal definitions of SRL describe SRL as a cyclical process [29, 33]. For example, Winne and Hadwin [29] describe the process of SRL as four interdependent and recursive stages, in which learners: 1) define the task, 2) set goals and form plans, 3) enact the plans, and 4) reflect and adapt strategies when goals are not met. Students who are skilled at SRL, such as those who engage in self-regulatory behaviors (e.g., planning

[1] J. Zhang and C. Borchers contributed equally to this research.

and reflecting; [5]) or demonstrate coherent and strategic self-regulating processes (e.g., planning before enacting; [5]) were found to be more successful in learning.

Computer-based learning environments (CBLEs) have design affordances for supporting learning processes, particularly tools such as intelligent tutoring systems (ITSs), which offer personalized learning experiences tailored to student needs and performance [28]. To achieve this, such platforms pose SRL demands to students by providing flexibility in deciding what and how to learn. A higher degree of regulation may be required for effective learning due to the flexibility and reliance on learners' agency in these environments [26]. As a result, students who do not self-initiate SRL processes may not benefit from learning in CBLEs [3].

While research exists on the ways to support SRL in CBLEs, such explorations remain limited in scope. Notably, CBLEs include diverse degrees of problem-solving and SRL support. ITS, in particular, are well-suited learning environments because they cover a wide range of domains [14]. While metacognitive scaffolding has been linked to increased SRL activities [19], another hypothesis suggests that more structured problem-solving scaffolding allows learners to rely less on self-initiated SRL processes [26]. Past research has studied SRL differences among students within single learning environments and provided personalized scaffolds. These scaffolds help students search, assemble, plan, and monitor during their learning activities [3].

Recent work in quantitative ethnography has leveraged the temporal nature of SRL through ordered-network analysis to analyze differences in student learning strategies [10]. However, these lines of research have focused on studying SRL differences in a *single* environment rather than *multiple* environments with different levels of support and domains. This is a problem because differences in SRL and problem-solving support between platforms are expected to change students' SRL processes [3]. Consequently, there is a lack of generalizable scientific knowledge about *what kind* and *how much* SRL support is adequate in different CBLEs, which is a fundamental design challenge in instruction [34].

Between-platform comparisons of SRL are scarce in the literature, as it is time-consuming to identify common behavioral proxies for SRL across platforms [9]. However, the analysis of think-aloud data through artificial intelligence (e.g., automated transcription and prediction of SRL [31]) has made platform-independent coding of SRL at scale feasible. Contrasting SRL across platforms not only boosts the generalizability of scientific findings but can also offer insights into how the design of online learning environments interacts with SRL processes.

This study investigates SRL across three intelligent tutoring systems (ITSs) that provide learners with varying degrees of problem-solving support. We use ordered network analysis (ONA) to examine SRL processes when students work in ITSs that vary in scaffolding (scaffolded vs. open-ended platform design) and domains (chemistry vs. formal logic). Given the temporal nature of SRL established in the literature, ONA is an appropriate methodology to capture temporal differences in SRL [10]. Through this analysis, we investigate how different levels of problem-solving scaffolding and domain differences relate to learners' SRL activities in ITSs.

2 Background

2.1 Intelligent Tutoring Systems and Their Instructional Support

An important domain of problem-solving practice in which SRL has been studied is intelligent tutoring systems (ITSs) [28]. ITSs provide adaptive support during problem solving through as-needed support in the form of hints and feedback. Further, the interaction design of ITSs (such as the types of components students use to complete problem-solving steps) constrains the range of inputs and viable problem-solving steps, unlike freeform problem solving on paper. This support is generally referred to as scaffolding, as it helps learners generate answers to steps beyond their current ability, posing additional opportunities to learn. Scaffolds can further include metacognitive support, such as by prompting to self-explain or plan.

Lack of scaffolding can take away learning opportunities; particularly when learners cannot generate solutions to problem-solving steps. Optimal scaffolding can be achieved by gradually reducing the amount of scaffolding and increasing the difficulty, a process known as fading [23]. Too much or the wrong kind of scaffolding can take away opportunities to learn from committing common errors [17]. Therefore, a better understanding of the instructional factors that relate to SRL demands and optimal learner support is a key research objective in ITSs [34].

2.2 Scaffolding and Self-Regulated Learning

While much of the literature on ITS focuses on scaffolds that support problem-solving, there have also been many successful attempts at providing metacognitive support that facilitates SRL [3]. For example, instructional models have been designed to tutor students on optimal help-seeking [2] or support learners in organizing SRL strategies [3]. The interplay between SRL and affect in computerized learning environments has also been studied in the context of desirable degrees of confusion resolution and learning [32].

Even if not explicitly designed to support SRL, problem-solving and learning in ITS relate to learners' SRL processes and the level of SRL support provided. Levels of SRL and prior SRL training have been linked to better learning outcomes in tutoring systems [4]. Conversely, task complexity and stages of SRL can work together to shape student affect during learning [32]. SRL, especially when analyzed with respect to learners' cyclical SRL patterns, can help explain performance differences during ITS practice [6]. Overall, the literature paints a diverse image of how SRL processes, levels of instructional support, and task complexity relate to student learning during tutored problem-solving.

2.3 Scaffolding and SRL Across Problem-Solving Domains

The appropriateness of scaffolds during problem-solving depends on the instructional domain and the design of the learning environment. There is a tradeoff between supporting flexible strategies and scaffolded strategies in ITS. Flexible strategies allow learners to perform multiple problem-solving steps at any time, while scaffolded strategies limit the interface to performing specific steps. For example, during linear equation-solving using a flexible, formula-based entry interface, learners may skip steps by entering solutions upfront (e.g., "x=2" for "2x+2=6" rather than "2x=4" as in the Lynnette ITS [16]). Conversely, an interface requiring students to enter graph coordinates into a predefined table (e.g., MathTutor Units 7.05 and 7.06 [7]), restricts flexibility and reduces planning demands.

These interface differences present unique challenges and affordances compared to free-form problem solving on paper [7]. Scaffolding styles are also related to domain differences; some domains, like equation-solving, are suited for open-ended problem solving, while others, like linear graphing, benefit from scaffolded approaches. With this understanding, domain-level differences in SRL demands and processes may be empirically studied to improve learning outcomes of interest.

Research comparing multiple domains of problem solving with one another regarding SRL is limited, however. Notable examples include a study of ill-structured problems across the domains of information problem solving, historical inquiry, and science inquiry that derived a domain-general framework of how SRL acquisition relates to such problems [11]. Another study highlighted that motivational differences relevant to SRL are often domain-dependent [21]. However, these studies did not link domain-intrinsic differences to SRL demands and processes (e.g., problem-solving structure, interaction design of instructional material). Indeed, because SRL is broadly considered a domain-general competence, domain differences in problem solving strategies and interaction design in relationship to SRL are understudied. The present research aims to close this gap by contrasting both levels of scaffolding and two domains (chemistry and formal logic) in three tutoring systems with think-aloud protocols coded according to [29]'s four-stage SRL model.

2.4 Epistemic Network Methods to Study SRL

Several studies have used quantitative ethnographic approaches such as epistemic and ordered network analysis to examine SRL processes (e.g., co-occurrence and transition between SRL behaviors), visualizing how SRL differs across student groups and contexts. For example, using standard networks, [20] examined the co-occurrence of SRL actions, such as reading and revising, between high and low performers in an open-ended problem-solving environment. Their findings align with prior research showing that high performers' SRL patterns align with SRL cycles [6]. Additionally, different SRL patterns (higher co-occurrence between certain SRL behaviors) are observed when students solve a difficult task [12] or when students become more familiar with a digital learning platform [30]. To understand directional patterns of SRL behaviors, [15] examined SRL sequences by constructing ordered networks for three groups of students (unsuccessful, successful, and mastery-oriented). They found that the mastery-oriented group had stronger interactions between SRL behaviors (denser network), and the sequence of behaviors supported the theory that early SRL behaviors influence conditions for later activities. These studies substantiate evidence that epistemic network analysis is a valuable approach to examining the intricacy and dynamics of SRL processes that can be expanded to compare across platforms.

3 Methods

The present study examines SRL differences across platform designs and domain areas. Specifically, we collected students' log data along with think-aloud transcripts while interacting with three ITSs that differ in platform designs (scaffolded vs. open-ended) and domains (stoichiometry chemistry vs. formal logic). We coded the think-aloud transcripts from the three platforms for four SRL categories and evaluated the frequency of these categories as well as their transitions using ordered networks.

Fifteen students participated in the study and worked on problems in at least one of the three platforms. All participants were enrolled in degree programs in the United States. The participants were 40.0% white, 46.6% Asian, and 13.3% multi- or biracial and included undergraduate first-year students (21.4%), sophomores (14.3%), juniors (35.7%), seniors (21.4%), and one graduate student (7.1%).

3.1 Learning Platforms

Three open-source ITSs were used in the study, which are Stoichiometry Tutor [1], Open-Response Chemistry Cognitive Assistant (ORCCA) Tutor [13], and Logic Tutor. The first two ITSs, Stoichiometry and ORCCA Tutors, focus on stoichiometry chemistry, while Logic Tutor covers formal logic. Example screenshots of the three ITSs can be found in a digital appendix: https://tinyurl.com/49yya6j4.

All three systems offer step-level tutoring, meaning that students receive feedback on whether their problem-solving step attempt was correct or not. All three systems include hints, which provide learners with as-needed instruction to complete problem-solving steps. All three ITSs logged students' interactions (e.g., clicks, formula entries, hints requested) as timestamped transactions to PSLC DataShop [35]. Together with comparable interaction design in terms of hints and immediate feedback, this procedure minimized potential differences in analytical results based on logging or tutoring styles. However, the three ITSs vary in their degree of problem-solving scaffolding, ranging from Stoichiometry Tutor (highly structured) to ORCCA and Logic Tutor (open-ended), as described next.

First, Stoichiometry Tutor employs a structured, fraction-based approach to problem-solving. Stoich Tutor's interface aligns with the factor-label method, a commonly taught strategy for stoichiometry problems in the United States which guides learners to convert units from a given to a target value [24]. This structured interface requires learners to select values from a drop-down menu or enter quantities while removing demands related to strategy selection. The tutoring system does not allow students to skip individual problem-solving steps.

ORCCA functions as an open-ended, rule-based ITS. The system aligns problem-solving rules with students' individual strategies, allowing for flexible problem-solving sequences through a formula interface. Students can enter transformations to stoichiometry equations to derive a final target; the system can recognize any viable transformation with the option of compounding or entering upfront problem-solving steps and delivering adaptive instruction. Relatedly, ORCCA's hints provide dynamic scaffolds, meaning that they prompt learners to construct a viable next step based on their current problem-solving stage. Finally, due to ORCCA's flexibility, its error feedback is constrained to telling the learner if an attempted step is right or wrong, rather than providing conceptual, error-specific feedback. By accommodating more flexible problem-solving and allowing learners to enter upfront or compound problem-solving steps (rather than entering each required value as in Stoichiometry Tutor), ORCCA potentially poses additional SRL demands on learners as it requires active strategy selection [25].

Logic Tutor is a rule-based system teaching propositional logic, where students construct truth tables and manipulate formulas using logical connectives. However, unlike ORCCA, which requires recalling domain knowledge (e.g., unit conversion, stoichiometric multiplication), Logic Tutor provides a cheat sheet for constructing logical equations. This aids students in focusing on problem-solving without needing to recall information, as recalling transformation rules is not a learning goal in formal logic. Rather, Logic Tutor teaches deriving proofs through transformation. Like ORCCA, Logic Tutor provides dynamic scaffolding for viable next steps in the problem-solving sequence. However, unlike ORCCA, Logic Tutor gives error-specific feedback by giving counterexamples for incorrect formula transformations.

3.2 Study Procedure

While working on a problem, students were asked to think aloud, verbalizing their problem-solving process. An experimenter reminded them to continue speaking if they remained silent for more than five seconds. These verbalizations were recorded and subsequently transcribed using Whisper, an open-source transcription model for voice, which segmented utterances with start and end timestamps [22]. Timestamped log data of student-tutor interactions (e.g., problem-solving step attempts, hint requests) was also collected from all three platforms and synchronized with think-aloud utterances. The method for synchronizing log data with transcripts and their verification is described in [6]. To prepare for ordered network analysis, data was organized chronologically by student and problem.

We defined lines of data as utterances occurring between two consecutive student transactions in the ITS, meaning button clicks, and formula entries, among others. This approach allows us to examine and code the use of SRL in a longer time span and contextualize the coding based on adjacent actions, as demonstrated in [6]. Together, the fifteen students worked on 40 problems and produced 955 lines, which were included in the following analysis. Log data and anonymized synchronized think-aloud transcripts are available via PSLC DataShop (datasets #5371 and #5820).

3.3 Coding SRL Categories

To examine temporal SRL processes, we grounded our coding in Winne and Hadwin's four-stage model as it suggests a cyclical process of SRL in which learners 1) understand a task, 2) set goals and make plans, 3) enact the plan, and 4) reflect and adapt strategies when goals are not met [29]. Upon reviewing the lines, we inductively identified and operationalized four SRL categories that capture a subset of behaviors within each model stage. These SRL categories include *Processing Information, Planning, Enacting, and Realizing Errors*. While broader in scope compared to other SRL think-aloud studies (e.g., [5]), this level of categorization enables the observation of finer-grained cognitive operations within relatively brief utterances made between problem-solving attempts. Table 1 provides an overview of the coding categories, associated behaviors, and example utterances.

Two coders established acceptable inter-rater reliability after coding 162 utterances ($K_{processing}$ = 0.78, $K_{planning}$ = 0.90, $K_{enacting}$ = 0.77, K_{errors} = 1.00). They then individually coded the remaining utterances, double-coding any instances lacking agreement within the inter-rater iteration. Each utterance can be coded for more than one category. In total, 955 utterances were coded for the four SRL categories. Overall, 19% of utterances were assigned the Processing Information code, 13% the Planning code, 25% the Enacting code, and 10% the Realizing Errors code, respectively. 48% of utterances were assigned to no code.

Table 1. SRL categories, behaviors, and example utterances.

SRL Category	Behavior	Example Utterance
Processing Information	· Assemble information The utterance demonstrates behaviors where students read or re-read a question, hints, or feedback provided by the system · Comprehend information The utterance demonstrates behaviors where students repeat information provided by the system with a level of synthesis	"Let's figure out how many hydrogen items are in a millimole of water molecule H2O molecules. Our result should have three significant features."
Planning	· Identify goals and form plans The utterance reflects behaviors where students verbalize a conceptual plan of how they will solve the problem	"Our goal of the result is hydrogen atoms. The goal of the result is the number of hydrogen atoms, right?"
Enacting	· Verbalize previous action The utterance reflects students' behaviors where they verbalize an action that has just been carried out explaining what they did · Announce the next action The utterance reflects student behaviors where they verbalize a concrete and specific action that they will do next	"Two molecules of this. How many atoms in a minimum molecule of M mole? 61023 divided by 2. 3.0115."
Realizing Errors	· Realize something is wrong The utterance demonstrates instances where students realize there is a mistake in the	"It's incorrect. What's happened? It is the thousand in the wrong

	answer or the process with or without external prompting (i.e., tutor feedback)	spot…No, the thousand is correct, so what am I doing wrong?"

3.4 Analytical Methods

With the SRL codes, we conducted two separate analyses (frequency and ordered network analysis) for each set of comparisons to evaluate the differences in SRL across platform designs and domains.

Frequency. For each set of comparisons, we first analyzed the frequency of SRL codes to determine if certain SRL categories appeared more frequently in one platform design or domain than the other. Specifically, we calculated the percentage of times each code was observed out of the total number of utterances for each platform design and domain. Subsequently, we conducted a Wilcoxon rank sum test to assess the significance of differences between the two groups.

Ordered Network Analysis (ONA). ONA identifies and quantifies directed connections among nodes in data by accounting for the order of events and visualizing these connections in network models [27]. As we are interested in examining the temporal order of SRL as it unfolds over time throughout the problem-solving process, we applied ONA using the SRL codes. We generated two sets of ordered networks using the WebTool (version 1.7.0) [18], visualizing the differences in SRL processes between platform designs and domains.

The first set of ordered networks compares the SRL processes between scaffolded and open-ended platform designs. The unit of analysis is platform design (i.e., scaffolded vs. open-ended), subset by students and problems. Conversations consisted of all the utterances collected from one student working on one problem (using conversation variables student ID and problem name). We chose a moving stanza with a window length of four. The window length was chosen primarily due to the length of an SRL process described in the four-stage model. However, as SRL may not always be a linear process, we also experimented with various window sizes (N=5,6), which did not result in substantial visual or statistical changes in the models. A mean rotation was used to maximize differences across the x-axis. Our model has Spearman co-registration correlations of $r = 0.88$ for the first dimension and $r = 0.97$ for the second, indicating a strong goodness of fit.

The second set of networks compares the SRL processes between students working on chemistry and formal logic problems. The unit of analysis is defined as domains (i.e., chemistry vs. formal logic), subsetted by student and problem. Conversation variables were set as student ID and problem name. A moving stanza with a window size of four was used to establish the connection, and a mean rotation was applied to maximize the differences between groups. The Spearman co-registration correlation is $r = 0.91$ for the first dimension and $r = 0.97$ for the second, indicating a strong goodness of fit. For each set of the networks, Mann-Whitney tests were calculated to assess if the mean difference between the two groups was significant.

4 Results

4.1 SRL Processes across Platform Designs: Scaffolded vs. Open-Ended

The Frequency of SRL Categories. We calculated the frequency of the four SRL categories in the scaffolded (Stoichiometry Tutor) and open-ended (ORCCA and Logic Tutor) platform designs. Students were significantly more likely to engage in *Processing Information,* *Planning, and Realizing Errors* when using an open-ended platform (see Table 2). Students were equally likely to engage in *Enacting* in both platform designs.

Table 2. The frequency of SRL between scaffolded and open-ended platforms.

Platform Design	Processing Information	Planning	Enacting	Realizing Errors
Scaffolded	0.132	0.107	0.226	0.064
Open-ended	0.241	0.154	0.270	0.140
Wilcoxon rank sum test	W = 126338, $p < .001$	W = 119404, $p = .029$	W = 118928, $p = .120$	W = 122623, $p < .001$

The Transition of SRL Categories. ONA provides insights into the temporality of the SRL processes. Figures 1a and 1b demonstrate the SRL process among the four SRL categories within the two platform designs. Figure 1c shows the differences between the two types of platform designs.

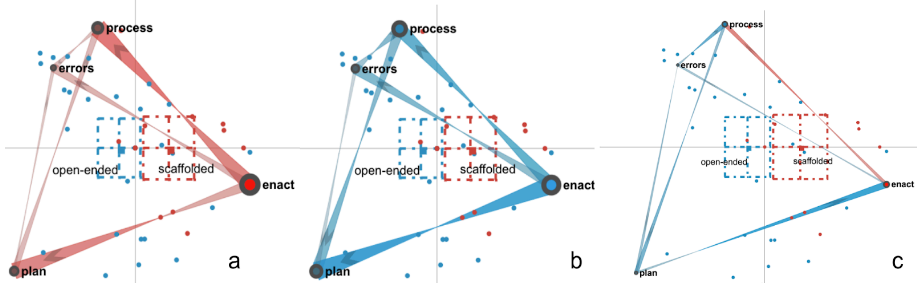

Fig. 1. Ordered network of SRL processes between scaffolded (1a, red) and open-ended (1b, blue) platforms, with a difference network (1c).

For scaffolded platforms (See Figure 1a), it is more likely for students to repeat *Enacting* (making attempts) than to repeat any other SRL strategies (lw=0.42). Upon examining pairwise connections between SRL codes in the form of edges, we see that students tended to *Enact* first and then *Plan* (lw=0.13) or *Enact* first and then *Process Information* (lw=0.21). Sometimes this attempt (*Enact*) leads to *Realize Errors* (lw=0.11). Additionally, we observe that students tended to *Plan* before *Processing Information* (lw=0.09). *Processing Information* (lw=0.07) or *Planning* (lw=0.04) also take precedence over *Realizing Errors*. However, these connections are less prevalent than edges that originate from the *Enact* node. This means that, in a tutoring system with high degrees of scaffolding, learners most commonly enacted multiple problem-solving step attempts in a row (e.g., entering a quantity or unit conversion). While doing so, they would verbalize multiple actions in a row, for example, “So canceling out the grams of P4, [ENACT] canceling out the moles of P4, [ENACT]” In this example, by providing a pre-defined template for a unit conversion, the student was able to enact multiple problem-solving steps without planning.

In open-ended platforms (See Figure 1b), students less often repeated *Enacting* (Enact node) (lw=0.29) compared to the highly-scaffolded Stoichiometry Tutor. Rather, students tended to *Enact* before transitioning into any of the other three SRL strategies. Unlike scaffolded platforms, where *Planning* typically preceded *Processing Information*, in open-ended platforms, students tend to *Process information* before *Planning* (lw=0.13). Moreover, students tended to *Realize Errors* first and then *Process Information* (lw=0.13) or *Realize Errors* first and then *Plan* (lw=0.04), as opposed to the reverse direction observed in scaffolded platforms – *Processing* or *Planning* first and then *Realizing Errors*. Specifically, students would often turn to instruction (e.g., the problem statement in the tutoring system) after errors and prior to planning to orient themselves toward the next problem-solving step. For example, one student working with ORCCA noted: “Why is that wrong? [ERROR]...Okay, let's try again. The COH4 in mole per kilogram [PROCESSING].” This means that, unlike in the highly-scaffolded Stoich Tutor, students would more commonly process instruction to enact steps again rather than enacting further steps.

The comparison plot (See Figure 1c) further highlights the differences in SRL transitions in the two networks. Along the X axis (MR1), a Mann-Whitney test showed that SRL processes in scaffolded platforms (Mdn=0.13, N=13) were statistically significantly different from SRL processes in open-ended platforms (Mdn=-0.04, N=27 U=264.00, p=.010, r=-0.50). When comparing the two networks, we find that when working in open-ended platforms, students were more inclined to engage in SRL transitions and were more likely to *Plan* first and then *Enact*. However, when working in scaffolded platforms, students were more likely to *Enact* and then *Process Information*. We suspect that this is due to the design of scaffolding used in the platform, which leads students to be more reactive than proactive in terms of engaging in SRL. As such, the pattern we observe (*Enacting* → *Processing*) may demonstrate a common behavior where students make an attempt (*Enact*) first and then *Process* system feedback, rather than *Processing Information* or *Planning* before *Enacting*, as suggested in the cyclical sequence of SRL. Specifically, in open-ended ITS, students would first verbalize *what* they would do next before doing it (i.e., planning a unit conversion) rather than enacting an attempt through the help of the tutoring system (through scaffolding or processing of instruction). In Logic Tutor, planning after processing typically involved selecting appropriate transformations from the cheat sheet: “And it showed me I have a syntax error so I'll try to match up the parentheses. [PROCESS]. And I try to look through the rules that I can use. I think we can try to use inverse absorption here so we can simplify [PLAN].”

4.2 SRL Processes across Domains: Chemistry vs. Formal Logic

The Frequency of SRL Categories. There is a significant difference in how often students engage in SRL across the four categories when working on chemistry and logic problems (see Table 3). Students were significantly more likely to engage in all four SRL strategies when solving formal logic problems.

Table 3. The frequency of SRL between chemistry and logic tutors.

Domain	Processing Information	Planning	Enacting	Realizing Errors
Chemistry	0.152	0.111	0.217	0.059
Formal Logic	0.256	0.170	0.309	0.188
Wilcoxon rank sum test	W = 91588, p < .001	W = 96210, p = .011	W = 92866, p = .002	W = 88970, p < .001

The Transition of SRL Categories. Figures 2a and 2b are the ordered networks of the SRL process when students work on chemistry (red) and logic (blue) problems. Students working on chemistry problems tended to *Enact* first and then *Plan* or they *Enact* first and then *Process Information*. However, during logic problems, students tended to *Plan* first and then *Enact*. In the comparison plot (See Figure 2c), along the X axis (MR1), a Mann-Whitney test showed that the SRL process during solving chemistry problems (red; Mdn=0.04, N=22) was significantly different from the SRL process during solving logic problems (blue; Mdn=-0.21, N=18 U=84.00, p<.001, r=0.58), showing that SRL transitions are more frequently engaged when students solve logic problems than chemistry problems. As mentioned in Section 4.1, students working in Logic Tutor would often select appropriate transformation strategies before enacting problem-solving steps, with one student noting: “I'm looking at the rules that I can use again um I can use DeMorgan here which will simplify not not parentheses P or not Q into [PLAN] not P not P and and Q then we just copy over the rest [ENACT].” The interface led students to select strategies (plan) rather than enacting steps in chemistry tutors, which may relate to the learning goal of selecting among several viable transformations in formal logic, as we go on to discuss. The connection strength and directionality of all ONA models are in Table 4.

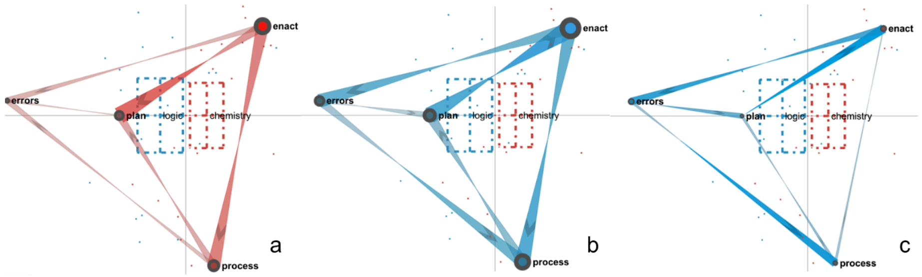

Fig. 2. Ordered network of SRL processes between logic (1a, red) and chemistry (1b, blue) domains with a difference network (1c).

Table 4. Strength of ordered networks. Bold values represent connections aligning (compared to misaligning) with the expected SRL sequence in the four-stage model [29].

	Platform Design		Domain	
Expected Sequence	Scaffolded	Open-ended	Chemistry	Logic
process -> plan	0.08	**0.14**	**0.11**	**0.14**
plan -> enact	0.20	0.24	0.22	**0.23**
enact -> errors	**0.11**	**0.14**	**0.08**	**0.19**
errors -> process	0.07	**0.13**	0.08	**0.17**
process -> enact	0.23	0.17	0.18	0.21
plan -> errors	**0.04**	0.06	**0.03**	0.09
process -> process	**0.12**	**0.22**	**0.18**	**0.2**
plan -> plan	**0.09**	**0.11**	**0.08**	**0.13**
enact -> enact	**0.42**	**0.29**	**0.35**	**0.32**
errors -> errors	**0.03**	**0.08**	**0.02**	**0.12**

5 Discussion

The present study has leveraged ordered network analysis (ONA) to study students’ use of SRL in three intelligent tutoring systems (ITSs) for chemistry and formal logic with low and high degrees of problem-solving support. Most past work compared SRL profiles between students and within single learning environments. As seminal models of SRL conceptualized SRL through a cyclical model which are *ordered* in a specific manner [29], ONA was employed to discover SRL-related differences with respect to sequences of *Processing Information, Planning, Enacting, and Realizing Errors* across ITSs. Our main contributions are as follows.

First, we contribute insights into the relationship of problem-solving scaffolding and SRL differences through novel applications of ONA to coded think-aloud data. Specifically, our findings indicate that ITSs with less scaffolding and structuring of problem-solving relate to higher SRL activity in learners, as signified by stronger ties between *Processing* and *Planning* as well as *Planning* and *Enactment*. In contrast, in highly scaffolded environments, students exhibited SRL patterns that were more centered around *Enacting* and less around *Planning*.

Why did planning more rarely precede enacting in the structured ITS? The tutoring system's scaffolding may allow students to engage in problem solving in a more straightforward way independent of existing planning abilities. If planning requires conceptual knowledge that some students do not yet have (as is common in stoichiometry [8]), such straightforward problem solving may aid learning. To the extent that the tutoring system allows (all) students to easily and actively engage in doing, such practice is desirable as it allows students to learn from feedback [14]. On the other hand, past studies documented higher performance in learners engaging in cyclical SRL, including planning [6]. How can both interpretations be reconciled? In open-ended environments such as ORCCA and Logic Tutor, it could be that students not only engage in more SRL during problem solving (as our results indicate) but *also* need to do so to learn successfully. Accordingly, in environments with high SRL demands, *some* students who follow an effective SRL cycle learn more, and *others* do not. In contrast, in highly scaffolded environments, all learners learn similarly well because all of them receive opportunities to learn, irrespective of their SRL profile. Recent large-scale analyses of practice data from tutoring systems align with that interpretation [14]. Alternatively, high levels of scaffolding would *not* lead to better learning as they remove opportunities to learn to plan after processing information. Enacting without planning might not give learners sufficient opportunity to construct solutions to complex problems through conceptual reasoning, an issue that prior work documented in stoichiometry [8]. Future research studying larger student samples than the present study could distinguish between both interpretations and relate results to SRL profiles distilled ONA to learning gain differences. Such investigations could answer whether the higher levels of SRL activity in lowly-scaffolded problem-solving environments relate to lower SRL *demands* required for learning.

For our second contribution, we found that students working in Logic Tutor had higher SRL activity than in the chemistry tutors, especially represented through a strong tie from *Planning* to *Enacting*. In other words, more planning is needed to act and learn from feedback in Logic Tutor. We suspect that there is a difference in problem difficulty between the two domains, where Stoich Tutor and ORCCA, unlike Logic Tutor, cover problems commonly taught in high school. Another instructional difference is that in Logic Tutor, strategy selection of transformations is a learning goal (e.g., DeMorgan's rule) that needs to be selected from a cheat sheet. This design might have focused learners on searching, selecting, and manipulating equations during problem-solving. Such requirements of strategy selection to construct answers are expected to increase SRL activity [25]. Rather than engaging in recall through active learning and getting feedback on the correctness of facts (e.g., unit conversions), formal logic requires the construction of a proof that cannot be easily recalled without planning. Hence, SRL differences captured through ONA can point to instructional differences between domains and inform instructional design.

5.1 Limitations and Future Work

Most participants in our sample worked with a single ITS. Qualitative differences documented between the tutoring systems could be confounded with student-level differences. To strengthen our interpretation that more problem-solving support leads to fewer variations in learning related to SRL, learning gain comparisons in larger samples are desirable for future work.

While Stoich Tutor has a comparatively high degree of scaffolding, it is worth noting that all tutoring systems used in this study provide relatively high levels of instructional support, more than many other virtual learning systems. Future work could replicate our think-aloud methodology with ONA to investigate SRL during even less scaffolded problem solving (e.g., during freeform problem solving on paper). Such efforts could pose further methodological advancements to measuring the optimal level of scaffolding through methods similar to those used in this study.

The generalizability of our findings is limited as all data collection was performed during relatively short (<60 mins) think-aloud sessions aimed at studying the usability of the three ITS. In naturalistic and long-term educational contexts (e.g., college courses running over a semester), ONA analyses might yield SRL trends across longer time periods, which could be related to learning outcomes.

6 Conclusion

Self-regulated learning is crucial for learning and fundamentally temporal. Intelligent tutoring systems spanning instructional domains pose different levels of support for engaging in problem solving and desirable SRL sequences. This study has demonstrated how ONA can capture these differences and generate insight into how students differently engage with tutoring systems. We investigated fifteen college students engaging with tutoring systems with low and high degrees of scaffolding in chemistry and formal logic. We find that scaffolding may more readily allow students to enact problem-solving steps, as planning more rarely preceded enacting in highly scaffolded tutoring systems. Such active learning in tutoring is generally desirable for learning. However, scaffolds removing opportunities to learn to plan could make tutoring systems less effective if such planning is a domain learning goal. Specifically, only in formal logic, planning exhibited strong ties to enacting, potentially being a prerequisite for successful problem solving, where constructing proofs and strategy selection are instructional priorities over memorizing transformation rules. ONA can offer insights into whether a given tutoring system fulfills learning goals of a domain and whether SRL profiles of learners coincide with relevant design decisions.

Acknowledgements

Carnegie Mellon University's GSA/Provost GuSH Grant funding was used to support this research.

References

[1] Aleven, V. et al.: Example-Tracing Tutors: Intelligent Tutor Development for Non-programmers. International Journal of Artificial Intelligence in Education. 26, 1, 224–269. DOI:https://doi.org/10.1007/s40593-015-0088-2 (2016).

[2] Aleven, V. et al.: Toward Meta-cognitive Tutoring: A Model of Help Seeking with a Cognitive Tutor. International Journal of Artificial Intelligence in Education. 16, 2, 101–128 (2006).

[3] Azevedo, R. et al.: Lessons Learned and Future Directions of MetaTutor: Leveraging Multichannel Data to Scaffold Self-Regulated Learning With an Intelligent Tutoring System. Frontiers in Psychology. 13, 813632 (2022).

[4] Azevedo, R. et al.: MetaTutor: Analyzing self-regulated learning in a tutoring system for biology, 635–637 (2009).

[5] Bannert, M. et al.: Process mining techniques for analysing patterns and strategies in students' self-regulated learning. Metacognition and Learning. 9(2), 161–185 (2014).

[6] Borchers, C. et al.: Using Think-Aloud Data to Understand Relations between Self-Regulation Cycle Characteristics and Student Performance in Intelligent Tutoring Systems. 14th International Conference on Learning Analytics and Knowledge (2024).
[7] Borchers, C. et al.: What Makes Problem-Solving Practice Effective? Comparing Paper and AI Tutoring. European Conference on Technology Enhanced Learning, pp. 44–59 (2023).
[8] BouJaoude, S.,Barakat, H.: Students' problem solving strategies in stoichiometry and their relationships to conceptual understanding and learning approaches. The Electronic Journal for Research in Science & Mathematics Education. (2003).
[9] Du, J. et al.: What can online traces tell us about students' self-regulated learning? A systematic review of online trace data analysis. Computers & Education (2023).
[10] Fan, Y. et al. 2023. Dissecting learning tactics in MOOC using ordered network analysis. Journal of Computer Assisted Learning. 39(1), 154–166 (2023).
[11] Ge, X. et al.: Detangling the interrelationships between self-regulation and ill-structured problem solving in problem-based learning. Interdisciplinary Journal of Problem-Based Learning. 10,(2) (2016).
[12] Huang, X. et al.: Exploring the co‐occurrence of students' learning behaviours and reasoning processes in an intelligent tutoring system: An epistemic network analysis. Journal of Computer Assisted Learning, 39(5), 1701–1713 (2023).
[13] King, E.C. et al.: The Open-Response Chemistry Cognitive Assistance Tutor System: Development and Implementation. Journal of Chemical Education. 99(2), 546–552. DOI:https://doi.org/10.1021/acs.jchemed.1c00947 (2022).
[14] Koedinger, K. et al.: An astonishing regularity in student learning rate. Proceedings of the National Academy of Sciences. 120(3) (2023).
[15] Li, S. et al.: Examining temporal dynamics of self-regulated learning behaviors in STEM learning: A network approach. Computers & Education. 158, 103987 (2020).
[16] Long, Y. et al.: What exactly do students learn when they practice equation solving? refining knowledge components with the additive factors model. 8th International Conference on Learning Analytics and Knowledge, pp. 399–408 (2018).
[17] Long, Y., Aleven, V.: Enhancing learning outcomes through self-regulated learning support with an open learner model. User Modeling and User-Adapted Interaction. 27, 55–88 (2017).
[18] Marquart, C.L. et al. 2018. Epistemic network analysis.
[19] Munshi, A. et al.: Analysing adaptive scaffolds that help students develop self‐regulated learning behaviours. Journal of Computer Assisted Learning. 39(2), 351–368 (2023).
[20] Paquette, L. et al.: Using Epistemic Networks to Analyze Self-regulated Learning in an Open-Ended Problem-Solving Environment. In: Ruis, A.R. Lee, S.B. (eds.) Advances in Quantitative Ethnography,Springer International Publishing. pp. 185–201 (2021).
[21] Phye, G.D.: Components of Self-Regulation during Within-and Between-Domain Problem-Solving Performance (1998).
[22] Radford, A. et al.: Robust Speech Recognition via Large-Scale Weak Supervision. International Conference on Machine Learning, pp. 28492–28518 (2023).
[23] Renkl, A. et al.: How fading worked solution steps works–a cognitive load perspective. Instructional science,. 32(1), 59–82 (2004).
[24] Schmidt, H.J.: An alternate path to stoichiometric problem solving. Research in Science Education. 27, 237–249 (1997).
[25] Seufert, T.: The interplay between self-regulation in learning and cognitive load. Educational Research Review, 24, 116–129 (2018).
[26] Shapiro, A.M: Hypermedia design as learner scaffolding. Educational Technology Research and Development. 56, 1, 29–44 (2008).
[27] Tan, Y. et al.:. Ordered Network Analysis. In: Damşa, C., Barany, A. (eds.) Advances in Quantitative Ethnography, pp.101–116, Springer Nature Switzerland (2023).

[28] VanLehn, K.: The behavior of tutoring systems. International journal of artificial intelligence in education. 163), 227–265 (2006).
[29] Winne, P.H., Hadwin, A.F.: Studying as Self-Regulated Learning. Metacognition in Educational Theory and Practice, 277–304 (1998).
[30] Wu, M. et al.: Understanding detectors for SMART model cognitive operation in mathematical problem-solving process: an epistemic network analysis. International Conference on Quantitative Ethnography,pp. 314–327 (2022).
[31] Zhang, J. et al.: Using large language models to detect self-regulated learning in think-aloud protocols. Proceedings of the 17th International Conference on Educational Data Mining (2024).
[32] Zhang, Y. et al.: Can strategic behaviour facilitate confusion resolution? The interplay between confusion and metacognitive strategies in Betty's Brain. Journal of Learning Analytics. 8(3), 28–44 (2021).
[33] Zimmerman, B.J.: Attaining Self-Regulation: A Social Cognitive Perspective. Handbook of Self-Regulation 13–39 (2000).
[34] Koedinger, K.R., Aleven, V.: Exploring the assistance dilemma in experiments with cognitive tutors. Educational Psychology Review. 19, 239–264 (2007).
[35] Koedinger, K.R., Baker, R.Sj., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J.: A data repository for the EDM community: The PSLC DataShop. Handbook of educational data mining. 43, 43–56 (2010).