Li, J., Liu, J., Yuan, R., & Shadiev, R. (2022). The Influence of Socially Shared Regulation on Computational Thinking Performance in Cooperative Learning. *Educational Technology & Society*, *25* (1), 48-60.

The Influence of Socially Shared Regulation on Computational Thinking Performance in Cooperative Learning

Jiansheng Li, Jiao Liu, Rui Yuan and Rustam Shadiev*
School of Education Science, Nanjing Normal University, Nanjing, China // 2869753244@qq.com // 2776571510@qq.com // 1622529878@qq.com // rustamsh@gmail.com
*Corresponding author

(Submitted March 9, 2021; Revised July 13, 2021; Accepted July 26, 2021)

ABSTRACT: This study explores the role of socially shared regulation on computational thinking performance in cooperative learning. Ninety-four middle school students from China aged between 16 and 18 participated in this study. Forty-six students were in the experimental group, and 48 students were in the control group. Students in the experimental group learned under the socially shared regulation of learning (SSRL) condition, which included planning and goal setting, task and content monitoring, and task and content evaluation. Students in the control group learned in a traditional way. The results showed that the students in the experimental group significantly outperformed their counterparts on the midtest and posttest. Additionally, the learning gain of the experimental group was much better from the pretest to the midtest. Different subgroups in the experimental group had different learning performances, and task monitoring and content monitoring were two important SSRL processes that led to improved computational thinking performance. Our results suggest that SSRL is beneficial for learning computational thinking subjects. Throughout the process of SSRL, different groups have different learning dynamics, and task and content monitoring plays a major role in computational thinking performance.

Keywords: Socially shared regulation, Group monitoring, Cooperation, Computational thinking, Learning performance

1. Introduction

With the development of artificial intelligence, cultivating students' computational thinking has become an important matter (Angeli et al., 2016). Related research explores how computational thinking in programming can be cultivated not only on the individual level but also through nurturing such skills through cooperative learning activities (Shadiev et al., 2014). For example, McDowell et al. (2003) carried out an experimental study on pair programming and found that pair programming was better for improving students' learning outcomes (i.e., learning performance, perception and persistence) than individual programming. Ge (2014) proposed the collaborative learning model based on computational thinking for training computational thinking and applied it to middle-school classroom teaching. Those results were positive and demonstrated the positive effects of the model on computational thinking skills. Denner et al. (2014) carried out research on computational thinking through cooperative task completion and obtained similar results, i.e., cooperative learning was beneficial for learning outcome improvement, especially for students who had less programming experience.

How learning interactions occur in cooperative learning and how they affect performance are important questions in the field and ones that scholars have attempted to solve in their research. Chi et al. (2014) and Hwang et al. (2012) explained that interaction in cooperative learning occurs when students share and present different perspectives on the problem-solving process. Hadwin et al. (2017) argued that social coregulation is an important component of successful cooperation. For example, individuals in the group negotiate a common goal, adjust the goal according to their own abilities, and discuss common goals and group progress. Without the skills and willingness of individuals and groups to act cooperatively, learning interactions cannot occur (Hadwin & Oshige, 2011b; Hwang et al., 2015). Furthermore, Hadwin et al. (2011a) emphasized the monitoring process by students in learning, as it is particularly important in the regulation process. However, little is known about how group regulation occurs in cooperative learning, whether it exerts an impact on computational thinking performance, or what the main influencing factors are, and there remain many other similar unanswered questions in the field. This paper thus attempts to answer them.

2. Background of study

2.1. Computational thinking and cooperative learning

Computational thinking involves solving problems, designing systems, and understanding human behavior by drawing on the concepts fundamental to computer science (Wing, 2006). Computational thinking is reformulating a seemingly difficult problem into one we know how to solve, perhaps by reducing, embedding, transforming, or simulating (Angeli et al., 2016). In large, complex tasks or designing a large, complex system, abstraction and decomposition are often used to model relevant aspects of the problem and make it easier to deal with by choosing the appropriate representation for the problem. Such an approach is planning, learning, and scheduling in the presence of uncertainty. Computational thinking is thinking in terms of prevention, protection, and recovery from worst-case scenarios through redundancy, damage containment, and error correction (Wing, 2006).

Aho (2012) considered computational thinking to be the thought processes involved in formulating problems such that their solutions can be represented as computational steps and algorithms. Computational thinking includes (but is not limited to) the following steps (International Society for Technology in Education, 2021): (a) formulating problems in a way that enables the use of a computer and other tools to help solve them; (b) logically organizing and analyzing data; (c) representing data through abstractions such as models and simulations; (d) automating solutions through algorithmic thinking (a series of ordered steps); and (e) identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources. Therefore, computational thinking can be cultivated through problem calculation steps and algorithms. For example, Brennan and Resnick (2012) claimed that computational thinking cultivation involves the following: (1) computational thinking concepts: sequences, loops, events, parallelism, conditionals, operators, data; (2) computational thinking practices: being incremental and iterative, testing and debugging, reusing and remixing, abstracting and modularizing; and (3) computational thinking perspectives: expressing, connecting and questioning. Similarly, according to Barefootcomputing (2021), computational thinking cultivation involves two aspects: concepts (logic, algorithms, decomposition, patterns, abstraction, evaluation) and approaches (tinkering, creating, debugging, persevering, collaborating).

The simplest measure of the results of computational thinking cultivation is the computational thinking test. The goal of the computational thinking test is to assess students' ability to solve complex problems using computational thinking by asking students to solve practical problems. For example, the Bebras Tasks, which most researchers have used in computational thinking (Dagienė & Futschek, 2008), have been noted as more than likely to be a foundation for a future PISA (program for international student assessment) test in the field of computer science (Román-González et al., 2019; Yağcı, 2019).

Cooperative learning is also concerned with the cultivation of students' computational thinking. Mcdowell et al. (2003) examined the effectiveness of pair programming in programming courses and found that students who used pair programming produced better programs and were more confident in their problem solving. This result was also verified in other studies (e.g., in Denner et al., 2014). Similarly, Turchi et al. (2019) believed that cooperative game-based learning could foster students' computational thinking skills.

2.2. Socially shared regulation of learning

There are three regulation modes in collaborative learning: self-regulation, coregulation and socially shared regulation (Winne & Hadwin, 1998). Self-regulation refers to the individual's regulation of cognition, metacognition, motivation, emotion and behavior to adapt to other members of the group (Hu & Driscoll, 2013). Coregulation emphasizes the influence among individuals, which means that learners adjust their learning strategies when they interact with other members of the group (Zheng et al., 2017). Socially shared regulation emphasizes conscious, strategic and interactive planning, task formulation, reflection and adaptation within the group (Winner et al., 2011; Hadwin et al., 2011a). In the educational context, such modes are called self-regulated learning, coregulated learning and socially shared regulation of learning (SSRL).

Of particular interest to this study is SSRL, which occurs in the learning process when group members complement one another's cognitive resources; that is, they set common goals together, share responsibility for appropriate strategy to formulate goals, and coordinate changes and adjustments to optimize the problem-solving process (Miller et al., 2017). SSRL seems best to mirror egalitarian, complementary monitoring and regulation over the task, thus bringing the research closer to phenomena relevant to joint, peer-mediated learning. SSRL is committed to common regulatory activities (Vauras et al., 2003). Therefore, SSRL involves different aspects of regulation to ensure that group members remain involved and provide consistent efforts. Low-level social regulation involves the simple exchange or sharing of facts and clarification of understanding, while high-level social regulation is characterized by the use of both shared regulation and deep-level content processing. In SSRL, group members actively and cooperatively monitor developed ideas (Rogat et al., 2011). They also regulate their metacognition, cognition, motivation and behavior (Hadwin et al., 2011a). Moreover, students share multiple ideas and perspectives to be weighed and negotiated (Järvelä et al., 2013). In essence, the SSRL begins to expand regulation activities to include the negotiation and regulation of the group's collective activities. Lee (2014) identified several socially shared regulation processes in computer-supported collaborative learning: planning, goal setting, task monitoring, content monitoring, task evaluation, and content evaluation (Table 1). At the group cooperation discourse level, Volet et al. (2013) believed that all regulation activities in the group come from questions (direct or implicit questions), explanatory statements or abstracts (usually tentative) or implicit or triggered suggestions (under certain background knowledge).

Table 1. Socially shared regulation process

Code	Definition
Planning and Goal Setting	• Presenting a question as a starter for the group's plan or goal • Posting the guiding questions as a starter • Discussing plans and goals • Expressing agreement
Task Monitoring	• Verifying the progress or the completion of each guiding question • Checking the time • Correcting typos
Content Monitoring	• Providing a reason to support the responses or ideas • Checking the accuracy of the task responses
Task Evaluation	• Checking the completion of all the guiding questions
Content Evaluation	• Checking whether the group met its initial goals • Checking whether the group's views were in agreement • Evaluating whether the group completed the task

From what has been discussed above, this study uses the group cooperation process (Lee, 2014), group cooperation discourse level (Volet et al., 2013), and computational thinking definition (Brennan & Resnick, 2012) to analyze the discourse of group cooperation and to explore group regulation activities and their role. Therefore, the following research questions will be addressed in this paper:

- Does SSRL affect computational thinking performance?
- In SSRL activities, which process leads to the improvement of computational thinking performance?
- Do different subgroup dynamics in SSRL activities exert different impacts on computational thinking performance?

3. Method

3.1. Participants

The participants were 94 middle school students aged between 16 years and 18 years from a senior high school in China. They all were at the same learning grade. The students were assigned to an experimental group (n = 46) and a control group (n = 48). There were 29 boys and 17 girls in the experimental group and 24 boys and 24 girls in the control group. All participants were informed about the study and gave informed consent prior to participation in the study.

3.2. Procedure

The teaching experiment lasted for eight weeks and involved the following related aspects: data, sequences, conditionals, loops, abstracting and modularizing, testing and debugging, reusing and recreating. The teaching content of the experimental group was to help students learn basic knowledge of Python. Teaching experiments were conducted in a classroom environment, and group collaborative learning and discussion were presented through Shimo Docs. Shimo Docs (https://shimo.im/) is enterprise office service software that supports clouds and real-time collaboration. It enables multiple users to edit the same document and to have real-time discussion among learners. This software is widely used in educational institutions in China.

The experimental and control groups were taught by the instructor according to the teaching plan. During classroom teaching, textual discussion data in Shimo Docs were collected. Then, the students in each group were divided into 8 subgroups, with 5 or 6 students in each subgroup according to the pretest results and the grouping principle of heterogeneity in the same group and homogeneity in different groups. After that, each subgroup established a discussion area in the platform and named it with their own student number. We designed guidance questions for the groups to facilitate their SSRL, as shown in Table 2.

Table 2. Description of SSRL implementation process

Code	Guidance questions to the group
Planning and Goal Setting	• Asking questions about Python drawing as a starting point for the group's plan or goal. • Issuing the guidance question regarding painting and using it as the starting point for the issue. • Discussing the planning ideas for the questions to be solved.
Task Monitoring	• Verifying the progress or the completion of each guiding question of the group: What have we completed? How much more? • Checking the time: How much time does the group require to complete the drawing? • Correcting spelling mistakes: Checking the spelling of Python code.
Content Monitoring	• Do you think this answer is correct? • Do you agree/disagree with your partner's answer? Everyone must express their own views and give reasons for their ideas. They cannot simply agree with each other. Finally, they have to reach an agreement.
Task Evaluation	• Whether each group completed all the guidance questions about Python painting.
Content Evaluation	• Checking whether the groups have completed the painting pattern we initially imagined. • Checking the use of relevant concepts in all relevant works. Are these drawings related to the knowledge of (loops, conditionals, sequences) in Python? • Evaluating the content to answer the task.

At the beginning of the cooperative learning stage, we required the students in the experimental group to become familiar with the rules of group discussion, and at the end of the learning stage, we asked them to check whether their opinions were aligned. As a result, the students in each group were asked to discuss and negotiate the guidance questions in the cooperation plan and implementation table. The guidance questions in collaboration involved understanding shared tasks (e.g., to describe group learning tasks and the purpose of this task) and shared goals (e.g., to set a goal for a group task) and included common tasks and content monitoring (e.g., Is this answer correct? How well does the task match the instruction? Please explain your responses), and task and content evaluation (e.g., Have we all completed the guidance? Have we achieved our original goal?).

The following is a group content monitoring example. The content monitoring events observed in the SSRL group included monitoring content contribution and understanding, checking for evolving task responses, and monitoring the development of the summary (see Table 2). The first feature of content monitoring was that the group participated in the monitoring process equally. Everyone had to actively elaborate, ask questions and pay attention to the contribution quality of the group members and the task response they negotiated. Next, according to the task requirements, they had to check whether others had completed assignments correctly. In response, all team members were engaged in content monitoring. The following fragment is the students' answer to guidance question 3 in the study.

Guidance question 3: Which statements (loops, conditionals, sequences) should be used to solve the problem? What is the problem?

ID2: Loops, conditionals and sequences can be used.
ID6: Does this require four times loops commands?
ID42: That pentagram has five sides. It should loop five times, shouldn't it?
ID26: Yes, I think so, too.
ID19: I think it's better to loop five times or one time.

ID2: Why is it called a loop? Why loop once?
ID6: Yes, why? Who can tell me why?
ID30: All right, because when there are 5 loops, only one line can be put in the statement block. When there is 1 loop, we can put all five lines in the statement block.
ID19: Yes, that's what I mean.
ID26: What kind of looping is it? "For" or "While"?
ID30: I think it's OK to use "For" or "While" because we can achieve our goal regardless of which statement we use.
ID2: The teacher often uses the "For" loops in the group, so why not use the "While" loops?
ID6: If there is a fixed number of times, we usually use the "For" loops.
ID42: I remember the teacher said this.
ID19: You are so excellent!

Although both groups covered the same learning content in the study, the control group's cooperative learning had no monitoring requirements.

3.3. Data acquisition

First, students were tested on their computational thinking through pretest questions. Second, SSRL was divided into four processes, and the number of regulations in different processes was recorded (see Table 2). Third, a Python knowledge test was conducted in the middle of the study. Fourth, the students' computational thinking was tested again at the end of the study, i.e., posttest. The computational thinking test and a Python knowledge test both included objective questions with objective answers. The experiment was carried out eight times, and the students collaborated on the Shimo Doc in each session. Thus, a total of eight SSRL cooperative learning sessions were conducted. Each class lasted for 45 minutes. We collected SSRL behavior through Shimo Docs. The regulation behavior of students in the SSRL process was based on statistical time and objective evidence.

3.4. The test of computational thinking

The pretest and posttest items were developed based on the Bebras tasks. According to Dagienė and Futschek (2008), the Bebras tasks comprise a set of activities designed within the context of the Bebras International Contest, which is a motivation competition in informatics and computer literacy for students of the lower, middle and upper levels of secondary school. Scholars argue that the Bebras tasks are a valid assessment tool and reliably measure CT skills, especially those that need to be transferred and projected to solve "real-life" problems. For this reason, the Bebras tasks have been widely used by scholars in the educational context (Román-González et al., 2019; Yağcı, 2019). The test of this study consisted of 20 multiple-choice questions. The test was divided into two parts: calculation concept and calculation practice. The calculation concept part tested how well students understood the basic concepts in the course and involved sequences (three questions), loops (two questions), events and parallelism (two questions), conditionals (three questions), operators (two questions) and data (two questions). The calculation practice part sought to test how well students understood practical operations in the course and involved incremental and iterative (two questions), testing and debugging (two questions), reusing and recreating (one question), and abstracting and modularizing (one question) items. A few examples of the tests are presented in the Appendix.

3.5. The test of Python knowledge

A midtest was also carried out in the middle of the experiment. Python was chosen as a programming tool in the study because of its interactive environment and its convenience in allowing beginner programmers to write meaningful but nontrivial programs within a short time (Maria & Tsiatsos, 2017). The test questions were developed by experienced grade instructors based on the knowledge objectives of the course and piloted with a few groups of students beforehand. The test items were of medium difficulty and met the Chinese middle-school information technology curriculum requirements. The test items involved knowledge of the calculation concept and calculation practice, including five fill-in-the-blank questions, ten multiple-choice questions, one programming question (open, the answer was not unique) and one short-answer question.

3.6. The coding framework of SSRL monitoring

Two kinds of monitoring – one in which each group member is responsible for regulating his/her own learning (SRL monitoring) and one in which group members regulate the learning process together (SSRL monitoring) – play an important role in the process of group cooperation (Zheng et al., 2019). In this experiment, group cooperative learning was arranged based on SSRL. Table 3 presents the coding framework for coding online texts.

In Table 2, SSRL can be distinguished by the regulation of object operations, the operations being performed (Malmberg et al., 2017), and activities that include SSRL planning, SSRL tasks, SSRL statements and progress. Students engage in SSRL tasks and planning if they activate their personal knowledge and consider personal behavior. The SSRL task and SSRL planning were represented by statements centered on group tasks and group actions, while SSRL progress involved the time management, conflict resolution and mutual understanding that guided the whole team's efforts.

Table 3. The coding framework of SRL and SSRL (Zheng et al., 2019)

Monitoring	Category	Description	Examples[1]	Features[2]
SRL monitoring	SRL task	Reviewing the prior knowledge required for the task	Well, I need to use the loops	I am… I need…
	SRL planning	Considering personal behavior	Let me see the program I wrote again	Let… I…I think…
	SRL statement	Putting forward their own view of the task		
	SRL progress	The current progress of one person and the whole team		
SSRL monitoring	SSRL planning	Plan setting and setting the purpose of the team	What's the target of our group? What's the purpose?	Let's… We need… Who?
	SSRL task	The next action to complete the task, form a statement or set of statements for other team members		Why? Please tell me…
	SSRL statement	Elaborating ideas and making your reasoning work for the team. Do you agree with others? Why? What's the reason?	I don't agree with you because conditionals make the procedure simple.	
	SSRL progress	Have you achieved your original goal?	Yes	

Note. [1]Examples and features were derived from content created by the participants based on recommendations of Zheng et al. (2019). [2]Features of regulatory activities at the individual or group level.

4. Results

4.1. Does SSRL affect computational thinking performance?

First, we explored whether the two groups were equal in terms of their basic demographic characteristics and prior knowledge. The results of *t*-tests and chi-square analysis demonstrated that the two groups did not differ significantly ($p > .05$) in average age or the proportion of boys and girls across both groups. The first column of Table 4 shows the mean and standard deviation values of the two groups for the tests. According to *t*-test analyses ($t = 1.29$, $p = .20$), the two groups were not significantly different in their pretest scores. Therefore, we concluded that the groups were equivalent in basic demographic characteristics and prior knowledge.

Next, we explored whether the groups differed in computational thinking performance. The second row of Table 4 shows the mean and standard deviation values for the midtest for the two groups. The average score for the experimental group was 62.63, while that for the control group was 44.06, and this difference was significant according to the *t*-test ($t = 4.27$, $p = .00$). To explore the impact of pretest scores on midtest scores, we conducted an analysis of covariance (ANCOVA) with pretest scores as a covariate. ANCOVA results showed that students in the experimental group scored significantly higher than students in the control group on the midtest, $F(1,91) = 4.937$, MSE = 316.210, $p = .029$, $d = 0.308$.

Table 4. Test performance for the two groups

Test	Experimental group		Control group	
	M	*SD*	*M*	*SD*
Pretest	52.97	10.80	49.78	9.78
Midtest	62.63	18.99	44.06	17.06*
Posttest	65.79	26.52	51.56	18.98*

Note. $^{*}p < .05$.

The third row of Table 4 shows the mean and standard deviation values on the posttest for the two groups. In the posttest, the computational thinking scores of the experimental group and the control group were 65.79 and 51.56, respectively, and this difference was significant according to the *t*-test (t = 2.61, p = .01). Similarly, to determine whether the pretest and midtest scores affected the posttest score, we conducted two ANCOVA tests. ANCOVA with pretest scores as a covariate showed that students in the experimental group scored significantly higher than students in the control group on the midtest, $F(1,91) = 6.658$, MSE = 473.237, p = .011, d = 1.023. However, ANCOVA with midtest scores as a covariate revealed that the experimental and control groups did not perform significantly differently from one another on the posttest, $F(1,91) = 3.816$, MSE = 425.513, p = .54, d = .608.

Thus, the major empirical finding in this study is that the students in the experimental group had higher learning gains than the students in the control group from the beginning of the study to the middle. However, the learning gain from the middle to the end of the study was not as large as that from the beginning to the middle. This finding suggests that the effect of SSRL on learning gains mainly occurs from the beginning to the middle of a course. That is, the regulation of SSRL was more effective during the first half of the experiment. Therefore, in the second half of the experiment, the regulatory effect was not as obvious.

4.2. In SSRL activities, which process leads to the improvement of computational thinking performance?

Based on Table 2, SSRL includes five processes: planning and goal setting, task monitoring, content monitoring, task evaluation, and content evaluation. In the experimental process, we combined the task evaluation with the content evaluation, and for this reason, we had four SSRL processes in the study. In these four different processes, students' monitoring time was tallied and then analyzed. Analysis of variance (ANOVA) results for experimental students' computational thinking posttest performances with respect to the four SSRL processes are shown in Table 5. According to the results, there were significant improvements in the task and content monitoring processes, $p < .05$. However, no statistically significant results were obtained for the planning and goal setting and the evaluation monitoring processes. Therefore, in the process of SSRL, monitoring (i.e., task and content monitoring) plays a major role in the learning performance.

Table 5. ANOVA results for learning performance with respect to four SSRL processes

Group	Process	*F*	*Sig.*
Experimental group	Planning and Goal Setting	2.395	.081
	Task Monitoring	3.882	.011
	Content Monitoring	3.335	.014
	Evaluation Monitoring	0.871	.532

4.3. Do different subgroup dynamics in SSRL activities have different impacts on computational thinking performance?

To test the differences in the computational thinking performance of each subgroup, ANOVA statistical tests were carried out. The results (Table 6) showed that computational thinking performance among subgroups was significantly different, F = 4.495, p = .001. That is, some subgroups had high scores, whereas other subgroups had low scores. For example, subgroup #2 had a mean value of 97.500, whereas subgroup #7 had a mean value of 38.333.

There are two possible reasons for such differences. First, the difference can be accounted for by how students cooperated during SSRL activities. The average frequency of subgroup regulation in SSRL was 13.667 (*SD* = 4.5019), 16.833 (*SD* = 1.7224), 10.333 (*SD* = 4.5898), 9.333 (*SD* = 5.5737), 12.333 (*SD* = 4.5461), 10.833 (*SD* = 5.2964), 4.500 (*SD* = 4.6797), and 6.250 (*SD* = 5.5050). After comparing the frequency values of the two subgroups with the best and the worst scores (i.e., subgroups #2 and #7) using *t*-tests, we found (see Table 7) that there was a significant difference, $t = 6.330$, $p = .001$.

Table 6. The computational thinking performance among experimental subgroups

Group	*N*	Mean	*SD*	*F*	*p*
1	6	72.500	8.512	4.495	.001
2	6	97.500			
3	6	59.167			
4	6	61.333			
5	6	65.000			
6	6	67.500			
7	6	38.333			
8	4	40.000			

Table 7. The difference between subgroups #2 and #7 in SSRL activities

Subgroup	Mean	SD	*t*	*Sig.*
2	16.833	1.722	6.330	.001
7	4.500	4.680		

Second, the difference can also be accounted for by how students participated in SSRL activities. A *t*-test was carried out to compare the four SSRL processes between the two subgroups (i.e., #2 and #7), and the results are shown in Table 8. There was a significant difference between subgroups #2 and #7 in the four processes. T and p values for each process were $t = 3.742$, $p = .004$ (planning and goal setting), $t = 4.472$, $p = .001$ (task monitoring), $t = 7.593$, $p = .000$ (content monitoring), and $t = 3.803$, $p = .003$ (evaluation monitoring). The greatest difference between the two subgroups was in content monitoring.

Table 8. The differences between subgroups #2 and #7 in the four SSRL processes

Process	Subgroup	*M*	*SD*	*t*	*Sig.*
Planning and goal setting	2	3.17	0.983	3.742	.004
	7	.83	1.169		
Task monitoring	2	3.00	0.894	4.472	.001
	7	1.00	0.632		
Content monitoring	2	6.00	0.632	7.593	.000
	7	1.33	1.366		
Evaluation monitoring	2	4.33	0.516	3.803	.003
	7	1.33	1.862		

In summary, there are differences in how students cooperate and participate in SSRL activities. Those who cooperated more had better scores. In addition, those who participated in SSRL activities more actively (especially in content monitoring) had better scores.

5. Discussion

5.1. Empirical contributions

The findings of this research demonstrated that the students in the experimental group significantly outperformed the students in the control group on the midtest and posttest. When students were engaged in SSRL activities, their task monitoring and content monitoring processes improved. We also found that the learning gain of the experimental group was significant from the pretest to the midtest.

The results of this study suggest that SSRL activities were beneficial for computational thinking performance. That is, SSRL involves different aspects of learning regulation to ensure that group members remain involved and provide consistent efforts during their learning process. As a result, these students' performance was much better than that of their counterparts. Our findings are supported by the related literature. For example, students in the experimental group were engaged in SSRL activities that advanced their learning outcomes (Hadwin et al., 2011a; Järvelä et al., 2013; Panadero & Järvelä, 2015; Rogat et al., 2011). Throughout the learning process, SSRL includes several processes, such as planning, monitoring and evaluation, in which the students are engaged from the beginning to the completion of their learning process. Through this process, students know what to learn, what homework to complete and how to evaluate their learning progress. Therefore, SSRL improves the pertinence and effectiveness of students' learning through group planning and monitoring as well as final evaluation. As a result, academic performance can be improved with SSRL.

Our results also suggest that regulation in SSRL is more effective during the first half of computational thinking learning. This finding implies that in the early stage of cooperative learning, individuals are not familiar with one another and need to adapt to the cooperative nature of learning. Group cooperative learning regulates individual behavior through engagement and group supervision to complete cooperative tasks and improve academic performance. At the later stage, such group supervision activities become increasingly less obvious. This research result verifies the views of Panadero and Järvelä (2015). These scholars stated that SSRL can contribute to students' learning performance and that coregulation occurs in some periods in groups. This would be the case as groups progress through different phases of their collaboration and do not always socially share regulations of learning (Panadero & Järvelä, 2015). In our research, coregulation occurred in the early stage.

In the process of SSRL, monitoring (i.e., task and content monitoring) plays a major role in student performance. This finding is consistent with the hypothesis that socially shared plans, tasks, and content are important factors for successful collaborative learning (Schoor & Bannert, 2012). The groups with higher learning outcomes tended to participate in socially shared tasks and socially shared content and had more interactive behaviors in completing the tasks. Successful teams put more effort into SSRL planning and task analysis. In the learning process, students need to monitor their learning behavior and learning outcomes, which can be facilitated by group work. The group members regulate learning through discussion, planning, implementation, monitoring and evaluation of learning. This study results prove once again that good regulation of learning is necessary, especially for the completion of learning tasks and the improvement of academic performance.

5.2. Theoretical contributions

There were also differences among subgroups in the SSRL performed. The supervision activities of the more successful groups were relatively stable and occurred more frequently, which was similar to the results of other studies (e.g., Järvelä et al., 2016). The reason for the difference among groups may be accounted for by the following factors. One reason is group norms. In the implementation of SSRL, students had a unified concept and executed it according to the needs of the task, which effectively promoted the generation of students' individual cognitive behavior. Moreover, group norms were beneficial in stimulating group motivation, exerted a positive and significant impact on group cognition, and promoted students' academic performance. In cooperative learning, with the enhancement of the awareness of supervision activities, students' group consciousness gradually formed. The shared learning rules and regulations of group members improved students' communication and cognitive awareness and cultivated students' self-regulation and social regulation. This result is similar to the study of Azevedo (2014). The other possible reason is group cognitive responsibility. In cooperative learning, social regulation ensures or enhances individual responsibility and supports behavior by defining tasks, scheduling and monitoring group processes and building mutual trust (Fransen et al., 2011). Therefore, in teaching, we can improve the instructional effect by arranging group cooperative learning. In particular, in group cooperative learning, team members can be asked to clarify their learning tasks first, plan the learning process, monitor their learning progress, and reflect on and evaluate the whole learning process. In this way, group cooperative learning can become more meaningful, and learning results can be improved. Furthermore, such an arrangement of the learning process will help improve trust among group members and straighten their collaboration.

5.3. Limitations and future directions

Some limitations of this research need to be acknowledged. The experimental data of this study came from the learning content and the learning object, which is specific, and the amount of data collected was limited. The computational thinking scores were calculated as total scores. In future research, computational thinking can be compared in terms of the calculation concept (7 aspects such as data, cycle and condition) and calculation practice (4 aspects such as increment and iteration), and respective conclusions can be drawn. In addition, the records generated by the team can also be used to improve the accuracy of research results by using sequence mining or process mining techniques (Winne & Philip, 2015). The focus on how SSRL appears in time, how it is triggered, and how SSRL may fluctuate in the process of participation needs to be further explored.

References

Aho, A. V. (2012). Computation and computational thinking. *Computer Journal, 55*(7), 832-835.

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Educational Technology & Society, 19*(3), 47-57.

Azevedo, R. (2014). Issues in dealing with sequential and temporal characteristics of self- and socially-regulated learning. *Metacognition and Learning, 9*(2), 217-228.

Barefootcomputing (2021). *C.A.S. Computational thinking.* Barefootcomputing. http://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/computational-thinking/

Brennan, K., & Resnick, M. (2012, April). *New frameworks for studying and assessing the development of computational thinking*. Paper presented at annual American Educational Research Association meeting, Vancouver, BC, Canada.

Chi, M. T. H., & Wylie, R. (2014). The ICAP framework: Linking cognitive engagement to active learning outcomes. *Educational Psychologist, 49*(4), 219-243.

Dagienė, V., & Futschek, G. (2008). Bebras international contest on informatics and computer literacy: Criteria for good tasks. In *International conference on informatics in secondary schools-evolution and perspectives* (pp. 19-30). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-69924-8_2

Denner, J., Werner, L., & Campe, S., & Ortiz, E. (2014). Pair programming: Under what conditions is it advantageous for middle school students? *Journal of Research on Technology in Education, 46*(3), 277-296. https://doi.org/10.1080/15391523.2014.888272

Fransen, J., Kirschner, P. A., & Erkens, G. (2011). Mediating team effectiveness in the context of collaborative learning: The Importance of team and task awareness. *Computers in Human Behavior, 27*(3), 1103-1113.

Hadwin, A. F., Järvelä, S., & Miller, M. (2011a). Self-regulation, co-regulation, and socially shared regulation: Exploring perspectives of social in self-regulated learning theory. *Teachers College Record, 113*, 240-264.

Hadwin, A. F., & Oshige, M. (2011b). Self-regulation, coregulation, and socially shared regulation: Exploring perspectives of social in self-regulated learning theory. *Teachers College Record, 113*, 240-264.

Hadwin, A. F., Järvelä, S., & Miller, M. (2017). Self-regulation, co-regulation, and shared regulation in collaborative learning environments. In D. H. Schunk & J. A. Greene (Eds.), *Handbook of self-regulation of learning and performance* (pp. 83–106). Routledge/Taylor & Francis Group.

Hu, H., & Driscoll, M. P. (2013). Self-Regulation in e-learning environments: A Remedy for community college? *Educational Technology & Society, 16*(4), 171–184.

Hwang, W. Y., Hsu, J. L., Shadiev, R., Chang, C. L., & Huang, Y. M. (2015). Employing self-assessment, journaling, and peer sharing to enhance learning from an online course. *Journal of Computing in Higher Education, 27*(2), 114–133.

Hwang, W. Y., Shadiev, R., Wang, C. Y., & Huang, Z. H. (2012). A Pilot study of cooperative programming learning behavior and its relationship with students' learning performance. *Computers & Education, 58*(4), 1267-1281.

Ge, M. (2014). 基于计算思维的协作学习模式在中学信息技术课程中的实践与研究 [A Study about the practice of collaborative learning mode based on computational thinking in middle school information technology curriculum]. https://cdmd.cnki.com.cn/Article/CDMD-10718-1014402627.html

Järvelä, S., & Hadwin, A. F. (2013). New frontiers: Regulating learning in CSCL. *Educational Psychologist, 48*(1), 25-39.

Järvelä, S., Malmberg, J., & Koivuniemi, M. (2016). Recognizing socially shared regulation by using the temporal sequences of online chat and logs in CSCL. *Learning and Instruction, 42*, 1-11.

International Society for Technology in Education (2021). *Computational Thinking Competencies*. https://www.iste.org/standards/computational-thinking

Lee, A. (2014). *Socially shared regulation in computer-supported collaborative learning.* (Publication No. 3641808) [Doctoral dissertation, Rutgers University-Graduate School-New Brunswick]. ProQuest Dissertations and These Global.

Malmberg, J., Järvelä, S., & Järvenoja, H. (2017). Capturing temporal and sequential patterns of self-, co-, and socially shared regulation in the context of collaborative learning. *Contemporary Educational Psychology, 49*, 160-174.

Maria, S., & Tsiatsos, T. (2017). Case study: Integrating computational thinking into the introductory course of computer science via the use of the programming language Python. In *Interactive Mobile Communication, Technologies and Learning* (pp. 531-541). https://doi.org/10.1007/978-3-319-75175-7_52

Mcdowell, C., Werner, L., & Bullock, H. E., & Fernald, J. (2003). The Impact of pair programming on student performance, perception and persistence. In *International Conference on Software Engineering* (pp. 602-607). https://doi.org/10.1109/ICSE.2003.1201243

Miller, M., Järvelä, S., & Hadwin, A. (2017). Self-regulation, co-regulation, and shared regulation in collaborative learning environments. In D. H. Schunk & J. A. Greene (Eds.), *Handbook of Self-Regulation of Learning and Performance* (pp. 83–106). New York, NY: Routledge.

Panadero, E., & Järvelä, S. (2015). Socially shared regulation of learning: A Review. *European Psychologist, 20*(3), 190-203.

Rogat, T. K., & Linnenbrink-Garcia, L. (2011). Socially shared regulation in collaborative groups: An Analysis of the interplay between quality of social regulation and group processes. *Cognition and Instruction, 29*(4), 375-415.

Román-González, M., Moreno-León, J., & Robles, G. (2019). Combining assessment tools for a comprehensive evaluation of computational thinking interventions. In *Computational Thinking Education* (pp. 79-98). https://doi.org/10.1007/978-981-13-6528-7_6

Schoor, C., & Bannert, M. (2012). Exploring regulatory processes during a computer-supported collaborative learning task using process mining. *Computers in Human Behavior, 28*(4), 1321-1331.

Shadiev, R., Hwang, W. Y., Yeh, S. C., Yang, S. J. H., Wang, J. L., Lin, H., & Hsu, G. L. (2014). Effects of unidirectional vs. reciprocal teaching strategies on web-based computer programming learning. *Journal of Educational Computing Research, 50*(1), 67-95.

Turchi, T., Fogli, D., & Malizia, A. (2019). Fostering computational thinking through collaborative game-based learning. *Multimedia Tools and Applications, 78*(10), 13649-13673.

Vauras, M., Iiskala, T, & Kajamies, A., Kinnunen, R., & Lehtinen, E. (2003). Shared-regulation and motivation of collaborating peers: A Case analysis. *Psychologia, 46*(1), 19-37. https://doi.org/10.2117/psysoc.2003.19

Volet, S., Vauras, M., Khosa, D., & Iiskala, T. (2013). Metacognitive regulation in collaborative learning: conceptual developments and methodological contextualizations. In *Interpersonal Regulation of Learning and Motivation* (pp. 81-115). Routledge.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.

Winne, & Philip H. (2015). What is the state of the art in self-, co- and socially shared regulation in CSCL? *Computers in Human Behavior, 52*, 628-631.

Winne, P. H., & Hadwin, A. F. (1998). Studying as self-regulated engagement in learning. In D. Hacker, J. Dunlosky, & A. Graesser (Eds.), *Metacognition in Educational Theory and Practice* (pp. 277–304). Hillsdale, NJ: Erlbaum

Winner, P. H. (2011). A Cognitive and metacognitive analysis of self-regulated learning. In B. J. Zimmermanand & D.H. Shunk (Eds.), *Handbook of Self-Regulation of Learning and Performance* (pp. 15-32). New York, NY: Routledge.

Yağcı, M. (2019). A Valid and reliable tool for examining computational thinking skills. *Education and Information Technologies, 24*(1), 929-951.

Zheng, L., Li, X., & Huang, R. (2017). The Effect of socially shared regulation approach on learning performance in computer-supported collaborative learning. *Educational Technology & Society, 20*(4), 35–46.

Zheng, J., & Xing, W. & Zhu, G. (2019). Examining sequential patterns of self- and socially shared regulation of STEM learning in a CSCL environment. *Computers & Education, 136*, 34-48.

Appendix

A few examples of test questions

A. Calculation concept

A1. Sequences
To have time for dinner, Sary (S) needs to communicate with five classmates: Alice (A), Bean (B), Cary (C), David (D) and Emil (E). S can communicate with E immediately. However, there are a few things that need to be clear about communicating with her classmates:
1. Before she talks to D, she must communicate with A.
2. Before she talks to B, she must communicate with E.
3. Before she talks to C, she must communicate with B and D.
4. Before she talks to A, she must communicate with B and E.
Question: If, according to the above requirements, Sary hopes to chat with all the above friends, what order should she follow?
A: E, B, A, D, C
B: B, E, A, D, C
C: E, A, B, D, C
D: E, A, B, D, C

A2. Data and operators
Grandma Fox does not know how to use a computer. However, she had to set a password for her mailbox to keep it safe, and Grandma Fox must follow the following requirements to set a password:
1. A minimum of 2 letters must be capitalized.
2. There must be more English letters than Arabic numerals.
3. Minimum 3 special characters (neither English nor numeric).
Question:
Which of the following passwords conforms to the above rules?
A: PearL@mb2953?
B: ##RedM3rgan-2688
C: R5#X&v73r68!?
D: *h9n3ytR33*§!

A3. Conditionals
The Styx operating system has a feature in which a poisoned Styx operating system computer will return an incorrect answer to any question received from the Internet. If it is asked, "Are you infected with the virus?” It will answer "no." An uninfected computer always answered correctly with “no” when asked, "Have you been infected?" Styx’s information engineers tested Styx servers and laptops over the Internet.
Question:
In which of the following sentences is a message returned only by a poisoned server?
A: I am a poisoned server.
B: I am not a poisoned server.
C: I am a poisoned laptop.
D: I am not a poisoned laptop.

B. Calculation practice

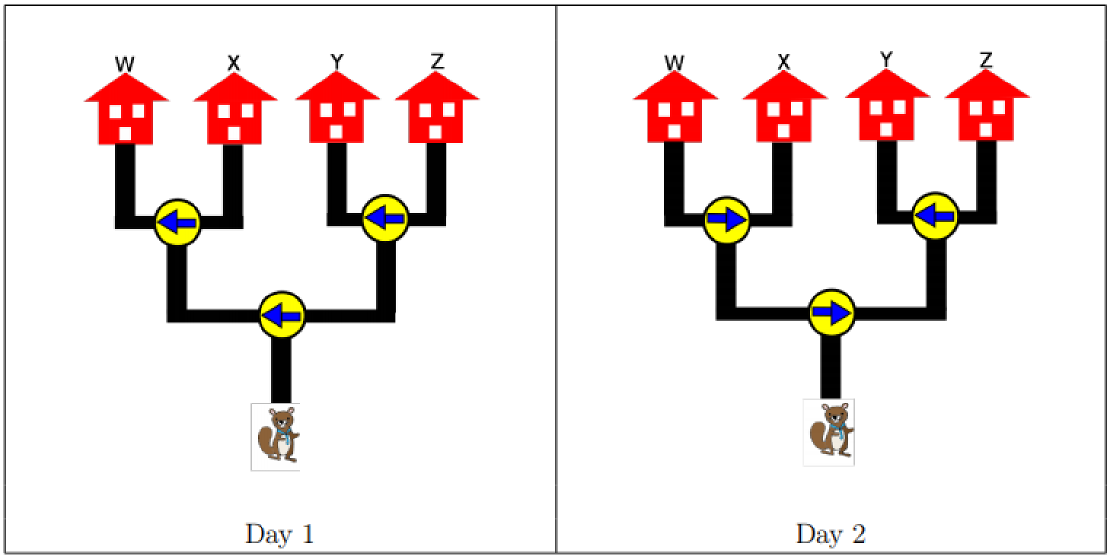
B1. Reusing and recreating
You have a beautiful paper airplane (one of those things we often fold), and then you need to transform or recreate it into a new shape (airplane or other object). What would you do? (multiple choice)
A: Take the old one apart, fold it and try a new idea over and over again.
B: Take the folded plane and make a slight change.
C: Follow the feeling and try new folding methods again and again.
D: Do not want to change the original, feel the original plane has been very good.

B2. Abstracting and modularizing
Class task: Let’s use information technology to create an "explore the moon" handwritten newspaper. In what way do you think we can best accomplish this task? (multiple choice)

A: Break down the handwritten newspaper into several columns and finish it one by one.
B: Use search engines to search for good templates and then modify them.
C: Design and make based on the production experience in class.
D: Ask teachers for help and let them guide me in completing it.

B3. Incremental and iterative

Mr. Beaver has 4 friends living in different villages, and he plans to visit one of these friends every afternoon. Mr. Beaver will follow the direction of the arrow on signs at each intersection. Initially, all arrows point to the left road. When passing an intersection, Mr. Beaver switches the arrow to the opposite direction. For example, on Day 1, Mr. Beaver takes the road on the left at the first intersection, takes the left road on the second intersection, and reaches Village W. On Day 2, Mr. Beaver turns right at the first intersection, then left at the second intersection, and arrives at Village Y.



Question:
Which village will Mr. Beaver visit on day 30?
A: Village W
B: Village X
C: Village Y
D: Village Z