Elinvo (Electronics, Informatics, and Vocational Education)
Vol 9, No. 2, 2024, pp. 309-320
ISSN 2580-6424 (printed) | ISSN 2477-2399 (online)

Students' Cognitive Load on Computer Programming Instructional Process Using Example-Problem-Based Learning and Problem-Based Learning Instructional Model at Vocational High School

Admaja Dwi Herlambang[1*], Muhammad Rifqy Ramadana[1], Satrio Hadi Wijoyo[1], Muneeroh Phadung[2]

[1]Universitas Brawijaya, Malang, Indonesia
[2]Yala Rajabhat University, Amphoe Mueang, Thailand

Article Info

Article history:
Received January 19, 2023
Revised August 27, 2024
Accepted September 03, 2024

Keywords:
Cognitive load theory; example-problem-based learning; vocational education; computer programming instruction

Abstract

This paper fills an essential gap in applying cognitive load theory in teaching computer programming within vocational settings. It is an important area to consider for improving students' learning processes who intend to enter the rapidly changing technology sector. This study assessed the distinct impacts of the instructional paradigms, specifically Example-Problem-Based Learning (EPBL) and Problem-Based Learning (PBL), on students' cognitive loads upon framing an iterative structure lesson on computer programming. Vocational programming education is chosen for this purpose because vocational education faces unique challenges in integrating practical skills development with theoretical understanding, and programming tasks involve high cognitive demands. In a quasi-experimental design, 68 vocational high school students were assigned to an EPBL (n = 34) and a PBL (n = 34) group. The measurement of ICL was operationalized by RPI, the ECL by ME, and the GCL by LO. The relationship among the various components of the cognitive load was tested using the Spearman correlation test. There are significant differences in the profile of cognitive load between the two groups: the EPBL group was always associated with the lower ECL and higher GCL. In other words, the present study is original because it systematically compares EPBL with PBL in the context of vocational programming education and provides empirical evidence based on instructional design decisions. These findings suggest a further refinement of the CLT within domain-specific contexts and practical guidelines for optimizing instructional strategies in computer programming education in vocational schools.

Corresponding Author:
Email: herlambang@ub.ac.id

INTRODUCTION

Cognitive Load Theory (CLT) has emerged as a pivotal framework in educational psychology, offering profound insights into the intricate processes of human cognition and learning. Developed by Sweller [1], [2], CLT posits that the human cognitive architecture, particularly working memory, has limited capacity when dealing with novel information. This limitation necessitates careful consideration in instructional design to optimize learning outcomes. Over the past three decades, CLT has significantly influenced educational research and practice, providing a robust theoretical foundation for understanding the cognitive demands placed on learners during complex learning tasks [3]. The fundamental premise of CLT is that cognitive load can be categorized into three distinct types: intrinsic, extraneous, and germane. Intrinsic cognitive load is inherent to the complexity of the learning material itself, extraneous cognitive load is imposed by suboptimal instructional design, and germane cognitive load represents the cognitive resources allocated to schema construction and automation [2], [4]. This triarchic model has been instrumental in guiding instructional designers and educators to develop effective learning strategies that minimize extraneous load while optimizing germane load, thereby enhancing learning efficiency and effectiveness.

Recent advancements in CLT research have expanded its application to various educational domains, including mathematics, science, and language learning. However, there remains a significant gap in the literature regarding applying CLT principles to computer programming education, particularly in vocational settings. Computer programming, characterized by its high-element interactivity and abstract nature, presents unique cognitive challenges to learners. The complexity of programming tasks, which often require simultaneous consideration of multiple concepts and procedures, aligns closely with the high intrinsic cognitive load described in CLT [1], [5]. The paucity of research on cognitive load in programming education is particularly pronounced in vocational education contexts. Vocational education, emphasizing practical skills and industry-relevant knowledge, presents a unique learning environment that warrants specific investigation through the lens of CLT. Integrating CLT principles into vocational programming education can significantly enhance instructional effectiveness and student learning outcomes in this critical sector of education [6], [7], [8].

This study aims to extend the application of CLT to computer programming education in vocational settings by investigating the differential effects of two distinct instructional models: Example-Problem-Based Learning (EPBL) and Problem-Based Learning (PBL). While both models are rooted in constructivist learning theory, they diverge significantly in managing cognitive load. EPBL, which integrates worked examples with problem-solving tasks, is hypothesized to reduce extraneous cognitive load by providing scaffolding that aligns with CLT’s emphasis on schema acquisition and automation [2], [9], [10]. Conversely, PBL, with its focus on student-centered, inquiry-based learning through complex problem exploration, potentially imposes higher initial cognitive demands but may enhance germane load through active knowledge construction [10], [11], [12]. This comparative analysis aims to elucidate how these contrasting approaches differentially affect the three components of cognitive load—intrinsic, extraneous, and germane—as defined by CLT. By examining students’ cognitive load during an iterative structure lesson in computer programming, we seek to provide empirical evidence on how each model’s unique features interact with the cognitive architecture described in CLT. This investigation is particularly pertinent in vocational programming education, where the high-element interactivity of programming tasks presents significant challenges in cognitive load management. The theoretical differentiation between EPBL and PBL concerning CLT lies in their contrasting knowledge acquisition and skill development mechanisms. EPBL’s use of worked examples is theorized to reduce the extraneous load by providing explicit problem-solving schemas, thereby freeing cognitive resources for schema acquisition [13]. In contrast, PBL’s emphasis on self-directed problem-solving may initially increase cognitive load but potentially leads to more robust schema construction through effortful processing [1], [14]. This study aims to empirically test these theoretical predictions, contributing to a more nuanced understanding of how different instructional strategies can be optimized to manage cognitive load in complex learning domains such as computer programming.

It systematically compares the added value created by Example-Problem-Based Learning and Problem-Based Learning within the framework of vocational programming education. That is why it was conducted based on a robust theoretical framework: Cognitive Load Theory. This work is entirely based on core theoretical considerations that generate an appropriate need for more research in this domain. CLT in programming education, therefore, stands on a theoretically firm ground because there is a high element of interactivity in programming tasks. Sweller et al. take element interactivity as one of the critical determinants of intrinsic cognitive load [1], [2], [15]. Indeed, programming entails several complicated syntaxes, logical structures, and abstract notions, typical of domains where element interactivity is sharp. We focus on expanding the CLT's explanatory power in a highly interactive learning domain by examining how EPBL and PBL differentially affect intrinsic, extraneous, and germane cognitive load levels. Because EPBL and PBL differ in their approaches to managing cognitive load, the theoretical basis is rooted in their divergent approaches for comparing EPBL and PBL. EPBL, conceptually based on the worked example effect [9], [13], theoretically reduces extraneous load by providing explicit problem-solving schemas. On the other hand, PBL, conceptually rooted in constructivist learning theory [16], [17], [18], [19], [20], may facilitate germane load through active knowledge construction but could lead to higher initial cognitive demands. The theoretical tension provides practical empirical research in programming education where a balance is called for between scaffolding and its opposite methodology, discovery learning. Thirdly, CLT is theoretically important in focusing on vocational programming education; learners experience specific cognitive demands in this setting. It is complex to develop practical skills and understand the theory concerning those skills; it is strategically positioned that CLT should extend such complexity. According to the argument presented by Paas et al., CLT offers a framework to optimize the instructional design of complex learning settings, and therefore, its full relevance finds its place in meeting the challenge posed by vocational education [21]. Furthermore, our attention to recurrent structures in programming echoes CLT's attention to the automated acquisition of schemas. Loop structures are central and demanding, making them fertile ground for investigating how different pedagogic treatments affect mental models' development in programming. It resonates with the recent theoretical development in CLT, placing a premium on the role of long-term memory structures in the organization of cognitive load [2]. We also hope to further advance the CLT application in programming education and contribute to the ongoing theoretical debate about effective instructional design in complex high-element interactivity domains with empirical evidence regarding the differential effects of EPBL and PBL on cognitive load within this context. Theoretically, implications can be drawn from this on how cognitive load is managed in working with vocational education, particularly in preparing students for high cognitive demand and rapidly changing fields, such as computer programming.

Furthermore, our research addresses the call for more domain-specific applications of CLT, as highlighted by recent reviews of the field. Leppink emphasized the need for CLT research to move beyond general principles and explore how cognitive load manifests in specific subject areas and learning contexts [22]. By focusing on vocational programming education, our study contributes to this growing body of domain-specific CLT research, potentially uncovering unique insights that may not be apparent in more general studies of cognitive load. The comparison between EPBL and PBL in our study also contributes to the ongoing debate about the efficacy of different instructional approaches in managing cognitive load. While worked examples, a key component of EPBL, effectively reduces extraneous cognitive load for novice learners [2], [6], [23], [24], the effectiveness of PBL in managing cognitive load remains controversial. Some studies suggest that PBL may impose high extraneous cognitive load, particularly for novice learners [1], while others argue that PBL can enhance germane cognitive load by promoting deep learning and schema construction [25]. Our study aims to shed light on this debate within the specific context of vocational programming education.

The methodological approach of our study, which operationalizes cognitive load through measures of Receiving and Processing Information (RPI), Mental Effort (ME), and Learning Outcomes (LO), provides a comprehensive assessment of the cognitive demands imposed by EPBL and PBL. This multifaceted approach to measuring cognitive load aligns with recent recommendations in the field for more nuanced and context-specific assessments of cognitive load [4], [8]. By examining the relationships between these different aspects of cognitive load, our study aims to provide a more holistic understanding of how EPBL and PBL affect students' cognitive processes during programming instruction. In conclusion, our research addresses a significant gap in the literature by extending CLT to the domain of vocational programming education and comparing the effectiveness of EPBL and PBL in managing cognitive load. This study contributes to the theoretical understanding of cognitive load in programming education and has practical implications for instructional design in vocational settings. By providing empirical evidence on the cognitive effects of different instructional approaches, our research aims to inform evidence-based practices in programming education, ultimately enhancing students' learning experiences and outcomes in vocational programs.

METHODS

This study employed a quasi-experimental design with a quantitative approach to investigate the differential effects of Example-Problem-Based Learning (EPBL) and Problem-Based Learning (PBL) instructional models on students' cognitive load during an iterative structure lesson in computer programming. The quasi-experimental nature of the design, which endorses the use of quantitative approaches, opens the possibility of directly comparing EPBL and PBL instructional models within strictly controlled premises. Two clear groups, an experimental group and a controlled group, further allow findings to be teased apart into the effects of each model on the students' cognitive load. Moreover, this study's operationalization of cognitive load uses multiple indices, such as RPI and ME, and allows detailed insight into complex cognitive processes while learning computer programming, as recently recommended regarding measurement issues in cognitive load theory. The research was conducted at a vocational high school in Malang, Indonesia, involving 68 students enrolled in a computer programming course. The participants were divided into two groups: an experimental group (n=34) that received instruction using the EPBL model and a control group (n=34) taught using the PBL model. Both groups underwent the exact duration of instruction, with the critical difference being that the specific steps of the instructional process aligned with each model.

Research Variables

The study operationalized cognitive load through three distinct measures, each corresponding to a specific type of cognitive load defined by Cognitive Load Theory (CLT). Intrinsic Cognitive Load (ICL) was measured through students' ability to receive and process information (RPI). It was assessed using a set of nine essay questions designed to evaluate task complexity across three categories: information component, information integration, and information application. The complexity of these questions was carefully calibrated to trigger high intrinsic processing requirements, as suggested by [2]. Extraneous Cognitive Load (ECL) was quantified through students' mental effort (ME), utilizing a Subjective Rating Scale questionnaire. This instrument was designed to gauge students' mental effort in comprehending the provided material. The questionnaire employed a four-point Likert scale ranging from Very Low to Very High, allowing for a nuanced assessment of perceived cognitive demand. This approach aligns with recent recommendations in cognitive load research, emphasizing the importance of subjective measures in capturing the experiential aspects of cognitive load [4]. Germane Cognitive Load (GCL) was evaluated through students' learning outcomes (LO) and assessed via a multiple-choice test. The test questions were developed based on four key indicators: (1) understanding and distinguishing various forms of looping in PHP, (2) explaining the elements and structure of looping systems in PHP code syntax, (3) analyzing displayed looping systems with various forms, and (4) completing displayed code syntax with various looping forms. This comprehensive assessment approach thoroughly evaluates students' ability to apply and integrate the learned concepts, reflecting the germane cognitive load associated with schema construction and automation [21]. All research instruments were standardized to a 100-point scale to ensure comparability across measures. This standardization facilitates more straightforward comparisons between different aspects of cognitive load and allows for more robust statistical analyses. The data collection process was meticulously planned and executed to minimize potential confounds and ensure the reliability of the gathered information.

Preliminary Test: Normality Test, Homogeneity Test, and Linearity Test

Before conducting the primary analyses, we performed a series of classical assumption tests to verify the suitability of the data for parametric statistical analyses. These tests included the following: (1) normality test, which assesses whether the data follows a normal distribution, which is a fundamental assumption for many statistical tests; We employed the Shapiro-Wilk test due to its superior power for small to medium-sized samples; (2) homogeneity test: to evaluate the equality of variances across groups, ensuring that the variability in scores is similar between the EPBL and PBL groups. Levene's test was utilized for this purpose, and (3) the linearity test was used to examine the linear relationship between variables, which is particularly important for the correlation analyses planned. It was assessed through visual inspection of scatterplots and formal statistical tests. These preliminary tests are crucial for determining the appropriate statistical methods for subsequent analyses and ensuring the validity of the study's conclusions. The primary analytical approach involved both descriptive and inferential statistics. Descriptive statistics were computed to provide an overview of the cognitive load profiles for each instructional model, including measures of central tendency and dispersion for RPI, ME, and LO scores. To investigate the relationships between different types of cognitive load, we conducted correlation tests examining three key relationships: (1) ME and RPI, (2) ME and LO, and (3) RPI and LO. These correlational analyses were performed separately for each instructional model to allow for a comparison of the cognitive load dynamics between EPBL and PBL approaches.

Correlation Analysis

The strength and nature of these relationships were interpreted based on the correlation coefficients (r) obtained, with values between 0.00-0.19 considered very weak, 0.20-0.39 weak, 0.40-0.59 moderate, 0.60-0.79 strong, and 0.80-1.0 very strong. The direction of the relationship (positive or negative) was also noted to understand whether increases in one type of cognitive load were associated with increases or decreases in another. In addition to the quantitative analyses, we incorporated a qualitative component to our methodology to provide a more nuanced understanding of the cognitive processes underlying the observed cognitive load patterns. It involved carefully examining students' responses to the essay questions and their problem-solving approaches in the multiple-choice tests. This mixed-methods approach allows for a richer interpretation of the quantitative findings and provides valuable insights into the cognitive strategies employed by students under different instructional models.

RESULT AND DISCUSSION

The present study aimed to elucidate the differential effects of Example-Problem-Based Learning (EPBL) and Problem-Based Learning (PBL) instructional models on students' cognitive load during an iterative structure lesson in computer programming. The results provide compelling evidence for the efficacy of the EPBL model in managing cognitive load and enhancing learning outcomes in the context of vocational programming education. We conducted additional statistical analyses to provide a more comprehensive view of the data distribution.

Descriptive Statistics

The descriptive statistics provide additional context for interpreting the cognitive load profiles of students in each instructional model. The lower standard deviation in ME scores for the EPBL group (SD = 7.62) compared to the PBL group (SD = 8.34) suggests that the EPBL approach may lead to more consistent levels of mental effort across students. This consistency could be attributed to the structured nature of worked examples, which may help standardize the cognitive demands placed on learners (Renkl, 2017). In the EPBL model, students demonstrated a mean RPI score of 76.04, indicating high information reception and processing. It suggests that the work examples provided in the EPBL approach effectively facilitated students' understanding and internalization of the programming concepts. Concurrently, the mean ME score for the EPBL group was notably low at 46.54, implying that students experienced reduced extraneous cognitive load while engaging with the learning material. It aligns with the cognitive load theory principle that well-designed instructional interventions can minimize extraneous load, freeing cognitive resources for germane processing [2]. The mean LO score of 71.40 for the EPBL group further corroborates the effectiveness of this instructional approach in promoting learning outcomes. Table 1 presents the descriptive statistics for each cognitive load component across both instructional models.

Table 1. Descriptive Statistics of Cognitive Load Components

Model	**Measure**	**Mean**	**Median**	**SD**	**Min**	**Max**
EPBL	RPI	76.04	77.50	8.91	55	90
	ME	46.54	45.00	7.62	30	65
	LO	71.40	72.50	9.18	50	90
PBL	RPI	59.19	60.00	9.76	40	80
	ME	64.48	65.00	8.34	45	80
	LO	60.22	60.00	8.97	40	80

In contrast, the PBL model yielded markedly different results. Students in this group exhibited a lower mean RPI score of 59.19, suggesting they encountered greater difficulties in receiving and processing the programming information without the scaffolding provided by worked examples. The mean ME score for the PBL group was substantially higher at 64.48, indicating that students experienced increased extraneous cognitive load during problem-solving. The heightened mental effort may be attributed to the cognitive demands of simultaneously grappling with novel concepts and problem-solving strategies without the benefit of worked examples [15]. Consequently, the mean LO score for the PBL group was lower at 60.22, reflecting the potential impact of increased cognitive load on learning outcomes.

Analysis of the Results

To provide a more nuanced understanding of the relationship between different aspects of cognitive load, we conducted correlation analyses for both instructional models. Table 2(a) presents the results of these analyses, offering insights into the interplay between ME, RPI, and LO. In the EPBL model, the correlation between ME and RPI yielded an r-value of 0.142, indicating a weak negative relationship. It suggests that as students' mental effort decreased, their ability to receive and process information slightly improved. This finding aligns with the cognitive load theory principle that reducing extraneous load can facilitate more efficient information processing [15]. The positive correlation ($r = 0.339$) between ME and LO in the EPBL model suggests that students who invested more mental effort tended to achieve better learning outcomes, possibly reflecting the germane cognitive load associated with meaningful learning. The positive correlation ($r = 0.258$) between RPI and LO further supports the effectiveness of the EPBL approach, indicating that students who were better able to receive and process information also demonstrated improved learning outcomes.

Table 2. Correlation Analysis of Cognitive Load Components and Independent Samples t-test Results

(a) Correlation Analysis

Model	**Cognitive Load Relationship**	**r value**	**Relationship Type**
EPBL	ME - RPI	0.142	Negative
	ME - LO	0.339	Positive
	RPI - LO	0.258	Positive
PBL	ME - RPI	0.013	Positive
	ME - LO	0.290	Positive
	RPI - LO	0.309	Negative

(b) t-test Results

Measure	**t-value**	**df**	**p-value**	**Cohen's d**
RPI	7.621	66	<0.001	1.847
ME	-9.342	66	<0.001	-2.263
LO	5.128	66	<0.001	1.242

The PBL model, however, exhibited different correlation patterns. The relationship between ME and RPI was negligibly positive ($r = 0.013$), suggesting no significant association between mental effort and information processing in this instructional approach. The positive correlation ($r = 0.290$) between ME and LO in the PBL model was similar to that observed in the EPBL model, indicating that increased mental effort was associated with better learning outcomes. However, the negative correlation ($r = -0.309$) between RPI and LO in the PBL model is noteworthy, as it suggests that students who reported higher levels of information reception and processing tended to achieve lower learning outcomes. This counterintuitive finding warrants further investigation and may reflect the complex interplay between cognitive load components in problem-based learning environments. To further elucidate the differences between the two instructional models, we conducted independent sample t-tests for each cognitive load component. The results of these analyses are presented in Table 2(b).

The t-test results reveal statistically significant differences between the EPBL and PBL groups across all cognitive load components ($p < 0.001$ for all comparisons). The large effect sizes, as indicated by Cohen's d values, underscore the practical significance of these differences. The particularly large effect size for ME ($d = -2.263$) highlights the substantial impact of the EPBL approach on reducing extraneous cognitive load compared to the PBL approach. These findings align with previous research on the efficacy of worked examples in managing cognitive load. For instance, Sweller et al. (2019) demonstrated that worked examples can significantly reduce extraneous cognitive load by providing learners with explicit problem-solving schemas [2]. Our results extend this understanding to vocational programming education, suggesting that integrating worked examples in the EPBL approach may benefit novice programmers grappling with complex iterative structures.

The lower ME scores observed in the EPBL group are consistent with the findings of Chen et al. (2018), who reported that worked examples can reduce the cognitive demands associated with problem-solving, particularly for learners with limited prior knowledge. In programming education, where students must simultaneously manage syntactical rules, logical structures, and problem-solving strategies, the scaffolding provided by worked examples appears to play a crucial role in managing cognitive load. The higher RPI scores in the EPBL group suggest that this approach may facilitate more efficient information processing. It aligns with the cognitive load theory principle that reducing extraneous load can free up cognitive resources for germane processing [21]. By providing students with clear examples of problem-solving strategies, the EPBL approach may enable learners to focus more on understanding and internalizing the underlying programming concepts rather than expending cognitive resources on navigating unfamiliar problem spaces. The superior LO scores observed in the EPBL group further support the effectiveness of this instructional approach in promoting learning outcomes. This finding is consistent with the work of [9], who demonstrated that example-based learning can enhance knowledge transfer and skill acquisition in complex domains. In programming education, where applying abstract concepts to concrete problems is crucial, the EPBL approach offers significant advantages over traditional problem-based learning [26], [27], [28].

However, it is essential to note that the relationship between cognitive load components is complex and may vary depending on learner characteristics and task demands. The positive correlation between ME and LO observed in both instructional models suggests that some level of mental effort is necessary for effective learning. It aligns with the concept of desirable difficulties in learning, which posits that some cognitive challenges are beneficial for long-term retention and transfer [29]. The negative correlation between RPI and LO in the PBL group is an intriguing finding that warrants further investigation. One possible explanation is that students in the PBL group who reported higher levels of information reception and processing may have been overwhelmed by the complexity of the programming tasks, leading to suboptimal learning outcomes. This interpretation is consistent with the expertise reversal effect described by Kalyuga et al. [30], which suggests that instructional techniques that are effective for novices may become less effective or even detrimental as learners gain expertise.

The Efficacy of the EPBL

In conclusion, the results of this study provide strong evidence for the efficacy of the EPBL instructional model in managing cognitive load and enhancing learning outcomes in vocational programming education. Integrating worked examples offers significant advantages in reducing extraneous cognitive load, facilitating information processing, and promoting knowledge acquisition. These findings have important implications for instructional design in programming education and highlight the potential of cognitive load theory as a framework for optimizing learning experiences in complex domains. The present study's findings offer compelling evidence for the efficacy of the Example-Problem-Based Learning (EPBL) instructional model in managing cognitive load and enhancing learning outcomes within the context of vocational programming education. The results demonstrate that EPBL significantly outperforms the Problem-Based Learning (PBL) approach across multiple dimensions of cognitive load and learning performance. These findings align with and extend the existing body of research on cognitive load theory and instructional design in complex learning domains. The observed lower Mental Effort (ME) scores in the EPBL group, coupled with higher Receiving and Processing Information (RPI) and Learning Outcomes (LO) scores, provide strong support for the theoretical underpinnings of cognitive load theory. This pattern suggests that the EPBL approach effectively reduces extraneous cognitive load while simultaneously enhancing germane load, thereby optimizing the allocation of cognitive resources during the learning process. These results are consistent with the findings of Sweller et al. (2019) [2], who demonstrated that well-designed instructional interventions can minimize extraneous load, freeing up cognitive resources for schema acquisition and automation. The superior performance of the EPBL model can be attributed to its strategic integration of worked examples, which are particularly effective in reducing cognitive load for novice learners [15]. By providing students with explicit problem-solving schemas, worked examples alleviate the need for extensive means-end analysis, a process that typically imposes a heavy cognitive burden on learners in complex domains such as computer programming [9]. This reduced extraneous load allows students to focus more cognitive resources on understanding the underlying principles and concepts, facilitating more efficient and effective learning.

The positive correlation between ME and LO in both instructional models underscores the complex relationship between cognitive load and learning outcomes. This finding aligns with the concept of desirable difficulties in learning, which posits that some level of cognitive challenge is beneficial for long-term retention and transfer [31]. However, the stronger positive correlation in the EPBL group suggests that this approach may be more effective in calibrating the level of challenge to students' cognitive capacities, striking a balance between cognitive load and learning efficiency. The negative correlation between RPI and LO in the PBL group is an intriguing finding that warrants further investigation. This counterintuitive relationship may indicate the expertise reversal effect, as described by Kalyuga et al. [30]. Students in the PBL group who said they received and processed more information may have been overwhelmed due to the lack of work examples to guide them. This interpretation is consistent with recent research by Klepsch et al. (2017) [4], who emphasized the importance of tailoring instructional approaches to learners' prior knowledge and expertise levels.

Limitations and Future Research

While the current study provides strong evidence for the efficacy of the EPBL approach in managing cognitive load and enhancing learning outcomes in vocational programming education, it is important to acknowledge several limitations that should be addressed in future research. Firstly, while adequate for detecting significant effects, the study's sample size was relatively modest. Future studies should aim to replicate these findings with more extensive and diverse samples to enhance the generalizability of the results across different educational contexts and student populations. Secondly, the current study focused on a specific programming concept (iterative structures) within a particular programming language. Future research should investigate its effectiveness across a more comprehensive range of programming concepts and languages to establish the broader applicability of the EPBL approach. This expansion would provide valuable insights into the robustness of the EPBL model across different levels of task complexity and domain-specific challenges. Thirdly, while the study employed multiple measures of cognitive load, including subjective ratings of mental effort and objective performance measures, future research could benefit from including more direct measures of cognitive load. For instance, incorporating physiological measures such as eye-tracking data or electroencephalography (EEG) could provide more nuanced insights into the cognitive processes underlying learning in EPBL and PBL environments [32].

Furthermore, the current study was conducted over a relatively short time frame, focusing on immediate learning outcomes. Longitudinal studies examining the long-term retention and transfer of programming skills acquired through EPBL would be invaluable in assessing the durability of learning gains and the development of expertise over time. Such studies could also investigate the potential of EPBL to foster self-regulated learning skills, which are crucial for ongoing professional development in rapidly evolving fields like computer programming [7]. Another avenue for future research lies in exploring the potential of adaptive learning systems that dynamically adjust the level of scaffolding provided based on individual students' cognitive load and performance. Recent advances in educational technology and artificial intelligence offer promising opportunities for developing intelligent tutoring systems that seamlessly transition between worked examples, completion problems, and problem-solving tasks based on real-time assessments of student progress [33]. Integrating collaborative learning elements within the EPBL framework also merits further investigation. While the current study focused on individual learning, future research could examine how peer collaboration and group problem-solving activities can be effectively incorporated into the EPBL model without inducing excessive cognitive load. This line of inquiry could draw upon recent work on collaborative cognitive load theory to optimize the balance between individual and group learning activities [1]. Additionally, future studies should investigate the role of motivational factors in mediating the relationship between cognitive load and learning outcomes in EPBL environments. Integrating self-determination theory [34] with cognitive load theory could provide valuable insights into how instructional design can simultaneously optimize cognitive load and enhance student motivation and engagement [35], [36].

From a practical standpoint, the findings of this study have significant implications for instructional design in vocational programming education and potentially in other complex learning domains. The superior performance of the EPBL model suggests that educators and curriculum designers should consider incorporating worked examples and completion problems as crucial components of their instructional strategies, particularly in the early stages of skill acquisition. However, it is essential to note that implementing EPBL may require significant changes to existing curricula and teaching practices. Future research should focus on developing practical guidelines and professional development programs to support educators in effectively implementing EPBL approaches in their classrooms. It could include the development of repositories of high-quality work examples and completion problems tailored to specific programming languages and concepts [37], [38].

Moreover, the potential of EPBL to address issues of equity and inclusion in computer science education should be explored. By providing structured support and scaffolding, EPBL may help to level the playing field for students from diverse backgrounds and with varying levels of prior programming experience. Future studies could investigate the efficacy of EPBL in reducing achievement gaps and promoting diversity in computer science education. In conclusion, this study provides robust evidence for the effectiveness of the EPBL instructional model in managing cognitive load and enhancing learning outcomes in vocational programming education. The findings underscore the importance of cognitive load theory in designing instructional materials and approaches for complex learning domains. While the results are promising, they also highlight the need for continued research to refine our understanding of the intricate relationships between instructional design, cognitive load, and learning outcomes in programming education and beyond. As computer science education evolves in response to rapid technological advancements and changing workforce demands, cognitive load theory and the EPBL approach offer a valuable framework for developing effective and efficient learning experiences. By building on the findings of this study and addressing the identified limitations and future research directions, educators and researchers can work towards creating more inclusive, engaging, and impactful programming education that equips students with the skills and knowledge needed to thrive in the digital age.

CONCLUSION

Note that throughout the presentation of the result, a great attempt has been made to ensure that an EPBL and PBL instructional sequence differentially affects students' cognitive load while learning vocational computer programming. Specifically, the format better coped with the increased cognitive load of students, seen through lower ME and higher RPI scores and better LO presentation compared to the PBL group. More critically, the results of the present study are consistent with CLT theory as EPBL significantly decreased extraneous cognitive load and increased germane load. These are supported by the negative correlations of ME and RPI with EPBL and the positive ones between ME and LO and RPI and LO. These associations suggest that EPBL effectively balances cognitive demands, thus facilitating students to use more resources towards schema acquisition and automation. Such a finding is noteworthy for its implications in designing and developing instructional strategies in vocational programming education. EPBL, as an alternative format to combine worked examples with problems to be solved, maximized positive load management for enhancement in overall learning. For example, problems and completion exercises should form part of instructional strategies early in the instructor's skill development cycle. Long-term retention and transferability of programming skills in EPBL further need to be researched in terms of how all these elements work with aspects such as motivational influences, self-efficacy, and prior knowledge that interact with cognitive load. An even more exciting issue, probably with a closer practical application of the CLT principles in teaching programming, would be the development of adaptive learning systems that could autonomously adapt to an individual learner's cognitive load and success. That makes this study very pivotal research for confirming that design methods used for instruction in complex learning environments require cognitive load theory. Evidence-based approaches, such as EPBL, might support effectiveness and efficiency in programming education in a vocational context, thus better-preparing students for the challenges of the rapidly changing technology industry.

REFERENCES

[1] P. A. Kirschner, J. Sweller, F. Kirschner, and J. R. Zambrano, "From Cognitive Load Theory to Collaborative Cognitive Load Theory," *Int J Comput Support Collab Learn*, vol. 13, no. 2, pp. 213–233, 2018, doi: 10.1007/s11412-018-9277-y.

[2] J. Sweller, J. J. G. van Merriënboer, and F. Paas, "Cognitive Architecture and Instructional Design: 20 Years Later," *Educ Psychol Rev*, vol. 31, no. 2, pp. 261–292, 2019, doi: 10.1007/s10648-019-09465-5.

[3] R. Duran, A. Zavgorodniaia, and J. Sorva, "Cognitive Load Theory in Computing Education Research: A Review," *ACM Transactions on Computing Education*, vol. 22, no. 4, 2022, doi: 10.1145/3483843.

[4] M. Klepsch, F. Schmitz, and T. Seufert, "Development and validation of two instruments measuring intrinsic, extraneous, and germane cognitive load," *Front Psychol*, vol. 8, no. NOV, 2017, doi: 10.3389/fpsyg.2017.01997.

[5] A. Abbad-Andaloussi, "On the relationship between source-code metrics and cognitive load: A systematic tertiary review," *Journal of Systems and Software*, vol. 198, 2023, doi: 10.1016/j.jss.2023.111619.

[6] S. Sepp, S. J. Howard, S. Tindall-Ford, S. Agostinho, and F. Paas, "Cognitive Load Theory and Human Movement: Towards an Integrated Model of Working Memory," *Educ Psychol Rev*, vol. 31, no. 2, pp. 293–317, 2019, doi: 10.1007/s10648-019-09461-9.

[7] Y. Shin and D. Song, "The Effects of Self-Regulated Learning Support on Learners' Task Performance and Cognitive Load in Computer Programing," *Journal of Educational Computing Research*, vol. 60, no. 6, pp. 1490–1513, 2022, doi: 10.1177/07356331211052632.

[8] S. C. Chang and C. Wongwatkit, "Effects of a peer assessment-based scrum project learning system on computer programming's learning motivation, collaboration, communication, critical thinking, and cognitive load," *Educ Inf Technol (Dordr)*, vol. 29, no. 6, pp. 7105–7128, 2024, doi: 10.1007/s10639-023-12084-x.

[9] A. Renkl, "Learning from worked-examples in mathematics: students relate procedures to principles," *ZDM - Mathematics Education*, vol. 49, no. 4, pp. 571–584, 2017, doi: 10.1007/s11858-017-0859-3.

[10] E. Lovellette, D. J. Bouvier, and J. Matta, "Contextualization, Authenticity, and the Problem Description Effect," *ACM Transactions on Computing Education*, vol. 24, no. 2, 2024, doi: 10.1145/3643864.

[11] T. Chamidy, I. N. S. Degeng, and S. Ulfa, "The effect of problem-based learning and tacit knowledge on problem-solving skills of students in computer network practice course," *Journal for the Education of Gifted Young Scientists*, vol. 8, no. 2, pp. 691–700, 2020, doi: 10.17478/JEGYS.650400.

[12] B. Wallace, D. Knudson, and N. Gheidi, "Incorporating problem-based learning with direct instruction improves student learning in undergraduate biomechanics," *J Hosp Leis Sport Tour Educ*, vol. 27, 2020, doi: 10.1016/j.jhlste.2020.100258.

[13] T. van Gog, N. Rummel, and A. Renkl, "Learning how to solve problems by studying examples.," in *The Cambridge handbook of cognition and education.*, New York, NY, US: Cambridge University Press, 2019, pp. 183–208. doi: 10.1017/9781108235631.

[14] A. D. Herlambang, I. Ismaiwati, and S. H. Wijoyo, "The Effect of Problem-Based Instruction Learning Methods on Improving Cognitive and Psychomotor Learning Outcomes in Computer Programming Subjects on Branching Control Structure Material," *JICTE (Journal of Information and Computer Technology Education)*, vol. 4, no. 2, pp. 1–6–1–6, 2020, doi: 10.21070/jicte.v4i2.920.

[15] O. Chen, J. C. Castro-Alonso, F. Paas, and J. Sweller, "Extending Cognitive Load Theory to Incorporate Working Memory Resource Depletion: Evidence from the Spacing Effect," *Educ Psychol Rev*, vol. 30, no. 2, pp. 483–501, 2018, doi: 10.1007/s10648-017-9426-2.

[16] X. Gao, L. Wang, J. Deng, C. Wan, and D. Mu, "The effect of the problem based learning teaching model combined with mind mapping on nursing teaching: A meta-analysis," *Nurse Educ Today*, vol. 111, 2022, doi: 10.1016/j.nedt.2022.105306.

[17] R. Ramadhani, R. Umam, A. Abdurrahman, and M. Syazali, "The effect of flipped-problem based learning model integrated with LMS-google classroom for senior high school students," *Journal for the Education of Gifted Young Scientists*, vol. 7, no. 2, pp. 137–158, 2019, doi: 10.17478/jegys.548350.

[18] A. Majid, A. D. Herlambang, and F. Amalia, "Pengaruh Metode Pembelajaran Problem Based Learning yang Diperkaya dengan ARCS Motivational Model terhadap Kualitas Manajemen Kelas dan Motivasi …," *… Informasi dan Ilmu Komputer*, vol. 9, no. 1, pp. 129–136–129–136, 2022, [Online]. Available: http://jtiik.ub.ac.id/index.php/jtiik/article/view/5500

[19] M. Robherta, A. Dwi Herlambang, and S. Hadi Wijoyo, "The Differences of Student's Learning Outcomes and Instructional Interactions between Project Based Learning and Problem Based Learning Methods by Using Web Based Learning Technique in the Course of Videography at SMKN 1 Purwosari," *Journal of Information Technology and Computer Science*, vol. 6, no. 3, pp. 225–235, 2021, doi: 10.25126/jitecs.202163305.

[20] V. L. Barth, V. Piwowar, I. R. Kumschick, D. Ophardt, and F. Thiel, "The impact of direct instruction in a problem-based learning setting. Effects of a video-based training program to foster preservice teachers' professional vision of critical incidents in the classroom," *Int J Educ Res*, vol. 95, pp. 1–12, 2019, doi: 10.1016/j.ijer.2019.03.002.

[21] F. Paas and J. J. G. van Merriënboer, "Cognitive-Load Theory: Methods to Manage Working Memory Load in the Learning of Complex Tasks," *Curr Dir Psychol Sci*, vol. 29, no. 4, pp. 394–398, Aug. 2020, doi: 10.1177/0963721420922183.

[22] J. Leppink, "Cognitive load theory: Practical implications and an important challenge," *J Taibah Univ Med Sci*, vol. 12, no. 5, pp. 385–391, 2017, doi: 10.1016/j.jtumed.2017.05.003.

[23] P. Toukiloglou and S. Xinogalos, "Ingame Worked Examples Support as an Alternative to Textual Instructions in Serious Games About Programming," *Journal of Educational Computing Research*, vol. 60, no. 7, pp. 1615–1636, 2022, doi: 10.1177/07356331211073655.

[24] Y. Shin, J. Jung, J. Zumbach, and E. Yi, "The Effects of Worked-Out Example and Metacognitive Scaffolding on Problem-Solving Programming," *Journal of Educational Computing Research*, vol. 61, no. 6, pp. 1312–1331, 2023, doi: 10.1177/07356331231174454.

[25] S. Ghanbari, F. Haghani, M. Barekatain, and A. Jamali, "A systematized review of cognitive load theory in health sciences education and a perspective from cognitive neuroscience.," *J Educ Health Promot*, vol. 9, p. 176, 2020, doi: 10.4103/jehp.jehp_643_19.

[26] D. Pérez-Marín, R. Hijón-Neira, A. Romero, and S. Cruz, "Is the use of makey makey helpful to teach programming concepts to primary education students?," *International Journal of Online Pedagogy and Course Design*, vol. 9, no. 2, pp. 63–77, 2019, doi: 10.4018/IJOPCD.2019040105.

[27] Ü. Çakiroğlu and Ş. Bilgi, “Exploring intrinsic cognitive load in the programming process: a two dimensional approach based on element interactivity,” *Interactive Learning Environments*, 2022, doi: 10.1080/10494820.2022.2137527.

[28] I. Lavy, “Leveraging the Pied Piper Effect – The Case of Teaching Programming to Sixth-grade Students Via Music,” *Informatics in Education*, vol. 22, no. 1, pp. 45–69, 2023, doi: 10.15388/infedu.2023.06.

[29] J. M. Schorn and B. J. Knowlton, “Interleaved practice benefits implicit sequence learning and transfer.,” *Mem Cognit*, vol. 49, no. 7, pp. 1436–1452, Oct. 2021, doi: 10.3758/s13421-021-01168-z.

[30] A. Armougum, A. Gaston-Bellegarde, C. J.-L. Marle, and P. Piolino, “Expertise reversal effect: Cost of generating new schemas,” *Comput Human Behav*, vol. 111, p. 106406, Oct. 2020, doi: 10.1016/j.chb.2020.106406.

[31] A. B. H. de Bruin, F. Biwer, L. Hui, E. Onan, L. David, and W. Wiradhany, “Worth the Effort: the Start and Stick to Desirable Difficulties (S2D2) Framework,” *Educ Psychol Rev*, vol. 35, no. 2, p. 41, Jun. 2023, doi: 10.1007/s10648-023-09766-w.

[32] K. Ouwehand, A. van der Kroef, J. Wong, and F. Paas, “Measuring Cognitive Load: Are There More Valid Alternatives to Likert Rating Scales?,” *Front Educ (Lausanne)*, vol. 6, Sep. 2021, doi: 10.3389/feduc.2021.702616.

[33] F. Kamalov, D. Santandreu Calonge, and I. Gurrib, “New Era of Artificial Intelligence in Education: Towards a Sustainable Multifaceted Revolution,” *Sustainability*, vol. 15, no. 16, p. 12451, Aug. 2023, doi: 10.3390/su151612451.

[34] R. M. Ryan and E. L. Deci, “Intrinsic and extrinsic motivation from a self-determination theory perspective: Definitions, theory, practices, and future directions,” *Contemp Educ Psychol*, vol. 61, p. 101860, 2020, doi: 10.1016/j.cedpsych.2020.101860.

[35] A. Al-Sakkaf, M. Omar, and M. Ahmad, “Social worked-examples technique to enhance student engagement in program visualization,” *Baghdad Science Journal*, vol. 16, no. 2, pp. 453–461, 2019, doi: 10.21123/bsj.2019.16.2(SI)0453.

[36] X. Zhang, Y. Chen, D. Li, L. Hu, G. J. Hwang, and Y. F. Tu, “Engaging Young Students in Effective Robotics Education: An Embodied Learning-Based Computer Programming Approach,” *Journal of Educational Computing Research*, vol. 62, no. 2, pp. 532–558, 2024, doi: 10.1177/07356331231213548.

[37] J. Castelhano *et al.*, “Software Bug Detection Causes a Shift From Bottom-Up to Top-Down Effective Connectivity Involving the Insula Within the Error-Monitoring Network,” *Front Hum Neurosci*, vol. 16, 2022, doi: 10.3389/fnhum.2022.788272.

[38] A. Qayum, S. U. R. Khan, Inayat-Ur-Rehman, and A. Akhunzada, “FineCodeAnalyzer: Multi-Perspective Source Code Analysis Support for Software Developer Through Fine-Granular Level Interactive Code Visualization,” *IEEE Access*, vol. 10, pp. 20496–20513, 2022, doi: 10.1109/ACCESS.2022.3151395.