Exploring the Relationship Between Usability and Cognitive Load in Data Science Education

Andrew A. Tawfik, Linda Payne, Andrew M. Olney, & Heather Ketter

DOI:10.59668/515.13078

Scaffolding | Computational Thinking | Learning Experience Design | Cognitive Load Theory | Data Science Education

THE JOURNAL OF APPLIED INSTRUCTIONAL DESIGN

This study explores how an aspect of learning experience design (usability) correlated with the learning process as individuals engaged in block coding. Although there were no differences when scaffolded with blocks or no blocks coding condition, the study found weak to moderate correlations with usability and factors of cognitive load (intrinsic, germane, extraneous). Whereas learning experience design literature is often approached from an evaluation perspective, the empirical data suggests learning experience design may be correlated with in situ elements of learning.

Introduction

The United States federal strategic plan notes that STEM education is a critical component of the country's immediate and future goals (Bottia et al., 2021; Fernandez et al., 2022). Not only does STEM support global competitiveness, but STEM education is also important for equity among all learners. One response has focused on a more holistic view of STEM, which integrates skills such as computational thinking and data science as learners resolve contextualized cases. For example, a student might employ data science to measure expected rainfall within a region. Another case might ask learners to explore clusters of mammals to investigate the importance of biodiversity. In doing so, data science requires learners to engage aspects of statistics, programming, and machine learning as part of their STEM learning experience.

As this integrated view of STEM has grown, there has been an increased emphasis on learning environments that make data science education more accessible to a wider array of learners. It follows that interactions within these learning environments and learning experience design (LXD) are therefore a key component of the design of STEM learning tools designed for these skillets. However, few studies have explicated the extent to which LXD aligns with key aspects of the learning process, such as cognitive load. Given this gap, the article first begins by exploring the growth of data science education. Second, we describe the importance of interactions within data science education and the importance of

LXD. Specifically, the literature review explores how the field of learning design has evolved from exploring usability as an aspect of evaluation towards a more comprehensive and holistic view of LXD, along with proposed definitions as the field seeks to understand the LXD phenomenon. We then present a study that looks at LXD and its relationship with cognitive load factors (intrinsic, germane, and extraneous) and conceptual knowledge (test scores). Finally, we conclude with a discussion and implications for data science learning, interactions with complex learning tools, and LXD.

Literature Review

Data science is an interdisciplinary field that combines aspects of statistics, machine learning, and computer science in order to analyze data. While data science as a field arguably dates back at least 20 years (e.g., with the journal *Data Mining and Knowledge Discovery*) and research on computer science education and statistics education goes back decades more, the confluence of skills and knowledge that come together in the modern definition of data science has received little attention. This lack of understanding of how data science is learned is perhaps due to the interdisciplinary nature of data science, the current science of learning research emphasis on K-12, and the recent emergence of the label "data science."

Learning data science appears to be particularly challenging in large part due to the deep prerequisites in statistics, programming, and machine learning. As such, 89% of U.S. degree programs in data science are graduate programs (Swanstrom, 2020). Data scientists are not using degree programs for the bulk of their training as indicated by a recent survey (Kaggle, 2021). Rather the majority of data scientists are learning on the job by participating in data science competitions, taking online courses, and via informal learning resources (Kaggle, 2021). The current state of data science education and learning presents both opportunities and challenges for the science of learning data science.

Prior work in computer science education suggests several instructional supports that may be applicable, but they have never been applied to learning data science. One way to understand the need for learner support is through the lens of cognitive load theory, which suggests that individual's have limited cognitive capacity as they process information via their working memory (Sweller, 2020). The theory further posits that the load consists of three factors - intrinsic cognitive load (ICL), germane cognitive load (GCL), and extraneous cognitive load (ECL). Intrinsic cognitive load is defined as the natural complexity of the information and is often considered fixed based and largely dependent on the inherent element interactivity (Klepsch et al., 2017). Low element interactivity allows individuals to learn a concept with minimal reference to other elements, which results in low working memory. Alternatively, high element interactivity results in a large strain on working memory because it requires elements to be learned in conjunction as they impact one another. Whereas intrinsic cognitive load is often dependent on the characteristics of the material, germane cognitive load is focused on schema acquisition as it directs working memory resources that are triggered by the design of learning resources. Lastly, extraneous cognitive load is described at the strain of working memory that distracts from the learning process (e.g., flashing words, distracting words) (Skulmowski & Xu, 2022). Therefore, a "decrease in extraneous cognitive load results in an increase in germane cognitive load as working memory resources are switched from elements associated with extraneous to elements associated with intrinsic cognitive load" (Sweller, 2010, p. 126).

There have been considerable efforts to support cognitive load as learners engage in computational thinking and data science education. For example, worked examples are a well-known pedagogical approach that benefits problem-solving learning and cognitive load in fields like mathematics, programming, and physics (Atkinson et al., 2000; Pashler et al., 2007). Minimally, a worked example is a problem solution that shows each worked step (Clark et al., 2011). Learners studying worked examples must induce, or self-explain, the missing reasons for each step (Chi et al., 1989, 1994; Nokes et al., 2011). In other words, learners must self-explain why each step is desirable (the latent goal structure of the problem) and why each step is permissible (the epistemic justification of the step). Various approaches for transitioning students to independent problem-solving have been investigated within education, including interleaving worked examples with problem-solving (Sweller & Cooper, 1985) and converting worked examples to problem-solving at the step level (Atkinson et al., 2003).

Another pedagogical approach, graphical programming languages (Cunniff et al., 1986), has seen recent wide adoption for teaching introductory programming in the form of blocks languages (Bau et al., 2017; Resnick et al., 2009). Blocks languages compose code elements via irregularly shaped graphical widgets. Their design typically makes syntactic mistakes impossible because the widgets cannot fit together in non-syntactic ways. Because blocks are visually browsable on an interface palette, students need only recognize them rather than the more difficult task of recalling code. Research suggests blocks languages appear to have multiple positive effects on learning, including both cognitive and motivational effects, in introductory undergraduate courses. For example, Armoni et al. (2015) found that an introductory course using a blocks language resulted in increased motivation and future enrollment in advanced computer science classes. Comparable effects have also been found for at-risk students using blocks languages, such that the blocks-based approach led to equivalent or greater retention and motivation outcomes for at-risk students than non-at-risk students (Moskal et al., 2004). While Armoni et al. (2015) found no difference in grades between students who previously took a blocks-based course and those who did not, another study found that a blended blocks/Java instructional method increased test scores by approximately a letter grade (Dann et al., 2012). These studies suggest that blocks may be an effective way to foster computational thinking and data science skills within STEM education.

Usability, UX, and Learning Experience Design

Given that data science is reliant on learning technologies to perform tasks, it follows that interaction is a key element of learning experience design. This multifaceted phenomenon includes aspects including usability and other interactions individuals engage in as they interact with the learning environment. In disciplines outside of education, a component of user-experience literature includes usability, which describes the efficiency by which users are able to use the features of the system. Although approaches may vary, studies have begun to empirically validate how interaction is an important aspect of learning technologies. In early stage research, Lohr et al. (2003) found that usability testing identified areas of refinement and specific features that were needed. Similarly, Lim et al. (2012) detailed how learners utilized an interactive textbook and credited usability testing for suggesting ways the interface could support self-directed learning and cognitive load management. Findings suggest key interface design and interaction considerations in terms of learnability, efficiency, effectiveness, and satisfaction. Rather than a distinct evaluation phase at the end of the design cycle, Lim and colleagues (2012) especially highlighted the highly iterative nature of usability testing and referenced its importance throughout the development process. In line with these findings, Schmidt and Glaser (2021) detailed how usability testing provided data as to the effectiveness, efficiency and appeal of a learning technology. Beyond the traditional views of evaluation, they underscored the need for usability testing to be inclusive of diverse demographics given their unique needs. Whereas design has often been approached from a learning theory perspective, the literature highlights how evaluation of user interaction data is a key component of the design process for diverse learning groups.

The literature on LXD within learning technologies has traditionally focused on data collection techniques (e.g., eye tracking) used by learning designers. However, the data denoting the importance of LX has been detailed as subconstructs within other theoretical frameworks. For example, the unified theory of acceptance and use of technology (UTAUT) describes a construct of effort expectancy (Marchewka et al., 2007), which takes into account how key elements of usability impact technology adoption. Other literature illustrates how extraneous cognitive load in the form of poor navigation and usability can strain working memory, which ultimately deters the learning process (Novak et al., 2018). A more comprehensive view of LXD is needed because it not only focuses on traditional cognitive or affective learning outcomes, but it considers how to design for the overall learning experience and the technology interactions that drive the goal-directed behavior.

As interest in LXD has emerged, theorists (e.g., Clark, 2022) have recently provided various definitions to provide clarity around the phenomenon of LXD. For example, Chang and Kuwata (2020) describe it as a "practice of designing learning as a human-centered experience leading to a desired goal," which merges design practice with the learner role. Yet others provide a broader view, especially considering the broader socio-technical context in which LXD takes place (Gray, 2020; Jahnke et al., 2020). More recently, data has also emerged to provide clarity on the elements inherent within LXD. Tawfik et al. (2022) employed a grounded theory to empirically describe LXD in terms of two complementary facets: *interaction with the learning space* (engagement with the modality of content, dynamic interaction, the perceived value of technology features to support learning, scaffolding) and *interaction with the learning environment* (customization, expectation of content placement, functionality of component parts, interface terms aligned with existing mental models, navigation). In doing so, their framework extends beyond theory as the data details unique interactions that are collectively part of the learning experience. When compared with the case study approaches, Schmidt and Huang (2022) surveyed the literature and described LXD as "human-centered, goal-oriented, theoretically-grounded, and interdisciplinary" (p. 149). Collectively, the aforementioned articles suggest emerging theory and empirical support as the field looks to apply LXD in the design of learning technologies.

Research Questions

Novak et al. (2018) argue that "Despite a growing body of research in the area of digital learning and information processing, the literature on how people process and interact with information on electronic devices and computers is still very scarce" (p. 151). To better understand its impact, more empirical data is needed to understand how interactions play a role as learners engage with technology. A number of qualitative studies have provided insights into the UX or user perceptions (Carey & Stefaniak, 2018), but they may be limited to describing the extent of the relationship between these aspects of design. While quantitative studies have been published, these are often in the form of descriptive statistics or from a technology acceptance perspective. Based on this gap, we proffer the following research questions:

1. To what degree are *learning outcomes* different when supported with a blocks or no blocks approach to data science?
    a. To what degree is *conceptual knowledge* different when supported with a blocks or no blocks approach to data science?
    b. To what degree are the three factors of *cognitive load* (i.e., intrinsic, germane, extraneous) different when supported with a blocks or no blocks approach to data science?
2. To what degree is *learning experience design* correlated with factors of cognitive load (intrinsic, germane, extraneous)?
3. To what degree is *learning experience design* correlated with aspects of *conceptual knowledge* of computational thinking?

Methodology

The study was part of a larger set of design-based research that explores specific learning supports that support data science education.

Participants

All participants were recruited from the psychology subject pool of a large urban university in the southeastern United States (*n*=59). Participants were compensated with course credit or extra credit, depending on the psychology courses in which they were enrolled.

Materials

Learning Environment

Research on worked examples and blocks programming in prerequisite areas of data science suggest that these instructional supports could be integrated into a single design for learning data science. In our recent work, we have reified these supports inside JupyterLab computational notebooks, which are the most popular platform for professional data scientists (Kaggle, 2021). JupyterLab computational notebooks combine narrative, executable code, and rich media like interactive plots, allowing analyses to be shared and reproduced by other data scientists. As such, they are naturally occurring worked examples. We have added blocks-based programming to JupyterLab using its extension framework, such that naive users can connect blocks to solve a data science problem, click a button to convert those blocks to Python code, and execute that code in JupyterLab as usual (Olney & Fleming, 2021). We have created over 100 hours of data science training materials using this approach and have run an 8-week data science internship open to all majors (Payne et al., 2021) for the past 3 years.

Two training videos were used in different experimental conditions: a video on how to use the computational notebook interface (JupyterLab) using code only and a video on how to use the Jupyter interface using blocks. The videos were carefully constructed to cover the same content except for the blocks/code component. Four Jupyter notebooks were used with variations based on experimental condition: a didactic worked example notebook that demonstrated solution steps using video (WE), a problem-solving notebook requiring near transfer to the worked example without demonstrated steps (NEAR1), a second near transfer problem-solving notebook (NEAR2), a far transfer problem-solving notebook (FAR), and an even further transfer problem-solving notebook (FAR+). An example of this level of problem solving transfer for filtering rows in a data frame based on a value in a column would be, WE: x < 7, NEAR: x < 10, FAR: x > 5, FAR+: x = "Smoking" (i.e., changing the value alone is near transfer, reversing the inequality is far transfer, and changing the operator and data type is further transfer).

Instruments

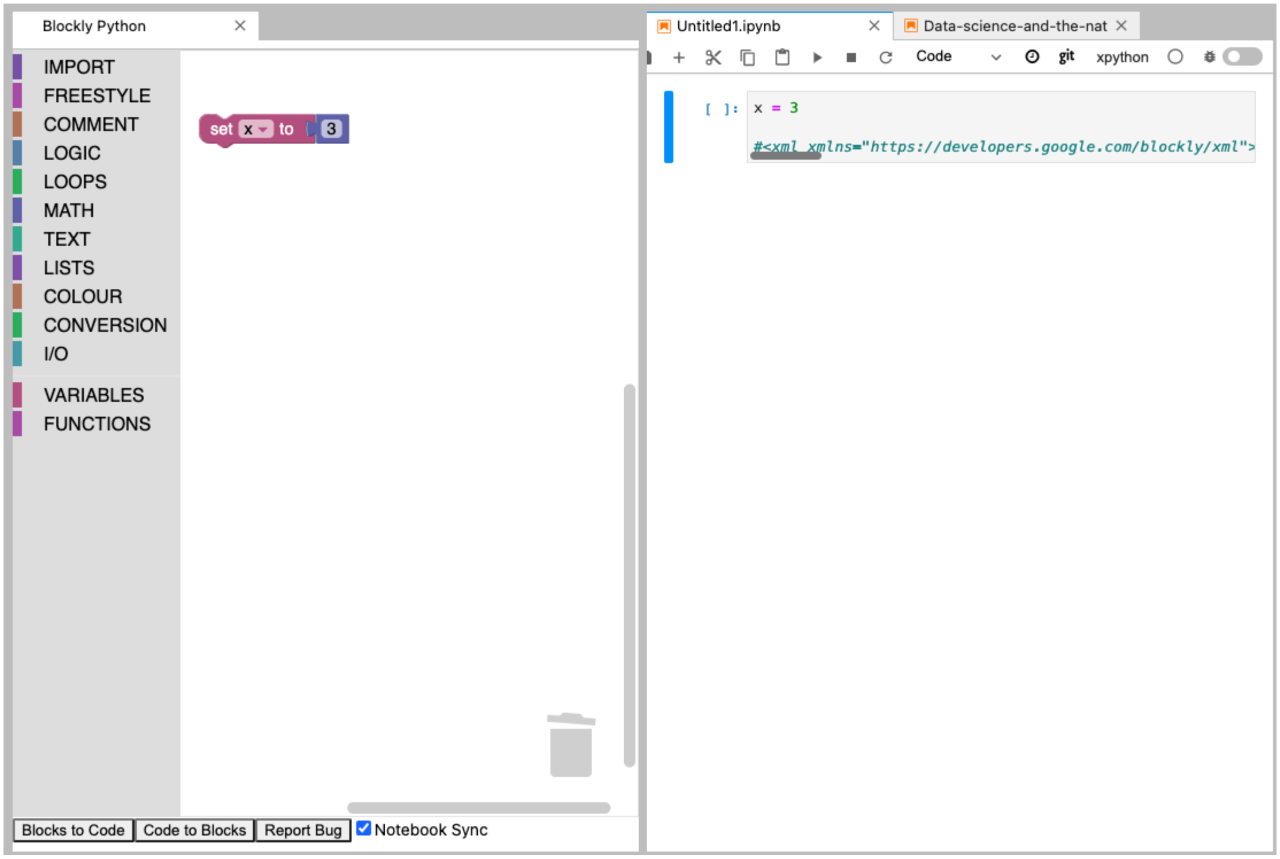
A posttest with eight word problems, multiple choice questions isomorphic to exercises in the notebooks was used to assess learning. Two UX surveys were used. The first was a five-item modified version of the system usability scale (SUS; Bangor et al., 2008; Lewis, 2018). An example of the modified item, "It was easy to learn from the system," is based on the original SUS item "I thought the system was easy to use." The second survey consisted of eight statements measuring the three elements of cognitive load, including extraneous cognitive load (ECL), germane cognitive load (GCL), and intrinsic cognitive load (ICL; Klepsch et al., 2017) and used a Likert-type scale. An example of an ECL survey item was "During this task, it was exhausting to find the important information." One of the GCL survey items used was "The task contained elements that helped me better understand the material." Alternatively, an example of an ICL was "For this task, many things needed to be kept in mind simultaneously." Finally, a demographics survey contained questions about gender, ethnicity, educational attainment, years of expertise in programming, statistics, and data science, and high stakes test scores (SAT, ACT, etc.). All materials were hosted on Qualtrics, except the Jupyter notebooks, which were hosted on a JupyterHub. Since the experiment was online, participants used their own computers and the Chrome browser (mobile devices and other browsers were blocked).

Procedure

Participants were recruited through their affiliated subject pool website and completed the entire experiment online using Qualtrics. After completing the informed conset form, participants were asked to read instructions about the experiment, including silencing their phones, not taking breaks, and completing the study in one session (approximately two hours). Next, participants were randomly assigned to one of the following conditions: (a) the Jupyter interface supported programming with blocks or (b) only supported programming with code (no blocks) (see Figure 1).

Figure 1

Sample Jupyter Notebook with Blockly Plug-In



```
x = 3
#<xml xmlns="https://developers.google.com/blockly/xml">
```

Note. Blockly plug-in appears on the left side, while the corresponding code appears on the right side.

Participants then watched instructional videos according to their condition and completed five notebooks in the order WE, NEAR1, NEAR2, FAR, and FAR+. Because the notebooks were on a JupyterHub, participants added them by clicking a link in Qualtrics, which opened another tab on their browser containing the notebook. When participants finished with a notebook, they clicked a link in the notebook that revealed a password and otherwise disabled the notebook interface. Participants had to enter the password correctly into Qualtrics to continue. This process prevented participants from backtracking or accessing more than a single notebook at a time with one important exception: they were permitted and encouraged to consult the worked example while solving NEAR1. This exception was accomplished by opening an additional tab for WE in Jupyter that was disabled when NEAR1 was completed.

Participants were included in the study if they completed the study and attempted a single cell on the worked example notebook (*n*=59). Some participants completed the study in more than one session; these participants were retained as long as they met the above inclusion criteria, even if they were assigned to a different condition in a later session. After the notebooks, participants took the posttest, two usability surveys, and the demographic survey, after which they were debriefed. Because the posttest came at the end of the experiment, participants who completed it tended to do so in the session in which they completed the experiment to preserve the instrument's integrity.

Results

Research Question 1

The first research question sought to understand the degree to which learning outcomes differ when learning is supported with blocks or without blocks. To answer this question, a one-way MANOVA was conducted to determine whether there is a difference between block-based programming on test scores, ICL, GCL, and ECL. Using Wilk's statistic, there was not a significant difference in test scores and cognitive load (ICL, GCL, ECL) based on block-based programming or no blocks programming, *F*(5, 53) = 1.17, *p* = .34, ƛ = .901, partial η^2 = .99.

Table 1 shows the average test scores for students and the results from the cognitive load survey given the instructional materials without the aid of block-based coding versus with blocks coding. In terms of RQ1.a, MANOVA findings reported above suggested no significant difference relative to test scores. In terms of RQ1.b, which was concerned with the three factors of cognitive load (ICL, GCL, and ECL), the MANOVA did not identify any significant difference between a block-based or no blocks programming approach on perceived ICL (*F* (1,57) = .89, *p* =.35, partial η^2= .02), GCL (*F* (1,57)= .20, *p* = .66, partial η^2= .004), and ECL (*F* (1,57)= 3.97, *p* = 0.51, partial η^2= 0.07).

Table 1

Means and Standard Deviations by Condition for Conceptual Knowledge (Test Score) and Cognitive Load Factors Across Treatment Conditions

	No Blocks (*n*=37)	Blocks (*n*=22)
Test Score	*M* = 51.35 (*SD* = 25.52)	*M* = 40.91 (*SD* = 8.67)
Intrinsic Cognitive Load	*M* = 73.14 (*SD* = 17.96)	*M* = 78.23 (*SD* =23.01)
Germane Cognitive Load	*M* = 63.51 (*SD* = 14.58)	*M* = 65.59 (*SD* =20.88)
Extraneous Cognitive Load	*M* = 55.38 *(SD = 23.12)*	*M* = 68.73 (*SD* =27.65)

Research Question 2

The second research question asked to what degree an aspect of *learning experience design* correlated with factors of *cognitive load*. That is, how *is* usability correlated with factors of *cognitive load* (intrinsic cognitive load, germane cognitive load, and extraneous cognitive load)? To answer this question, a correlation test was performed with responses to system usability scale (SUS) questions and cognitive load factors. All participants' SUS scores were averaged (*M* = 51.69, *SD* = 9.38), with the average SUS score falling below the threshold of 68 (Brooke, 1996), suggesting that ease-of-use was "OK" (Bangor et al., 2009). There was a significant weak correlation (Cohen, 1988) between the SUS learner responses and intrinsic cognitive load (*M* = 75.03, *SD* = 19.95, *r*(57) = .37, *p* = .004). There was a significant moderate correlation (Cohen, 1988) between the SUS learner responses and germane cognitive load (*M* = 64.29, *SD* = 17.05, *r*(57) = .66, *p* < .00). There was a significant weak correlation (Cohen, 1988) between the SUS learner responses and extraneous cognitive load (*M* = 60.36, *SD* = 25.51, *r*(57) = .33, *p* = 0.01).

Research Question 3

Whereas RQ2.a focused on cognitive load, the second part of RQ3 asked to what degree is *learning experience design* correlated with aspects of *conceptual knowledge* of computational thinking? To answer this second part of RQ3, a correlation test was performed with SUS and posttest scores. The correlation test for this research question showed no significant correlation between student SUS responses and test scores, (*M* = 51.69, *SD* = 9.38, *r*(57) = .15, *p* = .26).

Discussion

Computational thinking is an important component of learning within the STEM domain. Whereas other subdomains of STEM are largely conceptual, computational thinking necessitates the use of technology as students code and complete other programming tasks. This places a greater emphasis on the role of LXD and the interactions that are foundational to the tools that facilitate the learning process. Hence, it is important to explore the role of LXD as different scaffolding techniques are embedded within learning environments that support data science. To date, theorists have begun to theorize and formalize constructs of LXD (Chang & Kuwata, 2020; Gray, 2020); however, few quantitative studies exist that explore the relationship between LXD and learning outcomes. This study sought to address this gap by exploring the LXD in learning as it relates to cognitive load (intrinsic, germane, extrinsic) and conceptual knowledge within data science education.

The first research questions (RQ1.a, RQ1.b) sought to determine the degree to which there was a difference in learning outcomes (conceptual knowledge, ICL, GCL, ECL) when the data science learning environment employed blocks or no blocks. The research found no statistically significant differences between the conditions. This is somewhat surprising because one might hypothesize these additional supports engender various learning outcomes, especially given the literature associated with the importance of scaffolding during coding. There may be multiple interpretations for this finding. As it relates to learning outcomes in the form of test scores, it is possible that learners may have done well at the procedural aspect of coding *in situ*, but struggled on the testing activity which required learners to recognize and recall aspects of data science during testing. This data science coding process also has a significant degree of element interactivity in which learners need to understand the concept, perform functions, interpret the output, and iterate. It is possible that additional support is needed beyond the status quo or a block-based approach. Whereas blocks help to simplify the more technical aspects of coding, it may be that additional features are needed for the blocks that help with interpretation or next steps that are essential to data science.

Another finding relates to the LXD results that were the focus of the second research question, which specifically explored the relationship between LXD and the factors of cognitive load. Analysis found correlations with each factor of cognitive load, which is noteworthy because research often describes LXD as part of the evaluation portion that (e.g., ADDIE) occurs at the end of the development cycle (DeVaughn & Stefaniak, 2020; Lohr et al., 2003). As opposed to just user testing, this research suggests there may be a relationship between elements of LXD and *in situ* learning processes.

The finding of ECL correlated to LXD aligns with prior literature in which individuals speculate that ECL may be exacerbated by poor design of the learning environment (Mutlu-Bayraktar et al., 2019). Beyond just presentation of content and multimedia, the results extend the discussion suggesting that extraneous cognitive load may stem from both the design of learning materials *and* interactions that are embedded within the learning environment. In terms of intrinsic and germane cognitive load, one might conclude that designed LXD interactions (navigation, progression of content) can support learner's iterative knowledge construction for intrinsic load, which in turn supports schema formation associated with germane load. Whereas LXD was seen as a design aspect during the development process, the findings suggest LXD is related to cognitive load and thus an inherent part of the learning process.

Limitations and Future Studies

The current study utilized data from a larger research project to understand the degree to which learning outcomes differ with graphical supports (blocks vs. no blocks) and to explore the correlations among LXD and aspects of cognitive outcomes. Given that few research studies exist that explore the relationship between LXD and learning outcomes, our findings could be a step toward bridging that research gap. While the study did uncover correlations between the SUS and factors of cognitive load, the participants were limited to undergraduate psychology students. Considering this limitation, testing for duplication of results in other populations, such as adult learners in other fields of study or high school students, could provide further empirical validation of the results. Future research could replicate the findings with a large sample size and conditions.

In this study, participants were asked to complete the learning program and conceptual knowledge test in one session. Although many user experience studies leverage this type of methodology, one might argue it could impact the overall interaction. Instead, if the participants were provided access to interact with the learning materials/environment over multiple sessions/dates, this could influence how they view their own cognitive load upon completing the tasks provided. Therefore, a valuable future study would be to test and survey participants at various stages of familiarity with the learning materials to compare their perceptions of their own cognitive load levels at these different stages.

Lastly, the participant pool for the current study was offered a small incentive to participate in the study. However, it is not known if or how mandating the study (e.g., as part of a class assignment) would impact the results found. To answer that question, we recommend future research that replicates the study in a way that integrates the learning environment within the assigned classroom requirements. The current study also looked at self-reported cognitive load factors (i.e., ICL, GCL, ECL) and test scores, which helped us to understand correlations between an aspect of LXD and knowledge gains. However, the test items were conceptual in nature, whereas data science is more procedural in nature. Therefore, for this data set, it may have been beneficial to assess the accuracy of the participants' actual coding, rather than their post-hoc test scores. This may have helped us to better understand students' coding processes, which may more accurately represent their learning outcomes.

Funding

This material is based upon work supported by the National Science Foundation under Grant 1918751. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Armoni, M., Meerbaum-Salant, O., & Ben-Ari, M. (2015). From Scratch to "real" programming. *ACM Transactions in Computing Education, 14*(4), 1–15. https://doi.org/10.1145/2677087

Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from examples: Instructional principles from the worked examples research. *Review of Educational Research, 70*(2), 181. https://doi.org/10.3102/00346543070002181

Atkinson, R. K., Renkl, A., & Merrill, M. M. (2003). Transitioning from studying examples to solving problems: Effects of self-explanation prompts and fading worked-out steps. *Journal of Educational Psychology, 95*(4), 774–783. https://doi.org/10.1037/0022-0663.95.4.774

Bangor, A., Kortum, P., & Miller, J. (2009). Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of Usability Studies, 4*(3), 114-123. https://uxpajournal.org/determining-what-individual-sus-scores-mean-adding-an-adjective-rating-scale/

Bangor, A., Kortum, P. T., & Miller, J. T. (2008). An empirical evaluation of the System Usability Scale. *International Journal of Human–Computer Interaction, 24*(6), 574–594. https://doi.org/10.1080/10447310802205776

Bau, D., Gray, J., Kelleher, C., Sheldon, J., & Turbak, F. (2017). Learnable programming: Blocks and beyond. *Communications of the ACM, 60*(6), 72–80. https://doi.org/10.1145/3015455

Bottia, M. C., Mickelson, R. A., Jamil, C., Moniz, K., & Barry, L. (2021). Factors associated with college STEM participation of racially minoritized students: A synthesis of research. *Review of Educational Research, 91*(4), 614–648. https://doi.org/10.3102/00346543211012751

Brooke, J. (1996). SUS: A “quick and dirty” usability. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & I. L. McClelland (Eds.), *Usability evaluation in industry* (pp. 189-194). Taylor & Francis.

Carey, K. L., & Stefaniak, J. E. (2018). An exploration of the utility of digital badging in higher education settings. *Educational Technology Research and Development, 66*(5), 1211–1229. https://doi.org/10.1007/s11423-018-9602-1

Chang, Y. K., & Kuwata, J. (2020). Learning experience design: Challenges for novice designers. In M. Schmidt, A. A. Tawfik, I. Jahnke, & Y. Earnshaw (Eds.), *Learner and user experience research: An introduction for the field of learning design & technology*. EdTechBooks. https://edtechbooks.org/ux/LXD_challenges

Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science, 13*(2), 145–182. https://doi.org/10.1207/s15516709cog1302_1

Chi, M. T. H., De Leeuw, N., Chiu, M.-H., & Lavancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science, 18*(3), 439–477. https://doi.org/10.1207/s15516709cog1803_3

Clark, R. C., Nguyen, F., & Sweller, J. (2011). *Efficiency in learning: Evidence-based guidelines to manage cognitive load*. John Wiley & Sons.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences*. Lawrence Erlbaum.

Cunniff, N., Taylor, R. P., & Black, J. B. (1986). Does programming language affect the type of conceptual bugs in beginners' programs? A comparison of FPL and Pascal. *CHI '86: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 17*(4), 175–182. https://doi.org/10.1145/22627.22368

Dann, W., Cosgrove, D., Slater, D., Culyba, D., & Cooper, S. (2012). Mediated transfer: Alice 3 to Java. *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, 141–146. https://doi.org/10.1145/2157136.2157180

DeVaughn, P., & Stefaniak, J. (2020). An exploration of how learning design and educational technology programs prepare instructional designers to evaluate in practice. *Educational Technology Research and Development, 68*(6), 3299–3326. https://doi.org/10.1007/s11423-020-09823-z

Fernandez, F., Froschl, M., Lorenzetti, L., & Stimmer, M. (2022). Investigating the importance of girls' mathematical identity within United States STEM programmes: A systematic review. *International Journal of Mathematical Education in Science and Technology*, 1–41. https://doi.org/10.1080/0020739X.2021.2022229

Gray, C. (2020). Paradigms of knowledge production in human-computer interaction: Towards a framing for learner experience (LX) design. In M. Schmidt, A. A. Tawfik, I. Jahnke, & Y. Earnshaw (Eds.), *Learner and user experience research: An introduction for the field of learning design & technology*. EdTechBooks. https://edtechbooks.org/ux/paradigms_in_hci

Jahnke, I., Schmidt, M., Pham, M., & Singh, K. (2020). Sociotechnical-pedagogical usability for designing and evaluating learner experience in technology-enhanced environments. In M. Schmidt, A. A. Tawfik, I. Jahnke, & Y. Earnshaw (Eds.), *Learner and user experience research: An introduction for the field of learning design & technology*. EdTechBooks. https://edtechbooks.org/ux/sociotechnical_pedagogical_usability

Kaggle. (2021). *State of Data Science and Machine Learning 2021*. https://www.kaggle.com/kaggle-survey-2021

Klepsch, M., Schmitz, F., & Seufert, T. (2017). Development and validation of two instruments measuring intrinsic, extraneous, and germane cognitive load. *Frontiers in Psychology, 8*, 1997. https://doi.org/10.3389/fpsyg.2017.01997

Lewis, J. R. (2018). The System Usability Scale: Past, present, and future. *International Journal of Human–Computer Interaction, 34*(7), 577–590. https://doi.org/10.1080/10447318.2018.1455307

Lim, C., Song, H.-D., & Lee, Y. (2012). Improving the usability of the user interface for a digital textbook platform for elementary-school students. *Educational Technology Research and Development, 60*(1), 159–173. https://doi.org/10.1007/s11423-011-9222-5

Lohr, L., Javeri, M., Mahoney, C., Gall, J., Li, K., & Strongin, D. (2003). Using rapid application development to improve the usability of a preservice teacher technology course. *Educational Technology Research and Development, 51*(2), 41–55. https://doi.org/10.1007/BF02504525

Marchewka, J. T., Liu, C., & Kostiwa, K. (2007). An application of the UTAUT model for understanding student perceptions using course management software. *Communications of the IIMA, 7*(2), 93. https://doi.org/10.58729/1941-6687.1038

Moskal, B., Lurie, D., & Cooper, S. (2004). Evaluating the effectiveness of a new instructional approach. *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education*, 75–79. https://doi.org/10.1145/1028174.971328

Mutlu-Bayraktar, D., Cosgun, V., & Altan, T. (2019). Cognitive load in multimedia learning environments: A systematic review. *Computers & Education, 141*, 103618. https://doi.org/10.1016/j.compedu.2019.103618

Nokes, T. J., Hausmann, R. G. M., VanLehn, K., & Gershman, S. (2011). Testing the instructional fit hypothesis: the case of self-explanation prompts. *Instructional Science, 39*(5), 645–666. https://doi.org/10.1007/s11251-010-9151-4

Novak, E., Daday, J., & McDaniel, K. (2018). Assessing intrinsic and extraneous cognitive complexity of e-textbook learning. *Interacting with Computers, 30*(2), 150–161. https://doi.org/10.1093/iwc/iwy001

Olney, A. M., & Fleming, S. D. (2021). JupyterLab extensions for blocks programming, self-explanations, and HTML injection. In T. W. Price & S. San Pedro (Eds.), *Joint Proceedings of the Workshops at the 14th International Conference on Educational Data Mining*, Vol. 3051, CSEDM–8. https://ceur-ws.org/Vol-3051/CSEDM_8.pdf

Pashler, H., Bain, P., Bottge, B., Graesser, A., Koedinger, K., McDaniel, M., & Metcalfe, J. (2007). *Organizing instruction and study to improve student learning* (NCER 2007–2004). National Center for Education Research, Institute of Education Sciences, U.S. Department of Education. http://ncer.ed.gov

Payne, L. A., Tawfik, A., & Olney, A. (2021). Datawhys Phase 1: Problem solving to facilitate data science & STEM learning among summer interns. *International Journal of Designs for Learning, 12*(3), 102–117. https://doi.org/10.14434/ijdl.v12i3.31555

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM, 52*(11), 60–67. https://doi.org/10.1145/1592761.1592779

Schmidt, M., & Glaser, N. (2021). Investigating the usability and learner experience of a virtual reality adaptive skills intervention for adults with autism spectrum disorder. *Educational Technology Research and Development, 69*(3), 1665–1699. https://doi.org/10.1007/s11423-021-10005-8

Schmidt, M., & Huang, R. (2022). Defining learning experience design: Voices from the field of learning design & technology. *TechTrends, 66*(2), 141–158. https://doi.org/10.1007/s11528-021-00656-y

Skulmowski, A., & Xu, K. M. (2022). Understanding cognitive load in digital and online learning: A new perspective on extraneous cognitive load. *Educational Psychology Review, 34*(1), 171–196. https://doi.org/10.1007/s10648-021-09624-7

Swanstrom, R. (2020, November 14). *Data science colleges and universities*. Ryan Swanstrom; Data Science 101. https://ryanswanstrom.com/colleges/

Sweller, J. (2010). Element interactivity and intrinsic, extraneous, and germane cognitive load. *Educational Psychology Review, 22*(2), 123–138. https://doi.org/10.1007/s10648-010-9128-5

Sweller, J. (2020). Cognitive load theory and educational technology. *Educational Technology Research and Development, 68*(1), 1–16. https://doi.org/10.1007/s11423-019-09701-3

Sweller, J., & Cooper, G. A. (1985). The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction, 2*(1), 59–89. https://doi.org/10.1207/s1532690xci0201_3

Tawfik, A. A., Gatewood, J., Gish-Lieberman, J., & Hampton, A. (2022). Toward a definition of learning experience design. *Technology, Knowledge, & Learning, 27*(1), 309–334. https://doi.org/10.1007/s10758-020-09482-2