A Dynamic AI Scaffolding System Based on ZPD Theory: Reducing Cognitive Load and Enhancing Transfer in Mathematical Proof Learning

Ning Yao[1], Pengtao Huang[2,*]
[1]School of Mathematics and Physics, Hechi University, Yizhou, Guangxi, China
[2]Department of Science and Technology, Hechi University, Yizhou, Guangxi, China
**Corresponding Author*

Abstract: Although large language models (LLMs) offer new possibilities for personalized instruction, most educational implementations rely on static prompting that ignores learners' moment-to-moment cognitive and metacognitive needs. Grounded in Vygotsky's Zone of Proximal Development, this study introduces a four-layer dynamic AI scaffolding system that adaptively regulates LLM support during complex problem solving. In a randomized controlled experiment, 60 non-mathematics majors solved series convergence proof problems under either dynamic scaffolding or static prompting conditions. Learners receiving dynamic scaffolding reported lower extraneous cognitive load, demonstrated superior far-transfer performance, and engaged in more metacognitive questioning. Within the dynamic condition, metacognitive questioning was positively associated with transfer outcomes. These results indicate that dynamically operationalizing ZPD in generative AI tutors is more effective than static prompting for fostering deep learning and knowledge transfer.

Keywords: AI in Education; Zone of Proximal Development (ZPD); Cognitive Load Theory (CLT); Intelligent Tutoring System (ITS); Mathematical Problem Solving; Prompt Engineering

1. Introduction

The integration of Artificial Intelligence (AI), particularly Large Language Models (LLMs), into education is rapidly reshaping the landscape of intelligent tutoring systems (ITS). Models such as GPT-4 demonstrate an unprecedented capacity to generate coherent explanations, provide step-by-step guidance, and engage in open-ended dialogue, making them powerful potential partners in learning [1]. In structured domains like mathematics, research indicates that well-prompted LLMs can effectively tutor students and guide multi-step reasoning [2].

Despite this promise, a critical limitation persists. Most current LLM-based educational applications employ astatic and generic interaction mode, relying on fixed prompt templates that are invariant to the learner's ongoing cognitive state [3]. From an educational psychology perspective, this "one-size-fits-all" approach risks inducing extraneous cognitive load, a type of mental effort that does not contribute to learning and can actively hinder it [4]. An AI tutor that cannot perceive a learner's current level of understanding may offer overly vague hints when concrete guidance is needed or supply redundant information when the learner is ready for higher-order reflection. This "support mismatch" represents a new source of ineffective instructional design within AI-powered learning.

The key to resolving this contradiction lies in making AI support dynamic and adaptive, capable of mimicking the "scaffolding" behavior of an expert human tutor. Lev Vygotsky's theory of the Zone of Proximal Development (ZPD)provides the foundational framework [5]. The ZPD is defined as the distance between a learner's ability to solve problems independently and their potential ability when guided by a more knowledgeable other. Effective instruction, or scaffolding, must be dynamically adjusted: providing support when necessary and gradually withdrawing it as the learner's competence increases. While ZPD and scaffolding have been extensively discussed in traditional pedagogy and earlier, rule-based ITS [6], a significant research gap remains: How to

translate the essence of dynamic scaffolding into operational, computable rules that can be deeply integrated into a conversational LLM-based tutoring system.

Therefore, this study sits at the intersection of educational psychology and artificial intelligence. It addresses the core research question: How can the classic ZPD paradigm be transformed into a computational framework that drives an LLM to provide dynamic, adaptive learning support? We investigate this within the context of university-level mathematical proof learning (specifically, series convergence proofs), a domain characterized by high cognitive demand and clear hierarchical skill structures. Through the design, implementation, and empirical testing of a four-layer dynamic AI scaffolding system, this research explores the framework's efficacy in optimizing cognitive load and enhancing transfer of learning. Our goal is to provide both empirical evidence and a concrete design exemplar for the next generation of "theory-informed" intelligent educational technology.

2. Theoretical Foundation and Related Work

2.1 Core Educational Psychology Theories

2.1.1 Vygotsky's zone of proximal

(1) Development and Scaffolding Theory

Vygotsky's socio-cultural theory posits that learning first occurs through social interaction before being internalized. The ZPD is the central concept, defining the region where instruction is most effective [7]. Scaffolding, derived from this theory, refers to the temporary, adaptive support provided by a teacher or peer to help a learner cross the ZPD. This process involves diagnosing the learner's current level, providing tailored support (e.g., modeling, questioning, task structuring), and gradually transferring responsibility as the learner's ability grows—a cycle of "fading" support. Its effectiveness hinges on continuous diagnosis and responsive adjustment [8].

(2) Guidance for this study: This theory provides the foundational design philosophy. Our AI system must be a dynamic diagnostician and responsive agent, not a static information repository. The "four-layer prompt framework" is an operationalization of this philosophy, aiming to approximate the ZPD and simulate the "fading" process.

2.1.2 Sweller's cognitive load theory

Cognitive Load Theory (CLT) is a cornerstone of instructional design, based on the severe limitations of human working memory. It distinguishes three types of cognitive load [9]:

(1) Intrinsic Cognitive Load: Imposed by the inherent complexity and element interactivity of the learning material.

(2) Extraneous Cognitive Load: Caused by suboptimal instructional design or presentation (e.g., confusing layout, split attention) and does not aid learning.

(3) Germane Cognitive Load: The mental effort devoted to schema construction and automation, which directly facilitates learning.

The goal of instructional design is to manage total load by minimizing extraneous load, optimizing intrinsic load (e.g., through segmenting), and maximizing germane load. The "expertise reversal effect" further underscores the need for adaptive support, as instructional techniques that help novices can hinder experts.

(4) Guidance for this study: CLT provides our primary evaluation metric (extraneous load) and core design principles. Our hypothesis that dynamic scaffolding reduces extraneous load is rooted here. The system design aims to minimize irrelevant load through a clear interface and precise language, while the dynamic mechanism manages total load by offering structured support during high intrinsic load and prompting reflection to encourage germane processing.

2.1.3 Metacognition and self-regulated

(1) Learning Theory

Metacognition—"cognition about cognition"—encompasses knowledge about one's own thinking and the active monitoring and regulation of cognitive processes [10]. Self-Regulated Learning (SRL)is its application in academic settings, describing the cyclical process where learners set goals, select strategies, monitor progress, and adjust their approach. Strong metacognitive skills are strongly correlated with academic achievement, and fostering these skills is a key challenge for ITS design [11].

(2) Guidance for this study: This theory sets a higher-order goal for our AI system: to cultivate the learner's capacity for independent problem-solving, not just solve the immediate problem. This is directly instantiated in theL3 (Metacognitive Guidance) prompt layer, designed to externalize strategic thinking (e.g.,

"Why did you choose this method?") and prompt self-monitoring, thereby acting as a "metacognitive tutor."

2.2 Artificial Intelligence in Education

2.2.1 The evolution of intelligent tutoring systems

ITS have evolved through distinct generations: Model-based tutors (e.g., Cognitive Tutors), which relied on hand-coded expert models and production rules to provide precise feedback but were costly to build and inflexible; Data-driven adaptive learning systems, which use educational data mining for personalization but often lack deep explanatory dialogue; LLM-based generative dialogue tutors, which offer unparalleled generality and ease of deployment but typically lack an underlying pedagogical strategy, acting as reactive information sources rather than proactive pedagogical guides[12].

2.2.2 Prompt engineering in education

Prompt engineering is crucial for harnessing LLMs for educational purposes. Current research focuses on role-playing ("Socratic tutor"), advanced techniques like chain-of-thought prompting to elicit reasoning, and optimizing prompts for specific tasks like explanation or feedback generation. However, most work treats prompts as static, single-turn inputs, optimizing for immediate output quality rather than designing a dynamic sequence of pedagogical strategies across a sustained interaction [13].

2.2.3 The research gap: integrating dynamic theory with dynamic prompting

A significant disconnect exists. While learning science offers rich theories of adaptive support (ZPD, CLT), they remain largely descriptive for LLM contexts. Conversely, AI research focuses on output quality, not long-term pedagogical outcomes like transfer or metacognitive growth. There is a pressing need for a systematic framework that translates dynamic psychological principles into a computable decision-making process for LLM interaction, addressing: (1) how to quantify learner state from interaction data, (2) how to map this state to an optimal support strategy, and (3) how to execute this strategy via targeted prompting.

2.3 The Cognitive Process of Mathematical Problem Solving

2.3.1 Cognitive model of mathematical proof

Mathematical expertise requires the integration of procedural knowledge (how to execute steps) and conceptual knowledge (understanding why and when to apply them). Experts possess rich, interconnected cognitive schemas, allowing flexible strategy selection. Novices often have fragmented knowledge, possessing procedures without the conceptual understanding for proper application or transfer. Proof construction involves a cyclical process of planning(strategic), executing(tactical), and verifying [14], with novices struggling to move beyond local execution details.

2.3.2 Common obstacles and sources of cognitive load

Key obstacles in mathematical learning align with CLT:

(1) Working Memory Overload: The high element interactivity in proofs strains limited working memory, creating high intrinsic load.

(2) Inefficient Problem-Solving Search: Poor problem representation leads to means-ends analysis, generating high extraneous load [15].

(3) Deficient Metacognitive Monitoring: Failure to plan, monitor, and evaluate one's approach consumes resources wastefully [16].

(4) Inert Knowledge: Lack of conceptual understanding prevents transfer, as learners cannot adapt known procedures to novel situations [17].

These obstacles highlight the need for external support that can manage load, bridge the conceptual-procedural gap, and provoke metacognitive reflection—the precise goals of our dynamic scaffolding system.

3. Dynamic AI Scaffolding System Design

3.1 Overall Architecture

The system implements a Perceive-Decide-Actclosed loop, comprising five core modules (see Figure 1):

(1) Learner Module: An internal state representation (task progress, help count, inferred confusion/query type).

(2) Task Interface Module: Presents problems, captures help requests and free-form queries.

(3) Decision Engine Module: The "brain." Applies a rule-based logic to the learner state to select a support level.

(4) Structured Prompt Library Module: A repository of pre-authored prompt templates for each of the four support levels, tailored to specific task steps.

(5) AI Model (LLM) Module: The "strategy executor." Given the prompt template from the library, it generates the final, contextualized response to the learner.

(6) The interaction flows cyclically: Learner acts → State updated → Decision Engine selects level → Corresponding prompt retrieved → LLM generates final response → Response presented to learner.

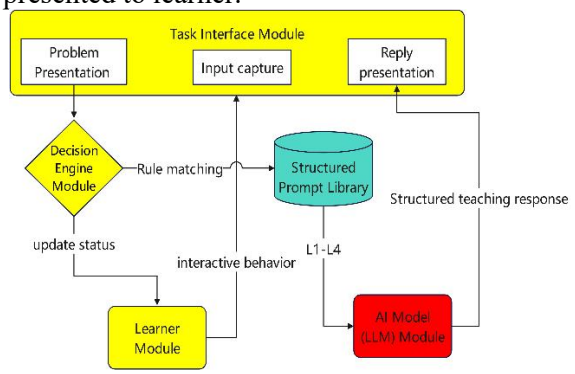

Figure 1. System Architecture Diagram

3.2 Four-Level Operationalization of ZPD

The scaffolding concept is operationalized into four discrete levels:

Level 1 (L1): Direct Guidance

Provides explicit steps, definitions, or formulas.

Goal: Reduce initial intrinsic/extraneous load, establish a correct starting point.

Example: "For series $\sum n/2^n$, the ratio test is often effective. Try calculating $\lim_{n\to\infty} |a_{n+1}/a_n|$."

Level 2 (L2): Heuristic Questioning

Poses guiding questions focused on key concepts.

Goal: Focus attention, promote active reasoning, encourage germane processing.

Example: "You simplified the ratio to $(n+1)/(2n)$. What is its limit as $n\to\infty$? If $L<1$, what does the ratio test conclude?"

Level 3 (L3): Metacognitive Guidance

Prompts reflection on strategy selection and process evaluation.

Goal: Foster metacognition and self-regulated learning.

Example: "What was your clue to choose the ratio test? Would you choose the same method for $\sum n^2/2^n$? Why?"

Level 4 (L4): Motivational & Emotional Support

Provides encouragement and task decomposition.

Goal: Manage frustration, maintain self-efficacy, reduce emotionally-induced extraneous load.

Example: "Limit calculations require care. You've persevered well. Let's break it down: first, ensure the ratio formula is correct, then focus on simplifying."

3.3 Dynamic Decision Rule Engine

The engine maps real-time interaction data to a support level using a transparent rule set.

Inputs: help_count, current_step, query_text, detection of words from a negative emotional_lexicon (e.g., "too hard", "confused").

Rule Set (Prioritized Order):

pseudocode

```
IF emotional_lexicon.detect_nega"tive(quer" "y"
_"t"  "ext)THENRETU" RN L4
ELSE IF help_count == 1 THEN RETURN L1
ELSE IF query_text.contains("why") OR
query_text.contains("how to choose") THEN
RETURN L3
ELSE IF current_step ==
"calculation_execution" THEN RETURN L2
ELSE IF help_count >= 2 THEN RETURN L3
ELSE RETURN L2 // Default rule
```

Output: A tuple <Selected_Level, Corresponding_Prompt_Template>.

3.4 Technical Implementation

A fully functional prototype was developed using Pythonand the Gradio library for the web interface. For experimental control and to isolate the effect of the dynamic logic itself, the system did not call a live LLM API during the study. Instead, it used apre-scripted, high-quality prompt library (stored in JSON format, keyed by task, step, and level). The Decision Engine's output directly triggered the retrieval and display of the corresponding pre-authored text. This ensured consistency in response quality across all participants, with the sequence and timing of support levels being the only manipulated variable.

4. Experimental Method

4.1 Experimental Design

A between-subjects, randomized controlled trial was conducted.

Independent Variable: AI support type with two levels: Dynamic Scaffolding (Experimental Group) vs. Static Prompting (Control Group). The control group received prompts randomly chosen from L1 or L2 libraries, irrespective of their interaction context.

Dependent Variables:

(1) Cognitive Load: NASA-TLX total score.
(2) Transfer Learning: Score on a novel post-test problem.
(3) Help-Seeking Behavior: Metrics from dialogue logs (frequency, proportion of metacognitive questions).
Control: Participants were randomly assigned. The task set, interface, and total time were identical across groups. The experiment followed a single-blind procedure.

4.2 Participants

60 first-year undergraduate students from Hechi University, all non-mathematics majors (28 male, 32 female, mean age=18.5, SD=0.7), participated. All had completed a calculus course covering series basics but had no formal training in proof techniques. Participants were randomly assigned to the Dynamic n=30 or Static n=30 group.

4.3 Materials and Tools

Learning Tasks: Three series convergence proof problems of ascending difficulty:
(T1) $\sum 1/n^2$ (p-series);
(T2) $\sum n/2^n$ (ratio test);
(T3) $\sum (\ln n)/n^2$ (comparison test).
Post-test Transfer Task: One novel problem: $\sum n^2+1 /(n^3\sqrt{n})$, requiring algebraic simplification before recognition as a convergent p-series.
Measurement Tools:
NASA-TLX Scale (Chinese version): Measured total cognitive load after the learning phase (Cronbach's α = .82 in this study).
The Dynamic AI Scaffolding System: As described in Section 3.
Automated Logging System: Recorded all interactions (timestamp, user_id, task, query, triggered rule, system response).

4.4 Procedure

The 85-minute experiment proceeded as follows:
(1) Pre-test & Group Assignment (10 mins): Background questionnaire, followed by automated, concealed random assignment to a group.
(2) Learning Phase (45 mins): Participants solved T1, T2, T3 sequentially (15 mins each). Unlimited help requests were allowed, answered according to group assignment.
(3) Cognitive Load Measurement (5 mins): Participants completed the NASA-TLX scale.
(4) Post-test Phase (15 mins): Participants solved the transfer problem without any AI assistance.
(5) Brief Interview & Debrief (10 mins): A subset of participants provided subjective feedback.

4.5 Data Analysis Plan

Data were analyzed using SPSS 26.0 α=.05.
(1) Cognitive Load: Independent samples t-test on NASA-TLX scores.
(2) Transfer Score: Independent samples t-test on post-test scores (0-5point rubric, inter-rater reliability > .85).
(3) Help-Seeking Behavior:
a. Independent t-test on total help counts;
b. Dialogue coding into three categories: Direct Answer Request, Conceptual Clarification, Metacognitive/Strategic Question (Cohen's Kappa > .80);
c. Chi-square test comparing the proportion of metacognitive questions between groups;
d. Pearson correlation between metacognitive question proportion and post-test score within the dynamic group.

5. Results

5.1 Descriptive Statistics

Random assignment was successful. No significant differences were found between groups in age, gender, or prior mathematics grade (all *p*> .05). Descriptive statistics for help-seeking are shown in Table 1. The dynamic group had a non-significantly higher mean total help count.

Table 1. Descriptive Statistics for Help-Seeking Behavior

Group	Total Help Count (M±SD)	Task 1 Help	Task 2 Help	Task 3 Help
Dynamic	8.20 ± 2.31	2.23 ± 0.81	2.87 ± 0.92	3.10 ± 1.05
Static	7.63 ± 2.84	2.40 ± 0.97	2.57 ± 1.07	2.67 ± 1.21
t(58)	0.86	-0.73	1.16	1.49
p	.395	.471	.252	.142

5.2 Hypothesis Testing Results

5.2.1 Hypothesis
(1) H1 (Cognitive Load):Supported. The dynamic group M=52.30,SD=8.71 reported significantly lower cognitive load than the static group M=59.17,SD=9.84 .t(58)=2.87,p=.006, Cohen'sd=0.74.
(2) H2 (Transfer Learning):Supported. The dynamic group M=3.63,SD=1.03 scored

significantly higher on the post-test than the static group M=2.57,SD=1.27,t(58)=3.21,p=.002, Cohen'sd=0.83.

(3) H3 (Metacognition): Supported.

5.2.2 Trend and correlation

A repeated-measures ANOVA on the metacognitive question proportion in the dynamic group showed a significant main effect of task order $F(2,58)=5.43, p=.006, \eta^2=0.16$. Post-hoc tests confirmed the proportion in Task 3 was significantly higher than in Task 1 $p=.012$. Within the dynamic group, the overall metacognitive question proportion was positively correlated with post-test score $r(28)=.42, p=.010$.

5.3 Exploratory Findings

5.3.1 Task completion time

The dynamic group spent slightly more total time on the learning phase (38.5±4.2 min) than the static group (36.8±5.1 min), though this difference was not statistically significant $t(58)=1.41, p=.164$. This may suggest deeper engagement.

5.3.2 Emotional language

After the third help request, the static group used significantly more negative emotional words in their queries than the dynamic group $\chi^2(1)=4.12, p=.042$.

6. Discussion

6.1 Interpretation of Key Findings

The results robustly support the efficacy of a theory-driven dynamic scaffolding system. The significant reduction in cognitive load (H1) demonstrates that "right-timing, right-granularity" support is key to managing extraneous load. Static prompts often create "support mismatch," forcing learners to expend extra effort to interpret unhelpful aid, thereby increasing extraneous load. The dynamic system's adaptive logic aims to provide a "tight fit," reducing friction and freeing working memory resources for genuine learning.

The superior transfer performance (H2) and the rising trend and positive correlation of metacognitive questioning (H3) jointly illuminate the pathway to deeper learning. The system does not just answer questions; it systematically structures the interaction to provoke higher-order thinking. By responding to strategic confusion with L3 prompts (e.g., "What was your clue?"), it externalizes the expert's internal metacognitive dialogue. The increasing proportion of metacognitive questions indicates learners were internalizing this reflective stance, which facilitates the transformation of inert procedural knowledge into flexible, transferable strategic schemas.

6.2 Theoretical Contributions

This study makes a primary contribution by providing a computational model that operationalizes ZPD and scaffolding within an LLM-based dialogue system. It moves these powerful psychological metaphors from descriptive principles to an implementable, rule-based framework for real-time pedagogical decision-making.

Furthermore, it highlights that in AI-supported learning, the granularity and the timing of support are two critical, separable design dimensions that jointly influence cognitive and metacognitive outcomes. Our framework shows that even with a simple rule engine (controlling timing), significant learning gains can be achieved over static prompts (which only fix granularity), deepening our understanding of how "pedagogical presence" can be algorithmically instilled.

6.3 Practical Implications

For educators and instructional designers, this work demonstrates a viable path to embodying classic pedagogical wisdom in modern AI tools. It argues that the effectiveness of an AI tutor depends as much on its theoretically-grounded interaction design as on the raw power of its underlying model.

For educational technology developers, it offers a lightweight, interpretable, and practical alternative to building opaque, data-hungry adaptive models. The rule-based approach is transparent (crucial for educational equity), easy to debug and modify, and requires no training data, making it a highly feasible engineering paradigm for developing effective ITS.

6.4 Limitations and Future Directions

6.4.1 Limitations

A sample drawn from a single university, limiting generalizability.

A focus on mathematical proofs, requiring validation in other domains (e.g., programming, scientific writing).

Reliance on relatively superficial interaction proxies (help count, keywords) for state

inference.

6.4.2 Future research

Incorporate multi-modal data (e.g., eye-tracking, facial expression analysis, electrodermal activity) to create a more fine-grained and accurate model of learner cognitive-affective state.

Explore hybrid decision mechanisms that combine the interpretability of rules with the adaptability of reinforcement learning, allowing the system to optimize its scaffolding policies based on long-term learning outcomes.

Generalize the framework to a wider array of disciplines and task types, testing its utility as a general-purpose design methodology for generative AI in education.

7. Conclusion

This study successfully translated Vygotsky's theory of the Zone of Proximal Development into a functional dynamic AI scaffolding system and provided rigorous empirical evidence for its effectiveness in the context of learning mathematical proofs. The results demonstrate that an AI prompt design informed by and dynamically aligned with educational psychology principles—specifically targeting cognitive load management and metacognitive activation—is significantly more effective than generic, static prompting in reducing extraneous mental effort and fostering transferable learning. This work provides both a concrete computational framework and a strong empirical foundation for the development of more intelligent, responsive, and ultimately more humane educational AI.

Acknowledgments

This paper is supported by Hechi University Education and Teaching Reform Project in 2025(NO.2025EB007).

References

[1] Kasneci E, Seßler K, Küchemann S, et al. ChatGPT for good? On opportunities and challenges of large language models for education. Learning and Individual Differences, 2023, 103: 102274.

[2] Kojima T, Gu S S, Reid M, Matsuo Y, Iwasawa Y. Large language models are zero-shot reasoners. Advances in Neural Information Processing Systems, 2022, 35: 22199–22213.

[3] Savel'ska O, Savel'skyy O. Prompt engineering in education: Towards a systematic framework for teaching and learning with large language models. arXiv preprint, arXiv:2310.16746, 2023.

[4] Sweller J. Cognitive load during problem solving: Effects on learning. Cognitive Science, 1988, 12(2): 257–285.

[5] Vygotsky L S. Mind in Society: The Development of Higher Psychological Processes. Cambridge, MA: Harvard University Press, 1978.

[6] VanLehn K. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. Educational Psychologist, 2011, 46(4): 197–221.

[7] Alvarez I, Ruiz M A. Scaffolding metacognitive strategies in LLM-based tutoring: Effects on self-regulated learning and problem-solving transfer. Computers & Education, 2023, 205: 104888.

[8] Van de Pol J, Volman M, Beishuizen J. Scaffolding in teacher–student interaction: A decade of research. Educational Psychology Review, 2010, 22(3): 271–296.

[9] Sweller J, van Merriënboer J J G, Paas F. Cognitive architecture and instructional design: 20 years later. Educational Psychology Review, 2019, 31(2): 261–292.

[10] Flavell J H. Metacognition and cognitive monitoring: A new area of cognitive–developmental inquiry. American Psychologist, 1979, 34(10): 906–911.

[11] Azevedo R, Aleven V, eds. International Handbook of Metacognition and Learning Technologies. New York: Springer, 2013.

[12] Kasneci E, Seßler K, Küchemann S, et al. ChatGPT for good? On opportunities and challenges of large language models for education. Learning and Individual Differences, 2023, 103: 102274.

[13] Wiggins J B, Grafsgaard J F. Do adaptive scaffolds improve help-seeking in an intelligent tutoring system?. Proceedings of the 12th International Learning Analytics and Knowledge Conference (LAK '22). New York: ACM, 2022: 220–230

[14] Hoyle E, Carpenter A. The cognitive structure of mathematical proof. Journal for Research in Mathematics Education, 2019, 50(3): 315–340.

[15] Sweller J. Element interactivity and intrinsic, extraneous, and germane cognitive load. Educational Psychology Review, 2010, 22(2): 123–138.

[16] Schoenfeld A H. Learning to think

mathematically: Problem solving, metacognition, and sense making in mathematics. In: Grouws D A, ed. Handbook of Research on Mathematics Teaching and Learning. New York: Macmillan, 1992: 334–370.

[17]Sweller J, van Merriënboer J J G, Paas F. Cognitive architecture and instructional design: 20 years later. Educational Psychology Review, 2019, 31(2): 261–292

