

Computational thinking and conceptual learning in first-year calculus: The role of teacher competence

Akbota Nurgaliyeva ^{1*} , Angisin Seitmuratov ¹ 

¹ Korkyt Ata Kyzylorda University, Aiteke Bi Street 29A, 120000, Kyzylorda, KAZAKHSTAN

*Corresponding Author: akbota.sci@gmail.com

Citation: Nurgaliyeva, A., & Seitmuratov, A. (2026). Computational thinking and conceptual learning in first-year calculus: The role of teacher competence. *International Electronic Journal of Mathematics Education*, 21(4), em0899. <https://doi.org/10.29333/iejme/19694>

ARTICLE INFO

Received: 10 May 2026

Accepted: 10 Sep. 2026

ABSTRACT

The persistent gap between procedural fluency and conceptual understanding is one of the most challenging problems in first-year calculus education. In many developing educational systems, calculus instruction is dominated by rule-based symbolic manipulation, limited lesson preparation, and little integration of computational tools, so students often perform acceptably on routine differentiation tasks while showing weak conceptual understanding of rate of change, optimization, and graphical interpretation. This study examines the role of computational thinking and teacher competence in improving conceptual learning in introductory calculus. A quasi-experimental comparative classroom study was conducted with 144 first-year engineering students over a four-week instructional intervention on derivatives and optimization, comparing traditional procedural teaching, structured conceptual teaching, and computationally enhanced conceptual teaching. Learning was assessed through pre-tests, post-tests, delayed retention tests, and transfer tasks across five dimensions: conceptual understanding, procedural fluency, transfer ability, computational reasoning, and engagement. One-way ANOVA revealed statistically significant differences among the three groups, $F(2, 141) = 19.42$, $p < .001$, with the computationally enhanced group achieving the highest post-test performance, retention, and transfer ability, and large effect sizes for every pairwise comparison. The findings suggest that computational thinking, when embedded in pedagogically competent teaching, substantially improves calculus learning, and the study proposes a scalable instructional framework for educational systems seeking to strengthen mathematics teaching through computational pedagogy.

Keywords: calculus education, computational thinking, teacher competence, conceptual learning, engineering mathematics, differential calculus

INTRODUCTION

Calculus occupies a central position in modern mathematics and is one of the most fundamental subjects in engineering, economics, physics, and computer science (Goodfellow et al., 2016; Stewart et al., 2021). In higher education, first-year calculus instruction functions not only as a mathematical subject in its own right but also as the language through which students later encounter mathematical modeling, optimization, numerical analysis, and scientific computation. Among the core concepts of introductory calculus, the derivative remains one of the most intellectually significant yet pedagogically difficult ideas: it represents the instantaneous rate of change, local linear approximation, the slope of a tangent line, sensitivity, and optimization behavior, and understanding it conceptually enables students to interpret mathematical relationships rather than merely manipulate formulas. Despite its importance, research in mathematics education has consistently shown that students experience substantial conceptual difficulty when first learning derivatives.

A major challenge in calculus teaching lies in the persistent gap between procedural fluency and conceptual understanding (Afriani et al., 2026; Durandt, 2026; Harahap et al., 2025). Many students can successfully apply differentiation rules such as the power, product, or chain rule, yet fail to interpret what the derivative means geometrically (Tarasov, 2016), numerically (Mazzia, 2022), or contextually (Ocal, 2017); students often know how to compute derivatives but struggle to explain why derivatives change sign, why critical points correspond to maxima or minima, or how derivative behavior reflects the local geometry of a function. This challenge becomes even more pronounced in many developing educational systems, where mathematics instruction often remains strongly procedural, examination-driven, and teacher-centered (Hassan et al., 2026). In such settings, teachers may enter classrooms with limited lesson preparation, insufficient pedagogical training, and little exposure to computational or visualization-based approaches to mathematics teaching (Özgeldi & Aydın, 2021), so calculus is frequently taught as a sequence of symbolic rules to memorize rather than as a coherent system of ideas to understand.

Table 1. Comparison of existing technology-assisted approaches in calculus education

Approach	Educational strength	Common limitation
GeoGebra / dynamic graphing	Visualization of derivatives and function behavior	Limited emphasis on computational reasoning
MATLAB	Numerical experimentation and modeling	Requires technical expertise
Python-based learning	Supports algorithmic thinking and computation	Steeper learning curve for beginners
Online learning platforms	Accessibility and flexibility	May encourage passive learning
Simulation-based activities	Increased engagement and exploration	Conceptual gains depend on instructional guidance

Mathematics education research has increasingly emphasized that meaningful learning requires students to engage with multiple mathematical representations, since symbolic manipulation alone is often insufficient; effective conceptual learning involves the interaction of symbolic, graphical, numerical, and contextual representations. In calculus, this means students should not only compute derivatives but also visualize function behavior, approximate rates of change numerically, interpret tangent slopes graphically, and apply derivatives to optimization and modeling (Sun et al., 2020). The growing availability of digital tools has created new opportunities for addressing these challenges: computational environments such as MATLAB, Python, and GeoGebra make it possible to visualize derivatives, approximate limits numerically, and explore local behavior dynamically, helping to transform abstract mathematical concepts into observable processes (Alam, 2020; Demetriou et al., 2019). Rather than seeing derivatives merely as formal algebraic expressions, students can observe how slopes emerge, how critical points develop, and how optimization behaves under changing parameters.

However, the integration of technology into mathematics teaching does not automatically lead to better learning. A recurring misconception in educational practice is the belief that simply introducing computers or digital tools will improve student understanding, yet evidence increasingly suggests that technology is effective only when it is embedded within pedagogically meaningful instructional design — in other words, computational tools matter, but teacher competence matters even more. In this study, teacher competence is conceptualized as a multidimensional construct involving four dimensions: (1) mathematical content knowledge, including an accurate understanding of calculus concepts; (2) pedagogical content knowledge, including the ability to explain concepts, address misconceptions, and select appropriate representations; (3) computational competence, including the ability to integrate digital tools and computational activities into mathematics instruction; and (4) instructional design competence, including the ability to organize learning activities that promote conceptual understanding (Ball et al., 2008; Mishra & Koehler, 2006; Shulman, 1986, 1987). A teacher who is computationally competent does not merely operate software, but understands how to use computational tools to support conceptual reasoning, mathematical inquiry, and reflective abstraction.

A substantial body of literature documents the conceptual difficulties students face in introductory calculus. Early work by Tall (1991) emphasized that learning calculus requires a transition from process-oriented symbolic manipulation to the conceptual objectification of mathematical ideas, and that many students remain trapped at the procedural level, unable to construct coherent conceptual images of derivatives and limits. Sfard (1991) similarly distinguished between operational and structural conceptions of mathematical knowledge, arguing that students often treat mathematical expressions as procedures rather than as conceptual objects — a distinction particularly relevant to differential calculus, where students may compute derivatives mechanically without understanding the derivative as a mathematical entity representing change. Hiebert and Lefevre (1986) highlighted the difference between procedural knowledge and conceptual knowledge, showing that deep mathematical learning requires rich interconnections among ideas rather than isolated algorithmic skills. In the context of derivatives specifically, Orton (1983), Weber et al. (2012), Zandieh (2000), and Thompson (1994) found that many students could apply differentiation formulas correctly while lacking understanding of tangent slope, rate of change, and graphical interpretation, and similar findings have been reported across educational contexts over several decades, suggesting that the conceptual difficulty of derivatives is structurally persistent rather than merely local.

With the growth of digital learning, scholars have increasingly investigated the role of technology in mathematics education. Kaput (1994) argued that computational technologies fundamentally change the representational possibilities of mathematical learning by enabling students to interact dynamically with mathematical objects, and Artigue (2002) emphasized that digital environments can support conceptual development by linking symbolic manipulation with graphical and numerical experimentation. More broadly, technology-rich environments appear to improve conceptual understanding only when carefully integrated into instructional design, and can easily become superficial if used merely as a presentation tool rather than as a cognitive instrument. A variety of technology-assisted approaches have been employed to support calculus instruction, each with characteristic strengths and limitations, as summarized in **Table 1**: dynamic visualization environments such as GeoGebra improve visual intuition but emphasize graphical exploration without necessarily developing computational reasoning; computational platforms such as MATLAB and Python support numerical experimentation and algorithmic problem-solving but can impose substantial cognitive demands on novice learners; and online learning platforms increase accessibility and flexibility but do not necessarily promote deep conceptual understanding unless supported by effective pedagogical design.

Despite this growing body of research, several important limitations remain. First, many studies examine digital tools, visualization software, or online learning environments in isolation, without systematically comparing them with traditional and conceptually focused teaching under comparable instructional conditions. Second, although teacher competence is widely recognized as critical to student learning, relatively few studies investigate how teachers' computational thinking competence mediates the effectiveness of technology-supported calculus instruction (Rybak, 2021). Third, much available research reports improvements in engagement or technology use while providing limited evidence regarding conceptual understanding, retention, transfer, and reasoning. Fourth, empirical investigations conducted in developing educational systems remain comparatively scarce despite substantial differences in teacher preparation, technological resources, and classroom practice (UNESCO, 2020;

World Bank, 2018). Finally, there is limited use of quantitative, decision-oriented evaluation frameworks capable of comparing multiple instructional models simultaneously and identifying which factors contribute most to conceptual learning outcomes. These gaps motivate the present study.

The central problem addressed in this paper is therefore: how can first-year calculus instruction be designed so that students move beyond symbolic manipulation toward genuine conceptual understanding, and what role do computational thinking and teacher competence play in achieving this transition? This paper makes several contributions: it provides a systematic comparative evaluation of three instructional models — traditional teaching, conceptual teaching, and computationally enhanced conceptual teaching — focused specifically on derivatives and optimization; it develops an instructional framework integrating symbolic reasoning, graphical interpretation, numerical approximation, and computational experimentation; it provides quantitative experimental evidence through pre-test, post-test, retention, and transfer-task analysis; and it examines teacher competence as a critical mediating factor rather than treating computational tools as a self-sufficient intervention. Its novelty lies not simply in using digital tools in calculus teaching, but in the integrated comparative framework through which computational thinking, teacher competence, and instructional design are studied together — a perspective especially relevant to developing educational systems, where the pedagogical value of technology depends critically on teacher preparation and instructional coherence. The proposed framework is intended to be adaptable across institutions, though large-scale implementation would require further validation across different populations and technological contexts.

The remainder of this paper is organized as follows. Section *Theoretical Framework* presents the theoretical framework underlying conceptual learning, computational thinking, and teacher competence in calculus education. Section *Method* describes the research method, including participants, instructional content, classroom implementation, and assessment instruments. Section *Results* reports the statistical results, including descriptive statistics, ANOVA, post-hoc comparisons, and effect-size estimates. Section *Discussion* discusses the results in relation to existing literature and their implications for calculus teaching, and Section *Conclusion, Limitations, and Future Research* concludes the paper by summarizing the principal findings, limitations, and directions for future research.

THEORETICAL FRAMEWORK

The present study is situated at the intersection of three complementary perspectives in mathematics education: conceptual understanding in calculus, computational thinking as a mode of mathematical reasoning, and teacher competence as the mediating condition of effective instructional design (Blömeke et al., 2022; Chiu et al., 2024; Yang & Du, 2024). These dimensions are not independent; rather, they interact in ways that shape how students encounter abstract mathematical ideas, how they construct meaning from formal procedures, and how classroom instruction supports or constrains conceptual development (Salloum & BouJaoude, 2019). In introductory calculus, the derivative is often the first concept requiring students to coordinate symbolic expressions, graphical interpretation, numerical approximation, and contextual reasoning simultaneously, which makes it an especially appropriate setting for examining the interplay among these three dimensions.

Conceptual Understanding and Computational Thinking in Calculus

The derivative is more than a symbolic operation: it represents rate of change, local linear approximation, tangent slope, sensitivity, and dynamic variation, and understanding it conceptually allows students to interpret mathematical relationships rather than merely manipulate formulas. Tall (1991) described how many students remain at a procedural level, unable to form coherent conceptual images of derivatives and limits, while Sfard (1991) distinguished operational conceptions — in which mathematical expressions are treated as step-by-step procedures — from structural conceptions, in which they are treated as coherent mathematical objects; students frequently begin with operational conceptions and only gradually, if at all, develop structural ones. Hiebert and Lefevre (1986) argued that conceptual knowledge consists of rich, interconnected relationships among ideas rather than isolated algorithms, and Orton (1983), Zandieh (2000), and Thompson (1994) documented that students frequently compute derivatives correctly while lacking understanding of tangent slope, rate of change, and the coordination of the ratio, limit, and function layers that underlie the formal definition of the derivative.

Computational thinking, understood as the mental processes involved in formulating problems so that a computer or numerical procedure can help solve them (Wing, 2006), offers one route toward such conceptual coordination. In a calculus context, computational activity can create a bridge between intuition and formalism: for example, before proving that $d/dx(x^2) = 2x$, students may first numerically approximate derivatives for small step sizes and observe a stable converging pattern, which helps them form conceptual expectations that later support formal reasoning (Artigue, 2002). Kaput (1994) similarly argued that computational environments transform mathematical objects into interactive systems that learners can perceive structurally through experimentation, and Voogt et al. (2015) argue more broadly that computational thinking supports abstraction, algorithmic reasoning, and pattern generalization when it is integrated into subject-matter learning rather than taught as an isolated skill.

It is important, however, not to idealize computation. The literature makes clear that computational tools do not automatically produce understanding: students may use software mechanically without engaging mathematically, and visualizations may remain superficial unless learners are explicitly guided toward interpretation (Voogt et al., 2015). For this reason, computational thinking is pedagogically meaningful only when computational activities are embedded within coherent instructional design — a point that leads directly to the role of the teacher.

Teacher Competence as a Mediating Factor

The educational value of computational thinking depends fundamentally on the competence of the teacher. Teacher competence in mathematics education extends well beyond subject knowledge: according to Shulman (1986, 1987), effective teaching requires pedagogical content knowledge — the ability to transform disciplinary knowledge into forms that are understandable to learners — and in calculus teaching this transformation is especially demanding because many concepts are simultaneously abstract, dynamic, and representationally complex. Ball et al. (2008) extended this idea into mathematical knowledge for teaching, distinguishing the common content knowledge shared with non-teachers from the specialized content knowledge unique to the work of teaching mathematics.

For the purposes of the present study, teacher competence is understood as comprising four interconnected dimensions. The first is conceptual mathematical knowledge: teachers must possess a deep understanding of how derivatives relate to rate of change, geometric behavior, optimization, local approximation, and modeling, since a teacher who lacks conceptual depth may unintentionally reduce instruction to symbolic rule delivery. The second is pedagogical sequencing, or knowing how to organize mathematical ideas progressively — moving from intuition to representation, from examples to abstraction, and from graphical insight to symbolic generalization — because poor sequencing can fragment learning even when the mathematical content presented is correct. The third is representational flexibility: a competent teacher can move fluently among symbolic, graphical, numerical, and verbal representations, which is crucial because students often understand one representation while struggling with another, and a teacher who can connect these forms helps students build relational rather than isolated understanding (Duval, 2006). The fourth is computational literacy: computationally competent teachers understand not merely how to operate software, but how to design tasks in which computation supports conjecture, exploration, comparison, and conceptual explanation (Mishra & Koehler, 2006).

A key theoretical assumption of this study is that technology itself does not produce learning gains; rather, teacher competence mediates the relationship between computational tools and student understanding. The same software may function very differently depending on how it is used — in one classroom, graphing software may merely replace board drawings, while in another it may become a tool for genuine conceptual inquiry — so the effectiveness of computational teaching depends primarily on the teacher's capacity to integrate it meaningfully rather than on the technology itself. Building on this reasoning, the present study is based on the theoretical proposition that conceptual understanding in calculus develops most effectively when students engage with multiple mathematical representations through computationally supported inquiry, and when that inquiry is guided by pedagogically competent teaching. This proposition motivated the comparison of three instructional models: traditional procedural instruction, emphasizing symbolic rules and routine exercises; conceptual instruction, emphasizing graphical meaning and interpretive reasoning; and computationally enhanced conceptual instruction, integrating symbolic, graphical, numerical, and computational exploration. The theoretical expectation is that the third model will produce the strongest outcomes — not simply because of technology, but because of the interaction among conceptual representation, computational thinking, and teacher competence.

METHOD

This section describes the empirical design of the study, the participants and instructional setting, the intervention procedures, the assessment instruments, and the statistical methods used to evaluate the comparative effectiveness of the three instructional approaches. The methodological design was guided directly by the theoretical framework developed above: in particular, the study was designed to examine whether differences in instructional structure — traditional procedural teaching, conceptual teaching, and computationally enhanced conceptual teaching — produce measurable differences in students' conceptual understanding, computational reasoning, transfer ability, and retention.

Participants and Research Design

The study adopted a quasi-experimental pre-test–post-test comparative design with three instructional groups, illustrated in **Figure 1**. A quasi-experimental design was considered appropriate because the study was conducted within an authentic university classroom environment where full randomization of all educational variables was not feasible; comparability across groups was instead supported through equivalent baseline testing, standardized instructional duration, common learning objectives, and uniform assessment procedures. The design included three instructional conditions of equal size — G1 (traditional procedural instruction), G2 (structured conceptual instruction), and G3 (computationally enhanced conceptual instruction) — all of which studied the same calculus topic over the same instructional period with equivalent classroom contact time.

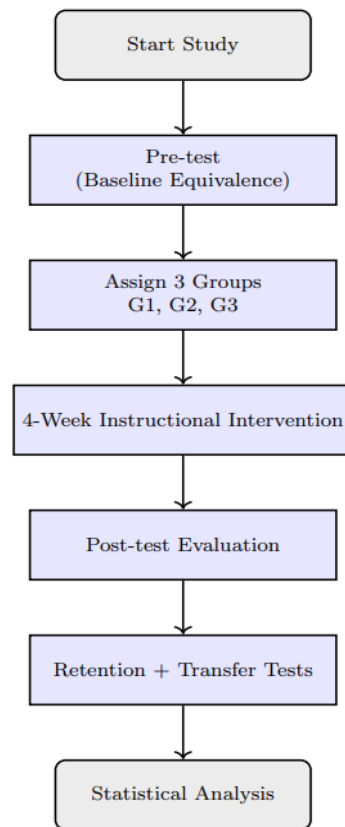


Figure 1. Overall experimental research design pipeline of the study (Source: Authors' own elaboration)

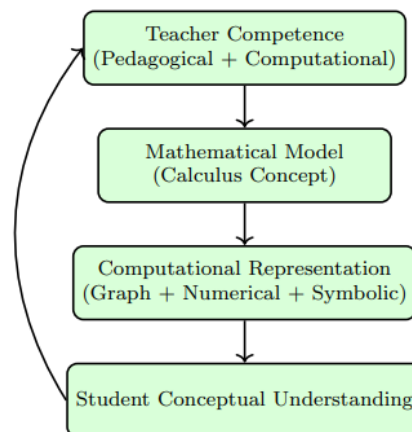


Figure 2. Computational thinking-mediated instructional cycle in calculus learning (Source: Authors' own elaboration)

To control for instructor-related effects, all three groups were taught by the same instructor throughout the intervention, following a standardized instructional plan, common learning objectives, and equivalent instructional schedules. Teacher competence was therefore treated as a controlled instructional condition rather than an independently measured variable: the instructor was required to demonstrate competence in calculus content knowledge, conceptual explanation, classroom facilitation, and computational tool integration appropriate to each group's instructional requirements (illustrated conceptually in **Figure 2**), so that differences in student outcomes could be interpreted primarily in relation to instructional design rather than instructor variability.

The study employed four assessment stages — a pre-test administered before the instructional intervention, a post-test administered immediately after it, a delayed retention test administered two weeks later, and a transfer task designed to assess the application of concepts to unfamiliar optimization problems — allowing the study to evaluate not only immediate performance gains but also conceptual durability and transferability of learning. The four-week intervention was aligned with the regular first-year calculus curriculum on derivatives and optimization; this duration was sufficient to evaluate the immediate

Table 2. Number of groups and participants

Group	Instructional model	Number of students
G1	Traditional procedural teaching	48
G2	Structured conceptual teaching	48
G3	Computationally enhanced conceptual teaching	48

Table 3. Pre-test descriptive statistics by group

Group	n	Mean	Minimum	Maximum
G1 Traditional	48	37.8	22	54
G2 Conceptual	48	38.5	23	55
G3 Computational	48	38.2	21	56

Table 4. One-way ANOVA for pre-test scores, confirming baseline equivalence

Source	SS	df	MS	F	p
Between groups	11.26	2	5.63	0.09	.918
Within groups	9205.74	141	65.29	—	—
Total	9217.00	143	—	—	—

Table 5. Common weekly content structure followed by all three instructional groups

Week	Main topic	Core mathematical ideas
1	Introduction to derivatives	Average rate of change, secant slope, derivative from limits
2	Derivative interpretation	Tangent slope, local linearity, graphical meaning
3	Monotonicity and critical points	Increasing/decreasing behavior, sign of the derivative
4	Optimization	Maxima, minima, and applied optimization problems

effects of the instructional approaches, so the findings should be interpreted as evidence of short-term instructional effectiveness rather than long-term conceptual development.

A total of 144 first-year undergraduate engineering students, enrolled in the same introductory calculus course and with comparable secondary-school mathematics backgrounds, were distributed into three intact groups of 48 students each (Table 2). Because the study focused on early calculus learning, first-year students were an especially relevant population, since most encounter formal derivative concepts for the first time in a university-level setting.

Before the intervention, all students completed a pre-test measuring prior knowledge of algebraic manipulation, function interpretation, elementary graph reading, and basic rate-of-change intuition, to verify that the three groups were comparable at baseline. As Table 3 shows, mean pre-test scores were closely matched across groups (G1: $M = 37.8$, $SD = 8.1$; G2: $M = 38.5$, $SD = 7.9$; G3: $M = 38.2$, $SD = 8.3$), and a one-way ANOVA confirmed no statistically significant difference among groups before the intervention, $F(2, 141) = 0.09$, $p = .918$ (Table 4). Because prior knowledge was equivalent and all groups subsequently followed the same syllabus, instructional time, and assessment procedures, differences in post-intervention outcomes are unlikely to be explained by initial ability, content exposure, time-on-task, or instructor variability; classroom interaction naturally differed across groups, but this reflects the intended pedagogical comparison rather than an uncontrolled confound.

All three groups followed an identical content trajectory over four weeks, summarized in Table 5, so that observed differences could be attributed to instructional design rather than to differences in topic coverage. Each group met three times per week for 90 minutes, for a total of 18 instructional hours; maintaining equal instructional time across groups was intended to rule out a simple time-on-task explanation for any differences in outcomes.

Within this shared content plan, the instructor's and students' roles were deliberately differentiated by group, as summarized in Table 6. Across all groups, the instructor was responsible for following the standardized weekly lesson plan and timing, opening each session with the same learning objectives, and administering the common assessments; what varied was how each session was facilitated. In G1, the instructor's role was to present differentiation rules and worked examples and to assign practice exercises, following the sequence definition $\rightarrow$ rule explanation $\rightarrow$ example $\rightarrow$ exercise, while students' role was to reproduce worked procedures with minimal discussion of graphical or numerical meaning. In G2, the instructor's role shifted to posing interpretive questions about graphs (for example, where the graph increases, where the tangent is horizontal, and what a positive derivative means) before introducing symbolic notation, and to facilitating classroom discussion, while students were expected to articulate reasoning about function behavior before formal computation. In G3, the instructor additionally designed and demonstrated the computational exploration tasks — Python-based plotting and numerical approximation activities — circulated to support students' interpretation of computational output, and explicitly connected computational observations back to symbolic results, while students' role expanded to include running and interpreting numerical approximations, testing conjectures computationally, and explaining the correspondence between computational and symbolic representations. Table 7 and Table 8 summarize the resulting instructional characteristics of G1 and G2 in more detail, and Table 9 details the five-stage pedagogical sequence used in G3.

Table 6. Roles and responsibilities of instructor and students, and instructional schedule, by group

Dimension	G1 — Traditional	G2 — Conceptual	G3 — Computational
Instructor role	Present rule, model worked example, assign exercises	Pose interpretive/graphical questions, then formalize	As G2, plus design and guide computational tasks
Student role	Reproduce procedures on practice items	Interpret graphs, articulate reasoning, then compute	Run/interpret numerical approximations, test conjectures, link to symbolic form
Weekly contact time	3 × 90 min	3 × 90 min	3 × 90 min
Total duration	4 weeks (18 h)	4 weeks (18 h)	4 weeks (18 h)

Table 7. Instructional characteristics of Group G1 (traditional procedural instruction)

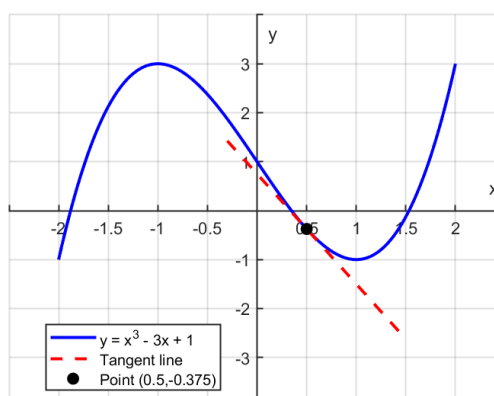
Dimension	Description
Teaching style	Lecture-centered
Representation	Primarily symbolic
Student activity	Rule application
Use of graphs	Minimal
Use of computation	None
Teacher questioning	Mostly procedural

Table 8. Instructional characteristics of Group G2 (structured conceptual instruction)

Dimension	Description
Teaching style	Guided conceptual instruction
Representation	Symbolic + graphical
Student activity	Interpretation and explanation
Use of graphs	Frequent
Use of computation	None
Teacher questioning	Conceptual and interpretive

Table 9. Five-stage pedagogical sequence used in Group G3 (computationally enhanced conceptual instruction)

Stage	Instructional activity	Learning purpose
1	Concept introduction	Establish intuitive meaning
2	Graphical exploration	Visualize behavior
3	Symbolic derivation	Formalize structure
4	Numerical approximation	Observe limiting behavior
5	Computational experimentation	Investigate dynamically

**Figure 3.** Conceptual interpretation of the derivative as tangent slope (Group G2) (Source: Authors' own elaboration)

A typical G1 lesson followed explanation → worked example → student exercise: the instructor presented a rule such as $d/dx(x^n) = nx^{n-1}$, worked several examples, and students then solved similar exercises — for instance, differentiating $f(x) = x^3 - 3x + 1$ to obtain $f'(x) = 3x^2 - 3$ — without extended discussion of what the result revealed about the function. In G2, students first examined the graph of the same function (Figure 3) and identified the approximate turning points and the intervals of increase and decrease before computing the derivative symbolically.

In G3, students additionally approximated the derivative numerically using $f'(x) \approx [f(x+h) - f(x)] / h$ for progressively smaller values of h , as shown in Table 10, observing how the approximate slope near $x = 1$ converged toward zero — that is, $f'(1) = 0$ — through numerical convergence (Figure 4) rather than immediate symbolic computation, before connecting this behavior to the critical points of $f'(x) = 3x^2 - 3$ at $x = \pm 1$ (Figure 5) and to the conceptual-computational correspondence summarized in Table 11.

Table 10. Numerical approximation of $f'(1)$ for $f(x) = x^3 - 3x + 1$ as h decreases (Group G3)

h	Approximate slope near $x = 1$
1.0	4.000
0.5	1.750
0.1	0.310
0.01	0.0301
0.001	0.003001

Table 11. Conceptual–computational mapping used in Group G3

Observed behavior	Symbolic interpretation
Graph increasing	$f'(x) > 0$
Graph decreasing	$f'(x) < 0$
Horizontal tangent	$f'(x) = 0$
Local maximum	sign changes $+\rightarrow -$
Local minimum	sign changes $-\rightarrow +$

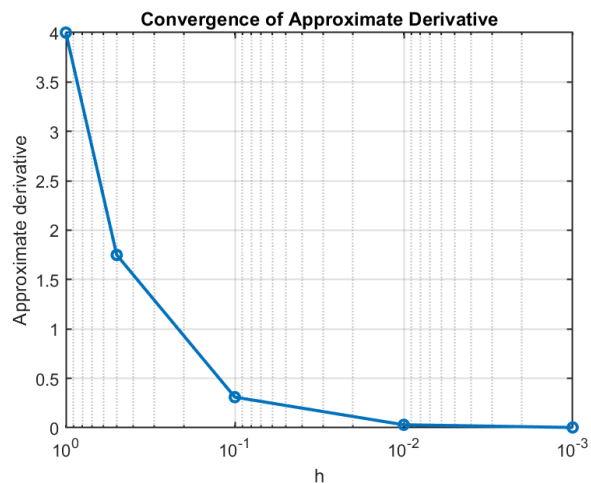
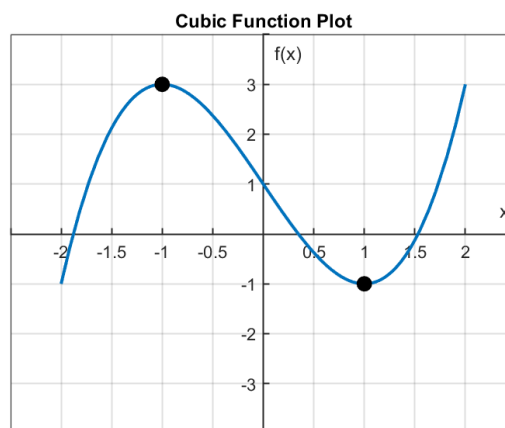
**Figure 4.** Numerical convergence of the derivative approximation (Source: Authors' own elaboration)**Figure 5.** Function behavior and critical points explored in Group G3 (Source: Authors' own elaboration)

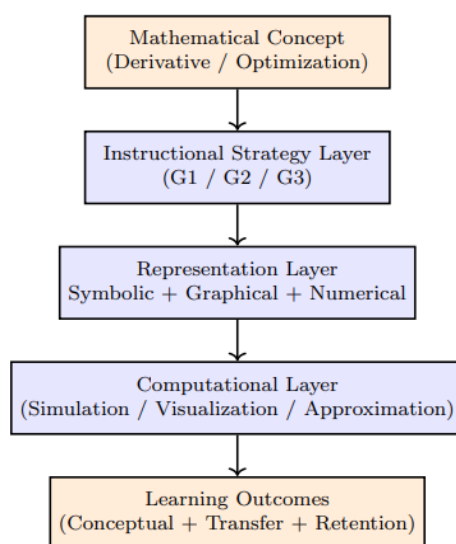
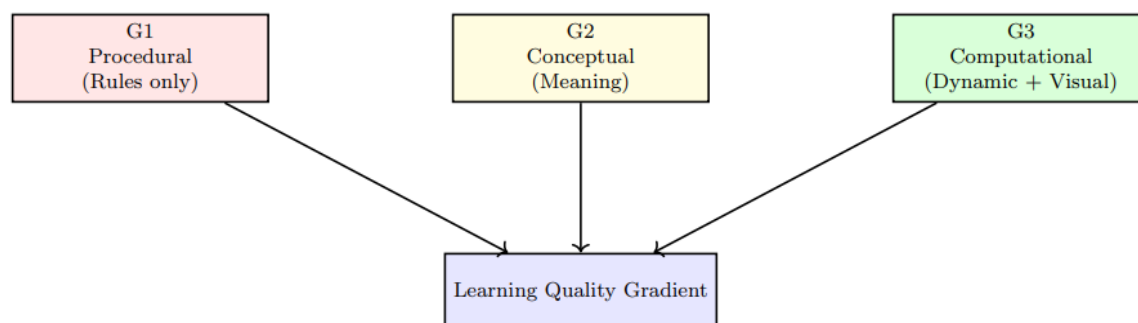
Table 12 summarizes the resulting qualitative instructional profile across all three groups. All groups received substantial exposure to symbolic differentiation, but only G2 and G3 incorporated graphical interpretation consistently, and only G3 provided numerical approximation, dynamic exploration, and a high degree of student inquiry and cross-representational integration — a pattern consistent with the classroom observations summarized in **Table 13**, which, although not used as formal statistical variables, help contextualize the quantitative findings reported in Section *Results*.

Table 12. Comparative instructional profile across the three groups

Feature	G1	G2	G3
Symbolic differentiation	High	High	High
Graphical interpretation	Low	High	High
Numerical approximation	None	Minimal	High
Dynamic exploration	None	None	High
Student inquiry	Low	Moderate	High
Representation integration	Low	Moderate	High

Table 13. Observed classroom behavior across the three instructional groups

Observation	G1	G2	G3
Passive note-taking	High	Moderate	Low
Student questioning	Low	Moderate	High
Graph-based reasoning	Low	High	High
Exploratory discussion	Low	Moderate	High
Computational experimentation	None	None	High

**Figure 6.** Instructional system architecture integrating mathematical, pedagogical, and computational layers (Source: Authors' own elaboration)**Figure 7.** Integration of mathematical content, pedagogical strategy, and computational tools within the unified instructional framework (Source: Authors' own elaboration)

Intervention and Data Collection

Four study-specific instruments were used to assess students' mathematical readiness, learning outcomes, retention, and ability to transfer derivative concepts to unfamiliar problems. The instruments were developed by the research team in accordance with the instructional objectives (Figure 6), the conceptual-understanding framework (Figure 7), and standard first-year calculus learning outcomes. A summary of the instruments, their item structure, and administration schedule is presented in Table 14 (full instruments provided in Appendix Table A1). The learning dimensions addressed across the instruments are summarized in Table 15.

Table 14. Data collection instruments, item structure, and timing of administration

Instrument	Purpose / content	Developer	Items	Timing/administration
Pre-test	Baseline mathematical readiness, including algebraic manipulation, graph interpretation, rate-of-change reasoning, and prerequisite differentiation concepts	Research team	15 (Appendix Table A1)	Immediately before the four-week intervention; all groups
Post-test	Procedural fluency, conceptual understanding of derivatives, graphical interpretation, optimization reasoning, and computational reasoning	Research team	20 (Appendix Table A2)	Immediately after the four-week intervention; all groups
Delayed retention test	Retention of derivative meaning, graphical reasoning, and optimization interpretation using a parallel form	Research team	15 (Appendix Table A3)	Two weeks after the post-test; all groups
Transfer task	Application of derivative concepts to unfamiliar optimization problems requiring mathematical modelling and justification	Research team	4 (Appendix Table A4)	Same session as the delayed retention test; all groups

Table 15. Learning dimensions assessed across instruments

Learning dimension	Description
Conceptual understanding	Meaning of the derivative, tangent slope, instantaneous rate of change, and relationships between derivative sign and function behavior
Procedural fluency	Correct application of symbolic differentiation procedures and derivative rules
Transfer ability	Application of derivative concepts to unfamiliar and non-routine problems
Computational reasoning	Interpretation of numerical, graphical, and computational representations of mathematical behavior
Engagement	Participation, task involvement, and active engagement during the instructional intervention

The pre-test consisted of 15 items designed to establish students' baseline mathematical readiness before the intervention. The items assessed algebraic manipulation, interpretation of simple graphs, basic rate-of-change reasoning, and prerequisite differentiation concepts. The pre-test was administered to all three groups in a single session immediately before the four-week intervention. Its primary purpose was to characterize baseline readiness and support the interpretation of post-intervention differences.

The post-test, which served as the primary outcome measure, consisted of 20 items covering procedural fluency, conceptual understanding of derivatives, graphical interpretation, optimization reasoning, and computational reasoning. The assessment included both routine and non-routine items. Conceptual items required students to interpret the meaning of the derivative, relate tangent slope to instantaneous rate of change, connect the sign of the derivative with the behavior of a function, and justify optimization decisions. Computational-reasoning items focused on interpreting numerical and graphical outputs, estimating rates of change, comparing numerical results, and identifying patterns from computational representations rather than assessing programming proficiency. The post-test was administered to all groups immediately after the intervention.

The delayed retention test consisted of 15 parallel-form items and was administered two weeks after the post-test. It assessed students' retention of derivative concepts, graphical reasoning, and optimization interpretation. The use of a parallel form reduced the possibility that performance reflected direct recall of the post-test items. The transfer task consisted of 4 unfamiliar optimization problems administered during the same session as the delayed retention test. The problems were deliberately designed to differ from routine classroom examples and required students to identify relevant variables, formulate an appropriate mathematical model, apply derivative-based reasoning, and justify their conclusions. The transfer task therefore assessed whether students could apply conceptual knowledge in a novel context rather than merely reproduce previously practiced procedures.

All instruments were reviewed for content relevance and clarity by two experienced university mathematics educators before administration. The items were subsequently refined to improve wording and alignment with the intended learning dimensions. The post-test demonstrated acceptable internal consistency, with Cronbach's alpha of approximately $\alpha = 0.82$.

Statistical Analysis

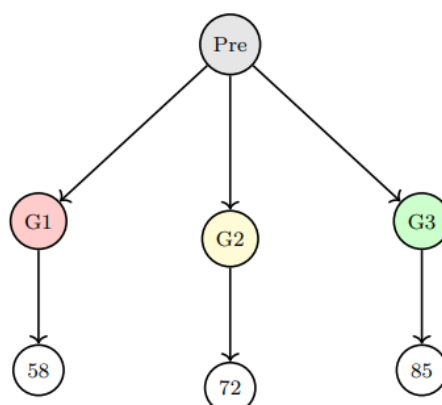
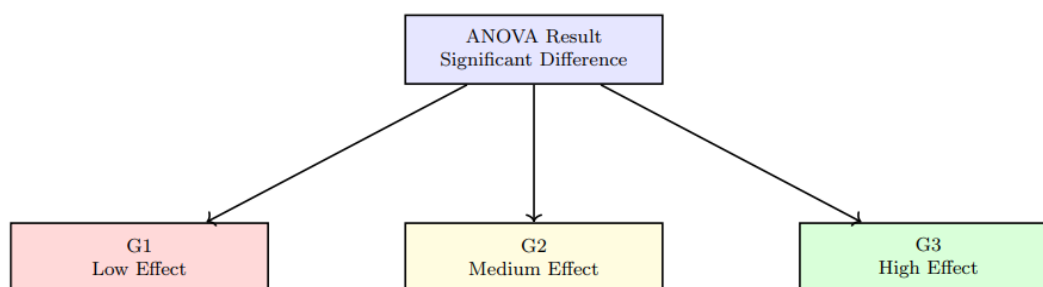
For each outcome, descriptive statistics (means, standard deviations, and score ranges) were computed by group. One-way ANOVA was used to test for statistically significant differences among the three groups on post-test, retention, and transfer-task scores, since the design compared the means of three independent groups; where ANOVA indicated a significant effect, Tukey's post-hoc test identified which pairwise group comparisons were significant. Cohen's d was computed for key pairwise comparisons to assess the educational, and not merely statistical, magnitude of any differences. Although the quasi-experimental design controlled several potential confounds — prior mathematical knowledge, instructional duration, course content, assessment conditions, and instructor identity — classroom interaction patterns and engagement levels may naturally vary across instructional approaches because they are inherent characteristics of the pedagogical models under comparison; randomized designs with larger, multi-institutional samples would strengthen causal inference in future work.

RESULTS

This section presents the empirical findings of the study, organized into baseline equivalence (already reported in Section *Participants and Research Design*), post-intervention descriptive statistics, inferential analysis using one-way ANOVA, post-hoc pairwise comparisons, and effect-size interpretation together with the multidimensional learning comparison. The analysis

Table 16. Post-test descriptive statistics by group

Group	n	Mean	SD	Minimum	Maximum
G1 Traditional	48	58.4	10.7	36	78
G2 Conceptual	48	71.8	9.4	50	88
G3 Computational	48	84.6	8.8	63	97

**Figure 8.** Learning trajectory model from pre-test to post-test across instructional groups (Source: Authors' own elaboration)**Figure 9.** Decision-strength interpretation of ANOVA results across instructional groups (Source: Authors' own elaboration)**Table 17.** Delayed retention test descriptive statistics (two weeks after the post-test)

Group	Mean	SD
G1 Traditional	51.3	11.2
G2 Conceptual	66.4	9.6
G3 Computational	80.1	8.9

focuses on four outcome variables: post-test performance, delayed retention, transfer-task performance, and the multidimensional learning indicators.

After the four-week intervention, all students completed the post-test; as **Table 16** shows, a clear progression was visible across groups (G1: $M = 58.4$, $SD = 10.7$; G2: $M = 71.8$, $SD = 9.4$; G3: $M = 84.6$, $SD = 8.8$), illustrated by the learning-trajectory pattern from pre-test to post-test in **Figure 8** and, in decision-strength terms, in **Figure 9**.

Two weeks after the post-test, students completed the delayed retention assessment (**Table 17**). The computationally enhanced group again achieved the highest mean retention score and the lowest variability among the groups, reflecting both stronger retention and greater consistency of learning outcomes, while the traditional group obtained the lowest mean retention score, suggesting that procedurally focused teaching supported only limited long-term conceptual preservation (**Figure 10**).

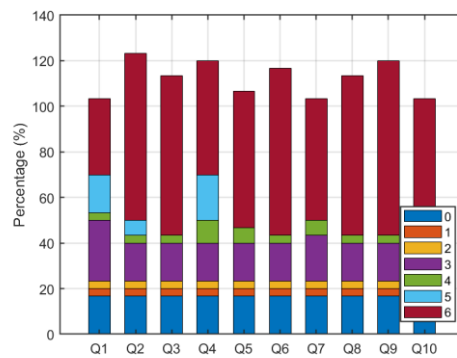


Figure 10. Comparison of delayed retention performance among instructional groups (Source: Authors' own elaboration)

Table 18. Transfer-task descriptive statistics

Group	Mean	SD
G1 Traditional	42.6	12.1
G2 Conceptual	63.9	10.2
G3 Computational	81.5	8.6

Table 19. One-way ANOVA for post-test scores

Source	SS	df	MS	F	p
Between groups	16542.8	2	8271.4	19.42	< .001
Within groups	60042.1	141	425.8	—	—
Total	76584.9	143	—	—	—

Table 20. Tukey post-hoc comparisons of post-test scores

Comparison	Mean difference	95% CI	p
G1 vs. G2	13.4	[7.2, 19.6]	< .01
G1 vs. G3	26.2	[19.8, 32.5]	< .001
G2 vs. G3	12.8	[6.7, 18.9]	< .01

Table 21. Effect-size estimates for pairwise group comparisons

Comparison	Cohen's d	Interpretation
G1 vs. G2	1.34	Large
G1 vs. G3	2.67	Very large
G2 vs. G3	1.42	Large

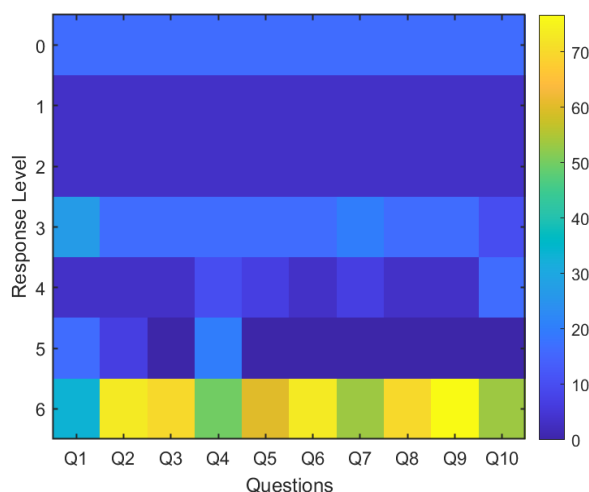
The transfer task, which required students to apply derivative concepts to an unfamiliar optimization problem, produced the largest instructional differences of any measure (**Table 18**): the computational group ($M = 81.5$, $SD = 8.6$) substantially outperformed the conceptual ($M = 63.9$, $SD = 10.2$) and traditional ($M = 42.6$, $SD = 12.1$) groups, indicating that computationally enhanced instruction promoted stronger conceptual generalization and improved application of derivative concepts in unfamiliar settings.

A one-way ANOVA on post-test scores confirmed a statistically significant difference among groups, $F(2, 141) = 19.42$, $p < .001$ (**Table 19**), indicating that instructional design had a significant effect on students' post-test performance. Tukey post-hoc comparisons (**Table 20**) showed that every pairwise difference was significant: G1 vs. G2, $M_{diff} = 13.4$, 95% CI [7.2, 19.6], $p < .01$; G1 vs. G3, $M_{diff} = 26.2$, 95% CI [19.8, 32.5], $p < .001$; and G2 vs. G3, $M_{diff} = 12.8$, 95% CI [6.7, 18.9], $p < .01$. Effect sizes were large throughout (**Table 21**): Cohen's $d = 1.34$ for G1 vs. G2, $d = 2.67$ for G1 vs. G3, and $d = 1.42$ for G2 vs. G3, indicating that the differences among groups were educationally, and not only statistically, substantial. Parallel patterns in the delayed retention and transfer-task scores reported above showed the same ordering ($G3 > G2 > G1$), indicating that the computationally enhanced group's advantage was not limited to immediate post-test performance but extended to the durability of learning and application to unfamiliar problems.

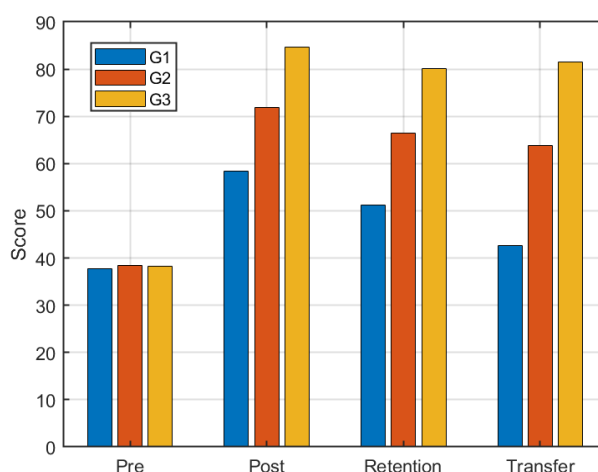
Table 22 reports outcomes across the five learning dimensions defined in **Table 15**. The computationally enhanced group scored highest on every dimension, with the largest between-group differences appearing in computational reasoning (18 vs. 35 vs. 89 for G1, G2, and G3, respectively), transfer ability (43 vs. 64 vs. 81), and conceptual understanding (58 vs. 73 vs. 86), while procedural fluency differed less sharply (69 vs. 74 vs. 82). This pattern, shown in **Figure 11**, suggests that computationally enriched instruction improved not only routine performance but also reasoning capacity and the ability to apply calculus knowledge in unfamiliar situations, and is consistent with the independent classroom observations reported in **Table 13**.

Table 22. Multidimensional learning profile by instructional group

Learning dimension	G1	G2	G3
Conceptual understanding	58	73	86
Procedural fluency	69	74	82
Transfer ability	43	64	81
Computational reasoning	18	35	89
Engagement	47	61	79

**Figure 11.** Multidimensional learning profile of the three instructional groups (Source: Authors' own elaboration)**Table 23.** Gain-score comparison from pre-test to post-test by group

Group	Pre-test M	Post-test M	Gain
G1 Traditional	37.8	58.4	20.6
G2 Conceptual	38.5	71.8	33.3
G3 Computational	38.2	84.6	46.4

**Figure 12.** Pre-test, post-test, retention, and transfer performance across the three instructional groups (Source: Authors' own elaboration)

Finally, learning gains from pre-test to post-test provide a further indicator of instructional effectiveness (**Table 23; Figure 12**). The computational group demonstrated by far the strongest gain (46.4 points), compared with 33.3 points for the conceptual group and 20.6 points for the traditional group, further confirming the advantage of computationally enhanced conceptual instruction for supporting meaningful calculus learning.

DISCUSSION

This section interprets the empirical findings in relation to the theoretical framework and existing literature, focusing on why the observed differences among instructional groups occurred, what they imply for computational thinking in mathematics

education, and how teacher competence mediates learning outcomes in both traditional and technology-enhanced environments. The results consistently demonstrate that instructional design significantly influences students' conceptual development in first-year calculus: computationally enhanced conceptual instruction produced the strongest outcomes across every measured dimension — conceptual understanding, procedural fluency, retention, transfer ability, and computational reasoning — but interpreting why requires moving beyond the statistical comparison itself.

The clearest empirical finding is a consistent hierarchy — computationally enhanced instruction > conceptual instruction > traditional instruction — across post-test performance, retention, and transfer (**Tables 16–23; Figures 8–12**). Traditional instruction supported procedural acquisition but was insufficient for conceptual understanding: consistent with Hiebert and Lefevre (1986), students in G1 could execute algorithms correctly (**Table 7**) yet performed poorly on transfer tasks requiring interpretation (**Table 18**), a pattern that can be described as procedural isolation, in which correct execution of algorithms does not translate into conceptual comprehension. From a cognitive perspective, traditional instruction placed heavy emphasis on symbolic manipulation but offered limited opportunity to explore mathematical structure, visualize dynamic behavior, connect representations, or interpret mathematical meaning. This pattern echoes recent evidence that first-year university students struggle specifically with the conceptual and procedural demands of introductory calculus, including difficulty connecting symbolic manipulation to underlying meaning (Kurudirek et al., 2025); the present findings are consistent with that work in identifying procedural weakness as a persistent barrier, but extend it by showing, through a direct three-group comparison, that this barrier can be substantially reduced when procedural instruction is replaced with conceptual and computationally enhanced approaches rather than treated as an inevitable feature of first-year learning.

Conceptual instruction (G2) improved interpretive reasoning through graphical interpretation, function-behavior reasoning, and guided inquiry (**Table 8**), consistent with Sfard's (1991) structural conception of mathematical understanding, and produced significantly better outcomes than G1, particularly in conceptual understanding and retention. However, despite these improvements, students in G2 relied mainly on static graphs and teacher-led interpretation and did not reach the performance level of G3, suggesting that conceptual instruction alone — while valuable — may still be limited when it does not provide dynamic interaction with mathematical objects or opportunities to test conjectures experimentally. This distinction between G2 and G3 is confirmed statistically by the ANOVA and Tukey results in **Tables 19** and **20**. This outcome aligns with Rodríguez-Muñiz et al. (2022), who found that broader, relational conceptions of mathematics promote deeper conceptual engagement among undergraduates. Our G2 results support this conclusion, while the superior performance of G3 further demonstrates that dynamic computational engagement can extend conceptual learning beyond what conceptual instruction alone achieves.

Computationally enhanced instruction (G3) outperformed both other groups across every measure, including transfer and retention (**Tables 16–18, 22, 23**). Three mechanisms plausibly account for this advantage. First, computational tools allowed students to observe derivatives as dynamic processes rather than static formulas — for example, the numerical approximation of derivatives in **Table 10** and **Figure 4** let students see how secant slopes converge to tangent slopes — which can support the transition from intuitive reasoning to formal abstraction (Kaput, 1994). Second, computational instruction integrated symbolic, graphical, numerical, and computational representations simultaneously (**Table 9**), consistent with Duval's (2006) account of how conceptual understanding emerges from coordinating multiple semiotic representations; students in G3 were continuously required to move between representations, which is reflected in their broader multidimensional profile (**Table 22, Figure 11**). Third, computational environments shifted students from passive recipients of rules to active investigators asking, for instance, what happens to the slope as h becomes smaller, or why the derivative changes sign — consistent with Schoenfeld's (1992) view of mathematical learning as reasoning, exploration, and sense-making rather than memorization. The higher engagement levels observed in G3 (**Table 13, Table 22**) likely contributed to deeper cognitive processing and, in turn, to the stronger retention and transfer performance evident in **Tables 17** and **18**. This interpretation is broadly consistent with a recent bibliometric analysis of computational-thinking research in mathematics education, which identifies a marked increase since 2020 in studies examining computational thinking through the lenses of representation and problem solving, rather than treating it primarily as a programming-related skill (Irawan et al., 2024). The present study aligns with that trend by treating computational thinking as an internal reasoning process rather than an external tool, but differs from much of the reviewed literature, which remains largely conceptual or bibliometric in nature, by providing direct experimental evidence — through a controlled three-group comparison with retention and transfer measures — of the learning gains such computational thinking can produce when embedded in competently designed calculus instruction.

A central theoretical claim of this study is that computational tools alone do not guarantee improved learning outcomes, and the results support this claim: the computational condition succeeded specifically as an extension of the same conceptually structured teaching used in G2 (**Table 6**), not as a technology substitute for teaching. The success of computational instruction therefore depended critically on teacher competence — particularly in designing meaningful computational tasks, guiding interpretation of visual and numerical output, connecting computational results with formal mathematics, and scaffolding student inquiry (Mishra & Koehler, 2006). Where such competence is limited, computational tools may be reduced to superficial visual aids rather than instruments of conceptual reasoning; the present study suggests that when teachers effectively integrate computational thinking into pedagogical design, technology becomes a powerful cognitive mediator rather than a passive display tool, and this mediation is reinforced by the large effect sizes reported in **Table 21**. This is in accordance with Alzubi's (2022) study of e-learning-supported mathematics teaching in a college setting, which likewise reported measurable gains in students' mathematical skills when digital platforms were incorporated into instruction; however, that study did not isolate the contribution of teacher-mediated pedagogical design from the technology itself, whereas the present three-group comparison shows that the size of the learning gain depends specifically on how competently computational tools are integrated into conceptual teaching, rather than on their mere presence.

This distinction has particular relevance for developing educational systems, where mathematics instruction is often shaped by large class sizes, exam-oriented curricula, and limited teacher training in pedagogy or computational tools (UNESCO, 2020; World Bank, 2018), and where access to technology may be increasing faster than pedagogical capacity building. This finding aligns with a recent Delphi study of experts examining the role of mathematics in an increasingly technological society, which concluded that preparing students for “Education 4.0” requires deliberate curricular and pedagogical redesign rather than the incidental adoption of digital tools (Taja-on et al., 2025). The present study extends this consensus by providing classroom-level experimental evidence of how such redesign can be implemented in first-year calculus and the learning gains it can generate. The findings suggest that meaningful improvement does not require advanced technological infrastructure so much as integrating computational thinking into teacher practice: even relatively simple computational tools, used pedagogically, produced large gains in this study (**Tables 21** and **23**). This implies that educational reform in such contexts may benefit from prioritizing teacher training in computational pedagogy, redesigning instructional materials around multiple representations, and shifting assessment toward conceptual and transfer-based evaluation, alongside — rather than instead of — investment in infrastructure.

From a theoretical perspective, the findings support a model in which conceptual learning in calculus emerges from the interaction of three components — conceptual knowledge structures, computational thinking processes, and teacher-mediated instructional design — rather than from any one of them independently. Computational thinking functions here as an internal cognitive and representational process rather than an external tool (**Figure 6**), while teacher competence acts as a stabilizing mechanism that determines whether computational environments lead to deep learning or superficial interaction (**Figure 7**). As this study was conducted in a first-year university calculus setting, the proposed framework is intended as a flexible instructional model that can be adapted to local educational conditions using available technologies and institutional resources. This theoretical positioning is in line with Ye et al.'s (2023) systematic review of computational-thinking-based mathematics instruction, which found across K-12 settings that computational thinking is most effective for mathematics learning when integrated as a reasoning process embedded within the mathematical content rather than taught as a separate skill. The present study is consistent with that conclusion and extends it to the first-year university level, and to a comparative experimental design that also isolates the mediating role of teacher competence, which the K-12 studies reviewed by Ye et al. (2023) did not systematically examine.

CONCLUSION, LIMITATIONS, AND FUTURE RESEARCH

This study investigated the impact of three instructional approaches — traditional procedural teaching, structured conceptual teaching, and computationally enhanced conceptual teaching — on first-year engineering students' understanding of differential calculus, focusing specifically on derivatives and optimization. The central objective was to examine how computational thinking, when integrated into pedagogically structured instruction and supported by teacher competence, influences students' conceptual development, retention, and ability to transfer knowledge to unfamiliar problems.

The empirical results provide clear and consistent evidence across all assessment dimensions. Students exposed to computationally enhanced conceptual instruction significantly outperformed those in both the conceptual and traditional groups in post-test achievement, delayed retention, and transfer tasks (**Tables 16–18**), and the magnitude of the observed differences was not only statistically significant but also educationally substantial, as confirmed by the one-way ANOVA (**Table 19**), the Tukey post-hoc comparisons (**Table 20**), and the large effect sizes reported in **Table 21**. From a pedagogical perspective, the findings suggest that traditional instruction is effective for procedural fluency but insufficient for conceptual development; that conceptual instruction improves interpretive understanding but may remain limited when mathematical ideas are not dynamically explored; and that computationally enhanced instruction creates a more powerful learning environment by integrating symbolic, graphical, and numerical representations within a dynamic framework of mathematical exploration. A key theoretical implication is that computational thinking should not be viewed merely as a technological add-on, but as a cognitive and representational bridge that enables students to perceive calculus concepts as evolving mathematical processes rather than static formulas — an advantage clearly visible in the multidimensional learning outcomes summarized in **Table 22** and **Figure 11**. Furthermore, the study highlights the critical role of teacher competence in shaping learning outcomes: the effectiveness of computational tools depends not on their presence alone, but on how effectively teachers design, structure, and guide computationally enriched learning experiences. Overall, the study supports a shift in calculus pedagogy from rule-based instruction toward an integrated computational-conceptual framework that fosters deeper mathematical understanding and improved transfer of knowledge, a conclusion reinforced by the delayed retention evidence (**Table 17**; **Figure 10**), the transfer-task outcomes (**Table 18**), and the gain-score comparison (**Table 23**; **Figure 12**).

Several limitations should be considered when interpreting these findings. First, the study was conducted within a single institutional and cultural context, so generalization to other educational systems should be made cautiously, particularly where curricular structures or technological infrastructure differ substantially. Second, the sample size, although adequate for statistical analysis, was limited to 144 students; larger multi-institutional studies would provide stronger external validity. Third, the intervention focused exclusively on a single calculus topic — derivatives and optimization — over four weeks, and other areas such as integration, sequences, or differential equations may exhibit different learning dynamics under computational instruction, so the findings should be read as evidence of short-term instructional effectiveness rather than long-term conceptual development. Fourth, the study did not explicitly control for prior individual differences in digital literacy or familiarity with computational tools, which may have influenced engagement and performance in the computational group. Fifth, although the proposed framework is intended to be adaptable across educational contexts, it was evaluated within a specific university setting and a limited instructional period, so future research should examine its scalability across different educational systems, larger student

populations, and varying levels of technological availability. Sixth, although teacher competence was identified as a critical factor conceptually and was held constant across groups, it was not quantitatively measured as an independent variable; future studies could benefit from developing structured instruments to assess teacher computational-pedagogical competence more explicitly. Finally, the study primarily measured cognitive learning outcomes, and affective or longer-term behavioral outcomes, such as sustained interest in mathematics or continued use of computational tools beyond the classroom, were not examined in depth.

These findings suggest several directions for future research. The proposed framework should be extended to other areas of mathematics, such as integration, multivariable calculus, and differential equations, to evaluate its broader effectiveness. Longitudinal studies are needed to examine the retention and transfer of computational reasoning, while future work should also develop validated measures of teachers' computational-pedagogical competence. Comparative studies across different educational systems could clarify the influence of institutional context, and research integrating AI-based tutoring, adaptive learning, and qualitative classroom analysis may further enhance understanding of computational thinking and conceptual development in mathematics.

In conclusion, this study provides empirical and theoretical evidence that integrating computational thinking into calculus instruction — when guided by competent pedagogical design — significantly enhances students' conceptual understanding, retention, and transfer of knowledge, and offers a scalable instructional framework for educational systems seeking to strengthen mathematics teaching quality through computational pedagogy.

Author contributions: AN & AS: conceptualization, design, implementation, formal analysis, writing – original draft, writing – review & editing. Both authors approved the final version.

Funding: No funding source is reported for this study.

Acknowledgements: The authors would like to acknowledge colleagues whose discussions helped improve the clarity of this manuscript.

Ethical statement: The authors stated that the activities conducted in this study were based on standard educational practices within the regular calculus learning environment. No sensitive personal information was collected, and the study involved no foreseeable risks to participants. Participants were informed that the collected data would be used solely for academic and research purposes, handled securely, and that participation was voluntary. Participants were also informed of their right to withdraw from the study at any time without any adverse consequences.

AI statement: The authors stated that generative AI and AI-assisted tools were used solely for English-language grammar checking and were not used in any way that violates ethical standards.

Declaration of interest: No conflict of interest is declared by the authors.

Data sharing statement: Data supporting the findings and conclusions are available upon request from the corresponding author.

REFERENCES

- Afriani, S. U., Mangaraja, A., & Siregar, R. N. (2026). Analysis of prospective mathematics teachers difficulties on learning calculus. *Journal of Innovative Mathematics Learning*, 9(2), 465–478. <https://doi.org/10.22460/jiml.v9i2.30746>
- Alam, A. (2020). Challenges and possibilities in teaching and learning of calculus: A case study of India. *Journal for the Education of Gifted Young Scientists*, 8(1), 407–433. <https://doi.org/10.17478/jegys.660201>
- Alzubi, K. A. A. (2022). The effect of teaching mathematics supported by e-learning platforms on the students' mathematical skills in a college course in Jordan. *International Journal of Emerging Technologies in Learning*, 17(12). <https://doi.org/10.3991/ijet.v17i12.30049>
- Artigue, M. (2002). Learning mathematics in a CAS environment: The genesis of a reflection about instrumentation and the dialectics between technical and conceptual work. *International Journal of Computers for Mathematical Learning*, 7(3), 245–274. <https://doi.org/10.1023/A:1022103903080>
- Ball, D. L., Thames, M. H., & Phelps, G. (2008). Content knowledge for teaching: What makes it special? *Journal of Teacher Education*, 59(5), 389–407. <https://doi.org/10.1177/0022487108324554>
- Blömeke, S., Jentsch, A., Ross, N., Kaiser, G., & König, J. (2022). Opening up the black box: Teacher competence, instructional quality, and students' learning progress. *Learning and Instruction*, 79, Article 101600. <https://doi.org/10.1016/j.learninstruc.2022.101600>
- Chiu, T. K., Falloon, G., Song, Y., Wong, V. W., Zhao, L., & Ismailov, M. (2024). A self-determination theory approach to teacher digital competence development. *Computers & Education*, 214, Article 105017. <https://doi.org/10.1016/j.compedu.2024.105017>
- Demetriou, A., Kazi, S., Spanoudis, G., & Makris, N. (2019). Predicting school performance from cognitive ability, self-representation, and personality from primary school to senior high school. *Intelligence*, 76, Article 101381. <https://doi.org/10.1016/j.intell.2019.101381>
- Durandt, R. (2026). What first-year students' prior mathematical knowledge reveals for modelling-oriented university mathematics teaching. *African Journal of Teacher Education and Development*, 5(1), 1–10. <https://doi.org/10.4102/ajoted.v5i1.161>
- Duval, R. (2006). A cognitive analysis of problems of comprehension in a learning of mathematics. *Educational Studies in Mathematics*, 61(1–2), 103–131. <https://doi.org/10.1007/s10649-006-0400-z>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. <https://www.deeplearningbook.org>
- Harahap, M. I., Herianto, R., & Siregar, R. N. (2025). Analysis of difficulties in understanding calculus concepts: A case of teachers and students. *Journal of Innovative Mathematics Learning*, 8(4), 877–889. <https://doi.org/10.22460/jiml.v8i4.30749>

- Hassan, U., Raja, A. H., Hassan, S., & Hassan, Z. (2026). Zero aptitude for learning: The silent epidemic killing curiosity in our classrooms. *Social Science Review Archives*, 4(2), 1448–1458. <https://doi.org/10.70670/sra.v4i2.2224>
- Hiebert, J., & Lefevre, P. (1986). Conceptual and procedural knowledge in mathematics: An introductory analysis. In J. Hiebert (Ed.), *Conceptual and procedural knowledge: The case of mathematics* (pp. 1–27). Lawrence Erlbaum Associates.
- Irawan, E., Rosjanuardi, R., & Prabawanto, S. (2024). Research trends of computational thinking in mathematics learning: A bibliometric analysis from 2009 to 2023. *Eurasia Journal of Mathematics, Science and Technology Education*, 20(3), Article em2417. <https://doi.org/10.29333/ejmste/14343>
- Kaput, J. J. (1994). The representational roles of technology in connecting mathematics with authentic experience. In R. Biehler, R. W. Scholz, R. Strässer, & B. Winkelmann (Eds.), *Didactics of mathematics as a scientific discipline* (pp. 379–397). Kluwer Academic Publishers.
- Kurudirek, A., Karim, B., Sarhang, D., & Machisi, E. (2025). Challenges faced by first-year university students in learning calculus. *Contemporary Mathematics and Science Education*, 6(2), Article ep25015. <https://doi.org/10.30935/conmaths/17298>
- Mazzia, F. (2022). A computational point of view on teaching derivatives. *Informatics and Education*, 37(1), 79–86. <https://doi.org/10.32517/0234-0453-2022-37-1-79-86>
- Mishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record*, 108(6), 1017–1054. <https://doi.org/10.1111/j.1467-9620.2006.00684.x>
- Ocal, M. F. (2017). The effect of GeoGebra on students' conceptual and procedural knowledge: The case of applications of derivative. *Higher Education Studies*, 7(2), 67–78. <https://doi.org/10.5539/hes.v7n2p67>
- Orton, A. (1983). Students' understanding of differentiation. *Educational Studies in Mathematics*, 14(3), 235–250. <https://doi.org/10.1007/BF00410540>
- Özgeldi, M., & Aydın, U. (2021). Identifying competency demands in calculus textbook examples: The case of integrals. *International Journal of Science and Mathematics Education*, 19(1), 171–191. <https://doi.org/10.1007/s10763-019-10046-9>
- Rodríguez-Muñiz, L. J., Aguilar-González, Á., Lindorff, A., & Muñiz-Rodríguez, L. (2022). Undergraduates' conceptions of mathematics teaching and learning: An empirical study. *Educational Studies in Mathematics*, 109, 523–547. <https://doi.org/10.1007/s10649-021-10105-5>
- Rybak, A. (2021). Effectiveness of teaching and learning in technology-supported mathematics education. *Journal of Physics: Conference Series*, 1946(1), Article 012004. <https://doi.org/10.1088/1742-6596/1946/1/012004>
- Salloum, S., & BouJaoude, S. (2019). The use of triadic dialogue in the science classroom: A teacher negotiating conceptual learning with teaching to the test. *Research in Science Education*, 49(3), 829–857. <https://doi.org/10.1007/s11165-017-9640-4>
- Schoenfeld, A. H. (1992). Learning to think mathematically: Problem solving, metacognition, and sense-making in mathematics. In D. A. Grouws (Ed.), *Handbook of research on mathematics teaching and learning* (pp. 334–370). Macmillan.
- Sfard, A. (1991). On the dual nature of mathematical conceptions: Reflections on processes and objects as different sides of the same coin. *Educational Studies in Mathematics*, 22(1), 1–36. <https://doi.org/10.1007/BF00302715>
- Shulman, L. S. (1986). Those who understand: Knowledge growth in teaching. *Educational Researcher*, 15(2), 4–14. <https://doi.org/10.3102/0013189X015002004>
- Shulman, L. S. (1987). Knowledge and teaching: Foundations of the new reform. *Harvard Educational Review*, 57(1), 1–23.
- Stewart, J., Clegg, D., & Watson, S. (2021). *Calculus: Early transcendentals* (9th ed.). Cengage Learning.
- Sun, Y., Sahinidis, N. V., Sundaram, A., & Cheon, M. S. (2020). Derivative-free optimization for chemical product design. *Current Opinion in Chemical Engineering*, 27, 98–106. <https://doi.org/10.1016/j.coche.2019.11.006>
- Taja-on, E. P., Dajero, B. K. C., & Barete, M. G. (2025). Mathematics and modern society: A Delphi study exploring mathematics education towards Education 4.0. *Educational Point*, 2(1), Article e120. <https://doi.org/10.71176/edup/16534>
- Tall, D. (Ed.). (1991). *Advanced mathematical thinking*. Kluwer Academic Publishers. <https://doi.org/10.1007/0-306-47203-1>
- Tarasov, V. E. (2016). Geometric interpretation of fractional-order derivative. *Fractional Calculus and Applied Analysis*, 19(5), 1200–1221. <https://doi.org/10.1515/fca-2016-0062>
- Thompson, P. W. (1994). Images of rate and operational understanding of the fundamental theorem of calculus. *Educational Studies in Mathematics*, 26(2–3), 229–274. <https://doi.org/10.1007/BF01273664>
- UNESCO. (2020). *Global education monitoring report 2020: Inclusion and education: All means all*. UNESCO. <https://doi.org/10.54676/JJNK6989>
- Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715–728. <https://doi.org/10.1007/s10639-015-9412-6>
- Weber, E., Tallman, M., Byerley, C., & Thompson, P. W. (2012). Understanding the derivative through the calculus triangle. *Mathematics Teacher*, 106(4), 274–278. <https://doi.org/10.5951/mathteacher.106.4.0274>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>
- World Bank. (2018). *World development report 2018: Learning to realize education's promise*. World Bank. <https://doi.org/10.1596/978-1-4648-1096-1>

- Yang, X., & Du, J. (2024). The effect of teacher self-efficacy, online pedagogical and content knowledge, and emotion regulation on teacher digital burnout: A mediation model. *BMC Psychology*, 12, Article 51. <https://doi.org/10.1186/s40359-024-01540-z>
- Ye, H., Liang, B., Ng, O.-L., & Chai, C. S. (2023). Integration of computational thinking in K–12 mathematics education: A systematic review on CT-based mathematics instruction and student learning. *International Journal of STEM Education*, 10, Article 3. <https://doi.org/10.1186/s40594-023-00396-w>
- Zandieh, M. (2000). A theoretical framework for analyzing student understanding of the concept of derivative. *CBMS Issues in Mathematics Education*, 8, 103–127. <https://doi.org/10.1090/cbmath/008/06>

APPENDIX

Table A1. Pre-test — Baseline mathematical readiness

No.	Assessment statement / task	Dim.
1	Explain in your own words what the rate of change of a function represents.	CU
2	Simplify algebraic expressions involving powers, fractions, and parentheses.	PF
3	Solve a basic algebraic equation involving one unknown.	PF
4	Interpret the slope of a straight line from its graph.	GI
5	Identify whether a function is increasing, decreasing, or constant from a graph.	GI
6	Estimate the average rate of change of a function over a given interval.	CU
7	Explain the difference between average rate of change and instantaneous rate of change.	CU
8	Use algebraic manipulation to simplify a difference quotient.	PF
9	Identify the point at which a tangent line is drawn on a function graph.	GI
10	Describe what the slope of a tangent line represents.	CU
11	Recognize the basic meaning of the derivative of a function.	CU
12	Use prerequisite differentiation rules to find the derivative of a simple polynomial.	PF
13	Determine from a graph where a function appears to have a local maximum or minimum.	GI
14	Explain how the sign of a derivative is related to the behavior of a function.	CU
15	Use a simple mathematical model to reason about how one quantity changes with another.	CU

Dimension codes: CU = Conceptual Understanding; PF = Procedural Fluency; GI = Graphical Interpretation; OR = Optimization Reasoning; CR = Computational Reasoning.

Table A2. Post-test — Procedural and conceptual learning

No.	Assessment statement / task	Dim.
1	Differentiate a polynomial function using appropriate differentiation rules.	PF
2	Differentiate a product or quotient of functions.	PF
3	Differentiate a composite function using the chain rule.	PF
4	Explain in your own words what the derivative represents at a given point.	CU
5	Interpret the derivative as the slope of the tangent line.	CU
6	Determine whether a function is increasing or decreasing from the sign of its derivative.	GI
7	Use a graph of a function and its derivative to interpret their relationship.	GI
8	Identify critical points from a derivative function.	PF
9	Use the first-derivative test to classify a critical point.	OR
10	Formulate an optimization problem from a stated real-world situation.	OR
11	Use derivatives to determine a maximum or minimum value.	OR
12	Justify why a calculated critical point gives the required optimum.	OR
13	Interpret an optimization result in the context of the original problem.	OR
14	Explain the mathematical meaning of a zero derivative.	CU
15	Explain the relationship between derivative sign and function behavior.	CU
16	Approximate an instantaneous rate of change from numerical data.	CR
17	Interpret a numerical or graphical computational output describing a derivative.	CR
18	Compare numerical approximations of rates of change and identify a meaningful pattern.	CR
19	Use graphical or numerical evidence to justify a conclusion about an optimization problem.	CR
20	Explain the reasoning used to connect a computational result with the underlying derivative concept.	CR

Dimension codes: CU = Conceptual Understanding; PF = Procedural Fluency; GI = Graphical Interpretation; OR = Optimization Reasoning; CR = Computational Reasoning.

Table A3. Delayed retention test — Parallel form

No.	Assessment statement / task	Dim.
1	Explain what the derivative means in the context of a changing quantity.	CU
2	Interpret the slope of a tangent line at a specified point.	CU
3	Determine whether a function is increasing or decreasing using derivative information.	GI
4	Interpret a graph showing a function and its derivative.	GI
5	Identify a possible local maximum or minimum from graphical information.	GI
6	Explain the relationship between a critical point and an optimization problem.	OR
7	Use derivative information to identify an optimal value.	OR
8	Interpret the meaning of an optimization result in context.	OR
9	Justify an optimization decision using mathematical reasoning.	OR
10	Explain why the sign of the derivative changes near a local extremum.	CU
11	Interpret an approximate derivative obtained from numerical data.	CR
12	Recognize a meaningful pattern in numerical rates of change.	CR
13	Use graphical evidence to support a conclusion about derivative behavior.	CR
14	Apply derivative concepts to a short optimization interpretation problem.	OR
15	Explain how derivative information can be used to reason about change.	CU

Dimension codes: CU = Conceptual Understanding; PF = Procedural Fluency; GI = Graphical Interpretation; OR = Optimization Reasoning; CR = Computational Reasoning.

Table A4. Transfer task — Unfamiliar optimization problems

No.	Assessment statement / task	Dim.
1	A rectangular enclosure is to be constructed with a fixed amount of fencing. Formulate a mathematical model and determine the dimensions that maximize the enclosed area. Justify your answer using derivatives.	OR
2	A container with a specified volume is to be designed under a stated geometric constraint. Develop an appropriate objective function and determine the dimensions that optimize the required quantity. Explain your reasoning.	OR
3	An unfamiliar cost or production situation is described by a nonlinear model. Formulate the objective function, use derivative information to locate the optimum, and interpret the result in context.	OR
4	Given a novel graphical or numerical optimization situation, identify the relevant variables, formulate the mathematical relationship, determine the optimum, and justify why the solution is appropriate.	OR

Dimension codes: CU = Conceptual Understanding; PF = Procedural Fluency; GI = Graphical Interpretation; OR = Optimization Reasoning; CR = Computational Reasoning.