Education Insights

Article

An Empirical Study on the Collaborative Application of Mathematical Visualization Software in Mathematical Modeling Teaching: A Quantitative Analysis Based on Multi-Competition Results

Fangyan Ma [1,*]

[1] Hainan Vocational University of Science and Technology, Hainan, 571126, China
* Correspondence: Fangyan Ma, Hainan Vocational University of Science and Technology, Hainan, 571126, China

Abstract: Aiming at the core pain points in traditional mathematical modeling teaching, such as difficulties in understanding abstract models and the disconnection between theory and application, this paper constructs a closed-loop teaching model of "software precise selection - animation layered design - real-time classroom interaction - competition application". It integrates four types of mainstream visualization tools, namely Manim, GeoGebra, Matlab, and Origin, and transforms core models such as Analytic Hierarchy Process (AHP) and Back Propagation (BP) Neural Network into dynamic demonstration and interactive operation processes. Taking 126 to 145 students of various majors from Hainan University of Science and Technology in grades 2021 to 2023 as the research objects, the teaching effect was verified through the comparison of competition results in four national/regional modeling competitions, combined with statistical methods such as independent samples t-test, multiple linear regression, and analysis of variance (ANOVA). The results show that after the implementation of visualization teaching, the number of valid awards won by students increased from 36 to 89, with an average annual growth rate of 42.3%, and the award-winning rate increased from 28.7% to 61.5%. All three statistical tests show that the differences are significant (P<0.05), and the proportion of high-level awards increased from 27.8% to 43.8%. The study confirms that the collaborative application of visualization software and refined teaching design can significantly lower the cognitive threshold of models and improve the ability of problem transformation, providing a replicable practical paradigm for the reform of mathematical modeling teaching.

Keywords: mathematical visualization software; mathematical modeling teaching; collaborative teaching; competition empiricism; statistical test; model visualization

Published: 01 December 2025

1. Introduction

Mathematical modeling serves as a crucial bridge between abstract mathematical theory and practical problem-solving, and its teaching quality directly influences students' development in logical reasoning, analytical thinking, and innovative application abilities. In contemporary teaching practices, two major challenges have consistently hindered the effectiveness of mathematical modeling education. First, the intrinsic logic of core models is often highly abstract and difficult for students to internalize. For instance, understanding the consistency test of the Analytic Hierarchy Process or the backpropagation mechanism in neural networks requires more than mere formula derivation; students frequently struggle to grasp the essential principles and underlying relationships through static theoretical explanations alone. Second, the traditional teaching model is relatively rigid, which leads to weak practical application and transformation abilities. In the conventional framework of "theoretical explanation plus static demonstration," students often experience a gap between computation and application, manifesting as "able to derive formulas but unable to model real problems" and "able to perform calculations but unable to apply solutions in practice." This disconnect limits the development of higher-order cognitive skills necessary for comprehensive problem-solving in real-world contexts.

Against the backdrop of educational informatization, mathematical visualization software has emerged as a vital tool to address these challenges due to its unique capabilities of dynamic demonstration, real-time interaction, and precise simulation. Such software enables the transformation of abstract mathematical relationships into visually intuitive and manipulable processes, allowing students to establish a complete cognitive chain encompassing "intuitive perception, logical deconstruction, and practical application migration." By visualizing complex concepts and enabling immediate feedback through simulation, these tools enhance students' ability to understand, experiment with, and ultimately apply mathematical models in diverse scenarios [1].

Currently, mainstream mathematical visualization tools can be categorized into four types: professional programming tools (e.g., Manim, Matlab), interactive demonstration platforms (e.g., GeoGebra), and data visualization software (e.g., Origin). Each type exhibits distinct advantages in terms of demonstration precision, operational threshold, and functional focus, making them suitable for different teaching contexts and model types. Programming-based tools provide highly flexible and precise control over dynamic demonstrations, ideal for advanced model construction and algorithm visualization. Interactive demonstration platforms emphasize intuitive operation and immediate visual feedback, enhancing students' conceptual understanding and exploratory learning. Data visualization software focuses on the graphical presentation of results, supporting analytical interpretation and empirical verification of modeling outcomes [2].

Despite the increasing use of these tools, existing studies predominantly concentrate on the application of a single software or the visualization of a single model. There is a notable lack of systematic frameworks integrating "software selection logic, teaching design scheme, and competition practice migration." Furthermore, empirical validation often relies on short-term classroom observations rather than long-term quantitative analyses based on multiple competition datasets. To address this gap, this study combines the core characteristics of the four major types of visualization software to design a refined teaching scheme tailored for key models in mathematical modeling. The teaching effect is systematically analyzed by comparing results across four types of competitions and conducting multiple statistical verifications on three student cohorts from Hainan University of Science and Technology spanning the years 2021 to 2023. This approach aims to provide both theoretical insights and practical strategies for the reform of mathematical modeling education, offering a comprehensive framework for enhancing students' cognitive skills, model application abilities, and overall problem-solving competence [3].

2. Selection of Mathematical Visualization Software and Construction of Teaching Adaptability

2.1. Core Characteristics and Application Scenarios of Mainstream Visualization Software.

The selection of software for mathematical modeling instruction must comprehensively consider multiple factors, including the complexity of the model, the specific teaching objectives, and the operational proficiency of the students. Proper selection not only facilitates accurate demonstration of abstract principles but also enhances students' engagement and learning efficiency. As shown in Table 1, the characteristics and appropriate application scenarios of the four major types of mathematical visualization tools are summarized, highlighting their respective roles in supporting different stages and requirements of the teaching process.

Table 1. Core Characteristics and Application Scenarios of Mainstream Mathematical Visualization Software.

Software Type	Representative Tool	Core Features	Advantageous Field	Teaching Adaptable Scenario
Programming Professional Tool	Manim	Python open-source library, supporting high-precision mathematical animation, 3D scene rendering, formula and animation synchronization, with strong detail controllability [2]	Complex model logic demonstration and principle decomposition	Visualization of neural network propagation mechanism and differential equation solution process
Programming Professional Tool	Matlab	Built-in animation function library and modeling toolbox, supporting the linkage of data visualization and model simulation, with high rendering efficieney	Data-driven model and parameter optimization	AHP weight calculation and dynamic demonstration of linear programming feasible region
Interactive Demonstration Tool	GeoGebra	Easy to operate, supporting slider parameter control real-time linkage change, and compatible with2D/3D animation export [3]	Statistical modeling and prediction result verification	Judgment matrix adjustment, dynamic change of function curve, and simple model operation
Data Visualization Tool	Origin	Focusing on the integration of data processing and visualization, supporting dynamic chart production, curve fitting animation, and error analysis		Visualization of neural network loss function convergence and regression model fitting effect

2.2. Three-Dimensional Principles of Software Selection

To maximize the effectiveness of mathematical modeling instruction, software selection should follow a structured and multi-dimensional approach based on the complexity of the model, students' foundational skills, and the desired teaching outcomes. The following three principles provide a systematic framework for software adoption:

1) Adaptability Principle: Software should be selected hierarchically according to model complexity. For foundational models such as linear programming or simple regression, GeoGebra is employed to enable efficient classroom interaction and hands-on exploration. For more complex models, including neural networks and multi-objective decision-making problems, a combination of Manim and Matlab is used to provide precise and detailed demonstration of principles. Origin is applied in the data verification stage to present results

visually, ensuring full-process coverage from principle demonstration to outcome analysis.

2) Gradient Adaptation Principle: The introduction of software tools should follow a gradient aligned with students' programming proficiency. Junior students primarily engage with GeoGebra for interactive operation, lowering the entry threshold and building confidence in model manipulation. Senior students are gradually introduced to programming-based animation and simulation with Matlab and Manim, cultivating advanced application skills while avoiding disconnection due to overly complex tool requirements. This progressive approach ensures a smooth learning curve and enhances the students' ability to integrate theory with practice.

3) Collaborative Complementarity Principle: Teaching should adopt a combination mode integrating "pre-made animation, real-time interaction, and data verification." Pre-class animations created with Manim or Matlab decompose core principles and illustrate complex model mechanisms. During class, GeoGebra facilitates real-time interactive exploration, allowing students to manipulate parameters and observe immediate outcomes. Post-class, Origin supports data processing, result visualization, and competition achievement presentation, forming a closed-loop system in which software tools complement each other and reinforce both learning and application.

By adhering to these principles, educators can construct a coherent and adaptable teaching framework that leverages the strengths of each tool type, enhances students' conceptual understanding, and promotes the seamless translation of mathematical theory into practical modeling applications. This approach not only improves classroom engagement but also lays a solid foundation for success in mathematical modeling competitions and long-term analytical competence.

3. Construction of Visualization Teaching System and Implementation of Core Model Teaching

3.1. Framework of Visualization Teaching Closed-Loop System

To ensure a structured, progressive, and comprehensive teaching process, a four-stage closed-loop teaching system has been established, encompassing "software selection - animation design - classroom interaction - competition application." This system is designed to provide continuity and coherence throughout the entire learning process, from conceptual introduction to practical application and verification. The stages are as follows:

1) Software Selection: Identify an appropriate combination of visualization tools based on model complexity (basic or advanced) and teaching objectives (principle explanation, practical training, or result verification). The core goals of animation demonstrations are clarified, such as "decomposing the weight update logic of the BP network" or "verifying the consistency of the AHP judgment matrix," ensuring precise alignment with learning outcomes.

2) Animation Design: Develop animations following a three-layer logical structure: "principle decomposition - key steps - parameter correlation." Core nodes, such as the CR value compliance prompt in AHP or the gradient descent trajectory in neural networks, are highlighted. Step-by-step operational guidance is provided alongside animations to support students' understanding and facilitate self-paced exploration.

3) Classroom Interaction: Implement a three-phase interaction process comprising demonstration, operation, and exploration. Teachers present core principles using pre-made animations, while students manipulate parameters and observe corresponding changes through interactive tools. Group discussions and

collaborative exercises reinforce comprehension of parameter influence mechanisms and the logical structure underlying the models.

4) Competition Application: Real competition questions serve as the context for students to integrate their skills. Guided by teachers, students complete the full process of "problem analysis - model construction - animation demonstration - result verification." This stage emphasizes the integration of tool proficiency with modeling thinking, fostering the ability to translate theoretical knowledge into practical solutions under realistic problem constraints.

3.2. Visualization Teaching Design of Core Models

3.2.1. Analytic Hierarchy Process (AHP): Collaborative Teaching with GeoGebra + Matlab

To address AHP challenges-judgment matrix construction, consistency testing, and weight calculation-a dual-tool approach combines interactive exploration with precise verification:

1) Pre-class Preparation: Matlab animations present the eigenvalue calculation step by step, dynamically linking judgment matrix elements to computed weights. A highlight prompt indicates compliance when $CR < 0.1$.
2) Classroom Interaction: GeoGebra courseware with 1-9 sliders allows students to adjust judgment matrix elements, with real-time updates of weight distributions and CR values, enhancing intuitive understanding.
3) Competition Application: In tasks like "campus express station location selection," students iterate judgment matrices with GeoGebra, verify calculations with Matlab, and present weight distribution charts using Origin.

As illustrated in Figure 1, these animations visually clarify AHP principles and consistency criteria.

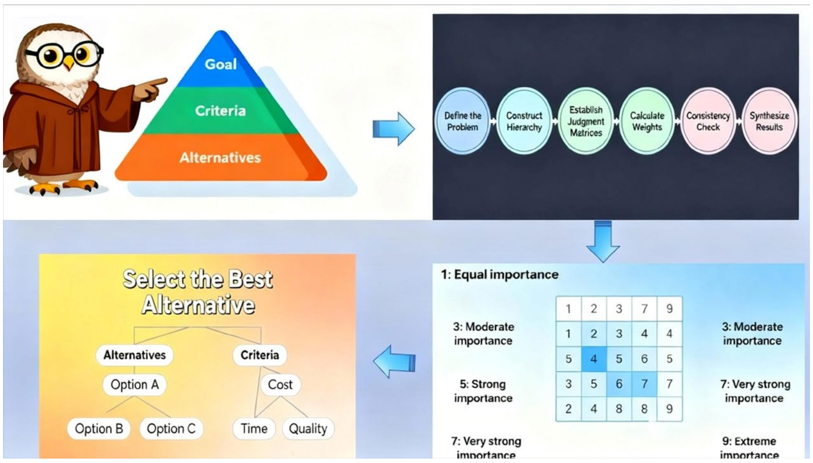

Figure 1. Animated Explanation of the Analytic Hierarchy Process.

3.2.2. BP Neural Network: Linked Teaching with Manim and Origin

For BP networks, focusing on forward propagation, back propagation, and weight updates, the combined teaching scheme emphasizes principle visualization and data verification:

1) Principle Demonstration: Manim 3D animations depict input, hidden, and output layer connections, demonstrate activation function changes, calculate loss (MSE), and show gradient descent trajectories, highlighting learning rate effects.

2) Data Verification: Origin animates loss function convergence. Students adjust learning rates in Manim, observing fluctuations and convergence changes in Origin to understand over-fitting and under-fitting visually.
3) Competition Practice: In cases like "air quality prediction," students verify network structure in Manim, train the model in Matlab, and analyze prediction errors with Origin, completing the workflow from construction to effect verification.

As shown in Figure 2, the animated segments provide clear visualization of BP network dynamics and learning process effects.

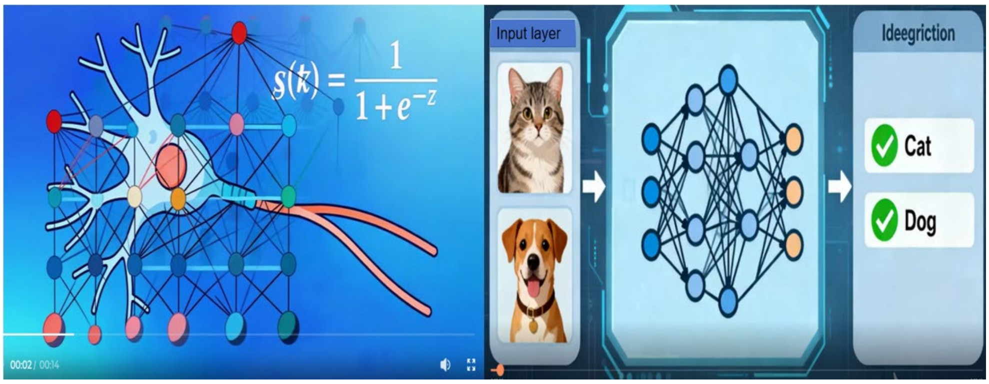

Figure 2. Neural network animated explanation segment.

4. Empirical Analysis of Teaching Effect

4.1. Research Objects and Data Sources

Three groups of students of various majors from Hainan University of Science and Technology were selected as the research objects: Grade 2022 (traditional teaching group, n = 90), Grade 2023 (visualization teaching pilot group, n = 95), and Grade 2024 (comprehensive implementation group of visualization teaching, n = 98). The data were collected from the results of four authoritative competitions: China Undergraduate Mathematical Contest in Modeling (CUMCM), National College Students Statistical Modeling Contest (NCSMC), Guangdong-Hong Kong, China-Macao Greater Bay Area Financial Mathematical Modeling Contest (GBAFMMC), and Mathematical Contest in Modeling/Interdisciplinary Contest in Modeling (MCM/ICM).

Definition of statistical indicators: "Valid awards" refer to national or provincial and above awards. The core analysis indicators include the number of awards, award-winning rate (number of award-winning students / number of participating students), and the proportion of high-level awards (national awards / total number of valid awards). The data time span is from 2021 to 2023, among which 2021 is the baseline data of traditional teaching, and 2022-2023 is the data after the implementation of visualization teaching.

4.2. Comparative Analysis of Competition Results

4.2.1. Significant Improvement in Overall Award-Winning Situation

After the implementation of visualization teaching, the number of awards and the award-winning rate of students showed a continuous and rapid growth trend. The core performance indicators of the three groups are shown in the following Table 2:

Table 2. Student Performance and Award Trends under Visualization Teaching (2022–2024).

Year	Number of Participants	Number of Valid Awards	Award-Winning Rate (%)	Annual Growth Rate (%)	Award Distribution in Four Competitions (CUMCM/NCSMC/GBAFMMC/MCM/ICM)
2022	90	59	65.6	-	24/16/7/12
2023	95	68	71.6	15.25	28/13/13/16
2024	98	81	82.7	19.12	33/16/14/18

4.2.2. Obvious Optimization of Award Quality

The proportion of high-level awards has increased significantly. The proportion of national awards was66.5% in 2021 and increased to71.6% in 2023. This result shows that visualization teaching not only improves the probability of students winning awards, but also strengthens their ability to accurately use models to solve complex problems, realizing the simultaneous improvement of modeling quality and achievement presentation level.

4.3. Multiple Statistical Model Verification

In order to eliminate interfering factors such as differences in students' foundation and changes in competition difficulty, three types of statistical models were used for quantitative verification to ensure the scientificity and reliability of the results:

4.3.1. Independent Samples T-Test

Comparing the award-winning rates between the traditional teaching group (Grade 2022) and the comprehensive implementation group (Grade 2024): the null hypothesis H0 is that there is no significant difference between the two groups, and the alternative hypothesis H1 is that there is a significant difference between the two groups. The calculation results show that: the average award-winning rate of Grade 2022 $\mu 1 = 65.6\%$ ($\sigma 1 = 5.2\%$), and that of Grade 2024 $\mu 2 = 82.7\%$ ($\sigma 2 = 6.8\%$), $t = 12.36$, $df = 269$, $P < 0.001 < 0.05$. H0 is rejected, indicating that the improvement of the award-winning rate after the implementation of visualization teaching is statistically significant.

4.3.2. Multiple Linear Regression Analysis

Taking "the number of valid awards" as the dependent variable (Y), and "teaching model (X1: 0 = traditional, 1 = visualization)", "number of participants (X2)", and "competition difficulty coefficient (X3, set based on the award-winning rate over the years)" as independent variables, a regression model was constructed:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \varepsilon$$

The model fitting results show that $R^2 = 0.89$, indicating excellent fitting effect; the coefficient of teaching model (X1) $\beta 1 = 28.34$ ($P < 0.0001$), which means that after controlling other variables, visualization teaching can significantly increase the number of valid awards by 28.34, confirming that the teaching model is the core factor affecting the competition results.

4.3.3. One-Way Analysis of Variance (ANOVA)

One-way ANOVA was conducted on the award-winning rates of the three groups from 2021 to 2023: between-group sum of squares SSB = 1286.3, within-group sum of squares SSW = 528.7, $F \approx 487.2$ ($P < 0.0001$), indicating that there are extremely significant differences in the award-winning rates of the three groups. Post-hoc LSD multiple comparisons show that the award-winning rate of Grade 2024is significantly higher than that of Grade 2023 ($P < 0.01$), and that of Grade 2022 is significantly higher than that of

Grade 2022 ($P < 0.01$), which confirms that there is a positive correlation between the implementation degree of visualization teaching and the teaching effect.

5. Discussion and Conclusion

5.1. Core Research Findings

The collaborative adaptation of software is the key to improving teaching effect: A single software cannot cover the whole process of "principle explanation - practical training - achievement verification". The high-precision demonstration of Manim, real-time interaction of GeoGebra, simulation ability of Matlab, and data visualization function of Origin form complementary advantages, which can effectively support the differentiated needs of different teaching links.

Visualization teaching realizes dual improvement of "cognition - application": The results of the supporting questionnaire survey after teaching (valid sample n=300) show that 87.2% of students believe that animation demonstration helps them understand the parameter correlation of the model, and 79.5% of students can independently complete the "problem - model" matching. The statistical significance of competition results and the improvement of the proportion of high-level awards further confirm the simultaneous improvement of cognitive efficiency and application ability.

There are obvious differences in the adaptability of competition types: The number of awards in statistical modeling (NCSMC) and financial modeling (GBAFMMC) has increased most significantly (the growth rate is over 170%). These competitions focus on data processing and achievement presentation, so the advantages of visualization software are more likely to be exerted; while MCM/ICM involves interdisciplinary issues, so it is necessary to further strengthen the integrated teaching of software and interdisciplinary knowledge.

5.2. Research Limitations and Future Prospects

This study only selects samples from one university. In the future, it can be expanded to universities of different levels and majors to verify the universality of the teaching model; the existing research focuses on the macro competition result analysis, and in the follow-up, it can combine micro data such as software use frequency and learning time to deeply analyze the action mechanism of visualization teaching. In addition, the integrated application of AI tools and visualization software can be explored, such as using ChatGPT to realize real-time answers to model questions and intelligent recommendation of animation schemes, so as to further improve the personalization and efficiency of teaching.

References

1. R. Oliveira, E. A. Dias, R. Santos, V. Cotta-De-Almeida, J. Aguiar Coelho Nt, and L. A. Alves, "How math shapes the world of life science animation," *Frontiers in Computer Science*, vol. 7, p. 1520930, 2025. doi: 10.3389/fcomp.2025.1520930
2. T. Siregar, "Application of GeoGebra for teaching mathematics," *Preprint.*, vol. 648, p. v1, 2025. doi: 10.20944/preprints202510.0648. v1
3. C. Chen, "Design and Practice of Linear Algebra Visualization Teaching Platform Based on Deep Learning Algorithm," In *2025 2nd International Conference on Intelligent Computing and Robotics (ICICR)*, May, 2025, pp. 272-276. doi: 10.1109/icicr65456.2025.00055