Academe *Journal of Education & Psychology* *ISSN: 2249-040X*
Volume-16, Issue-1, (Special Issue: June 2026) *Peer Reviewed & Refereed Journal (IF: 6.25)*
PP: 37-45 *Journal Website:* www.academejournal.in

RETHINKING ASSESSMENT IN UNDERGRADUATE GRAPH THEORY COURSES WHEN STUDENTS HAVE ACCESS TO AI COMPUTATION: AN EMPIRICAL STUDY AND ETHICAL PROTOCOL

Jagadeesh R

Associate Professor
Department of Mathematics, Government First Grade College, Channapatna, Karnataka, India

ABSTRACT

Undergraduate mathematics students in Indian colleges can now compute topological indices, eigenvalues, and various graph energy formulations by entering a few lines of code into freely available packages. When a student can obtain the minimum dominating distance energy of a crown graph or the hyper-Zagreb index of a graphene sheet in seconds, the pedagogical question shifts from "how do we teach the computation?" to "what should we assess, and how?" This paper reports on a structured teaching experiment conducted across two academic semesters with final-year B.Sc. Mathematics students at a government college in Karnataka. In the control cohort, students solved graph energy and topological index problems entirely by hand. In the treatment cohort, students were given access to an open-source Python-based graph computation tool and were instead assessed on formulating conjectures, interpreting spectral properties, and constructing proof sketches. A comparison of pre-test and post-test scores, supplemented by qualitative interviews, reveals that conceptual retention improved in the treatment group when assessment focused on interpretation rather than calculation, while a subset of students exhibited uncritical acceptance of AI outputs. The paper proposes a four-component ethical protocol for the responsible integration of AI tools in undergraduate mathematics education.

Keywords: Mathematics pedagogy; Graph theory education; Ethical AI protocol; Topological indices; Graph energy; Assessment redesign; Undergraduate mathematics; Academic integrity; Indian higher education.

1. INTRODUCTION

The discipline of graph theory, once confined to abstract combinatorics seminars, has become a standard component of the undergraduate mathematics curriculum in Indian universities. Topics such as graph colouring, Eulerian and Hamiltonian circuits, adjacency matrices, and spectral properties now appear in syllabi prescribed by state universities across Karnataka, Tamil Nadu, Kerala, and Maharashtra. Alongside this curricular expansion, a parallel development has occurred in the computational landscape: open-source software packages—NetworkX in Python, SageMath, MATLAB's Graph Theory Toolbox, and specialised libraries such as MathGraph (Rajesh Kanna et al., 2024)—now make it trivially easy to compute quantities that once demanded hours of careful hand calculation.

Consider, for instance, the computation of the minimum dominating energy of a crown graph on 20 vertices. A student working by hand must first identify a minimum dominating set, construct the modified adjacency matrix, compute its 20 eigenvalues, and sum their absolute values. This process requires sustained concentration, fluency with matrix algebra, and careful arithmetic—skills that the traditional assessment model is designed to evaluate. The same student, using a three-line Python script, can obtain the answer in under two seconds. The availability of such tools does not merely change how students compute; it fundamentally alters what computation means in the pedagogical context.

This transformation has been accompanied by broader anxieties about AI in education. The emergence of large language models capable of generating proofs, explanations, and even research-quality mathematical content has prompted heated debate in academic circles (Borji, 2023; Cotton et al., 2024). However, much of this debate has focused on text-generation AI and its implications for essay writing, programming assignments, and examination integrity. Comparatively little attention has been paid to the specific impact of computational AI tools on mathematics education, where the relationship between computation and understanding is uniquely intimate.

This paper addresses that gap through an empirical study conducted at a government first grade college in Karnataka over two consecutive semesters (2024–25 and 2025–26). The study compares learning outcomes between a control cohort that relied exclusively on manual computation and a treatment cohort that used an open-source graph computation tool, with assessment redesigned to emphasise conceptual interpretation over mechanical calculation. The findings inform a four-component ethical protocol for AI integration in undergraduate graph theory instruction.

2. LITERATURE REVIEW

2.1 AI in Mathematics Education

The integration of technology into mathematics education has a long history, from the introduction of graphing calculators in the 1990s to the widespread adoption of computer algebra systems (CAS) such as Mathematica and Maple in university settings. Research has consistently shown that CAS use improves procedural efficiency but can reduce students' engagement with underlying concepts if not accompanied by pedagogical scaffolding (Thomas & Hong, 2005). More recently, Drijvers (2015) proposed an instrumental genesis framework in which students must develop both technical proficiency with the tool and conceptual understanding of the mathematics the tool implements.

The advent of AI-powered tools has accelerated this tension. Okonkwo and Ade-Ibijola (2021) surveyed chatbot applications in education and found that while AI tools increased student engagement, they also fostered dependency when used without structured pedagogical intervention. In mathematics specifically, Kasneci et al. (2023) argued that AI tools can serve as "cognitive partners" if instructors redesign assessment to focus on reasoning processes rather than final answers. However, empirical studies testing this proposition in the Indian higher education context remain scarce.

2.2 Graph Theory Pedagogy

Graph theory presents unique pedagogical challenges because it straddles the boundary between visual intuition and formal abstraction. Students readily grasp the idea of a network but struggle with the transition to algebraic representations such as adjacency matrices and their spectra (Gibson, 2012). The computation of topological indices and graph energies, which involves matrix construction, eigenvalue extraction, and algebraic manipulation, sits precisely at this boundary and is therefore an ideal context in which to study the impact of computational tools on learning.

In the Indian context, Rajesh Kanna et al. (2024) developed MathGraph, a Python package for computing graph energies and topological indices. While designed primarily as a research tool, MathGraph has pedagogical applications that have not been systematically studied. The present study uses MathGraph as the computational tool provided to the treatment cohort, thereby connecting a research-oriented software artefact to its potential educational uses.

2.3 Ethical Frameworks for AI in Higher Education

Several ethical frameworks for AI in education have been proposed. The UNESCO Recommendation on the Ethics of AI (2021) emphasises transparency, fairness, and human oversight. Prinsloo and Slade (2016) focused on learning analytics and proposed principles of proportionality and informed consent. More specific to assessment, Dawson (2020) argued that academic integrity policies must evolve to distinguish between prohibited use of AI (passing off AI-generated work as one's own) and permitted use (using AI as a declared tool within an assignment). The present study draws on these frameworks to construct a protocol tailored to the specific characteristics of mathematics education.

3. OBJECTIVES

The study pursues four objectives. First, to compare conceptual retention of graph energy and topological index concepts between students who compute manually and those who use an AI-based computational tool. Second, to identify patterns of uncritical acceptance of AI-generated outputs among tool-using students. Third, to document student perceptions of originality and academic integrity in the context of AI-assisted computation. Fourth, to propose an ethical protocol for AI tool use in undergraduate graph theory courses that is implementable within the administrative structure of state university-affiliated colleges in India.

4. METHODOLOGY

4.1 Research Design

A quasi-experimental pre-test–post-test design with a non-equivalent control group was employed. Random assignment was not possible because students were already allocated to class sections by the college administration. Section A (n = 38) served as the control group and Section B (n = 41) as the treatment group. Both sections were taught by the same instructor (the author) using identical lecture content, textbooks, and contact hours. The only difference was the mode of computation: Section A solved all problems by hand, while Section B was introduced to MathGraph in the fourth week and permitted to use it for all subsequent assignments and practice sessions.

4.2 Participants

Participants were final-year B.Sc. Mathematics students at a government first grade college affiliated with Bangalore University. The study spanned two semesters: Semester V (August–December 2024) and Semester VI (January–May 2025). Entry-level mathematical ability was assessed via a pre-test covering matrix algebra, eigenvalue computation, and basic graph theory concepts. Table 1 summarises the demographic and baseline characteristics of the two groups.

Characteristic	Section A (Control)	Section B (Treatment)
N	38	41
Female / Male	22 / 16	25 / 16
Mean age (years)	20.4	20.7
Pre-test mean (out of 30)	16.8 (SD = 4.2)	17.1 (SD = 3.9)

Prior Python experience	8 (21%)	10 (24%)

Table 1. Baseline characteristics of control and treatment groups.

An independent-samples t-test on pre-test scores confirmed no statistically significant difference between the groups ($t(77) = 0.34$, $p = 0.74$), indicating comparable baseline mathematical ability.

4.3 Intervention

The graph theory module covered: adjacency and distance matrices, graph energy (Gutman, 1978), minimum dominating energy (Rajesh Kanna et al., 2013), Zagreb indices (Gutman & Trinajstić, 1972), and topological indices of graphene (Parashivamurthy et al., 2019). The treatment group received a two-session (4-hour) workshop on Python basics and MathGraph installation in Week 4. From Week 5 onward, treatment students used MathGraph for all computational tasks but were required to submit written justifications for their results, including: (a) a description of the graph family and its parameters, (b) an interpretation of the computed eigenvalues in terms of graph structure, and (c) a conjecture about how the energy or index would change if the graph were modified.

The control group completed identical tasks by hand, submitting full computational workings. Both groups received identical fortnightly assignments (6 in total) and identical mid-semester and end-semester examinations.

4.4 Assessment Instruments

Three instruments were used. (i) A pre-test (30 marks) administered in Week 1, covering prerequisites. (ii) A post-test (50 marks) administered in the final week, divided into two sections: Section P (25 marks) testing procedural computation (compute the minimum dominating energy of a given graph) and Section C (25 marks) testing conceptual understanding (interpret, compare, conjecture, or prove properties of graph energies). (iii) A semi-structured interview protocol administered to a purposive sample of 12 students from each group (24 total) in the week following the post-test, exploring their experiences with computation, their understanding of the concepts, and their views on AI and academic integrity.

4.5 Data Analysis

Quantitative data were analysed using independent-samples t-tests and ANCOVA (with pre-test scores as covariates) in SPSS 27. Effect sizes were computed using Cohen's d. Qualitative interview data were transcribed, coded thematically using Braun and Clarke's (2006) six-phase framework, and analysed for recurring patterns across both groups.

5. FINDINGS

5.1 Quantitative Results

Table 2 presents the post-test results. On the procedural section (Section P), the control group outperformed the treatment group by a modest but statistically significant margin. On the conceptual section (Section C), the treatment group scored significantly higher. The overall post-test difference was not statistically significant, but the pattern of differential performance across sections is the central finding.

Section	Control Mean	Treat. Mean	t	p	Cohen's d
P (Procedural)	18.4 (3.6)	16.1 (4.3)	2.59	0.012	0.58

C (Conceptual)	14.2 (4.1)	17.8 (3.5)	-4.21	<0.001	0.94
Total (P + C)	32.6 (6.8)	33.9 (6.1)	-0.90	0.371	0.20

Table 2. Post-test performance: control vs. treatment group. Standard deviations in parentheses.

ANCOVA with pre-test scores as a covariate confirmed the pattern: Section C scores were significantly higher in the treatment group ($F(1,76) = 17.83$, $p < 0.001$, partial $\eta^2 = 0.19$) while Section P scores were significantly higher in the control group ($F(1,76) = 6.47$, $p = 0.013$, partial $\eta^2 = 0.08$). The larger effect size for conceptual understanding ($d = 0.94$, large effect) compared to procedural computation ($d = 0.58$, medium effect) suggests that the trade-off favours the treatment condition when conceptual learning is the primary objective.

5.2 Uncritical Acceptance of AI Outputs

A deliberate error was introduced into the MathGraph assignments in Week 10. Students were asked to compute the distance energy of a path graph P7, for which MathGraph returned a value affected by a known floating-point rounding issue (returning 14.9997 instead of the exact value 15.0). Of the 41 treatment-group students, 29 (70.7%) reported the value as 14.9997 without comment. Only 7 students (17.1%) noted the discrepancy and rounded to the correct integer value with a written justification. Five students (12.2%) identified the issue as a floating-point artifact and explained why it occurs. Table 3 summarises these responses.

Response Category	Count	Percentage
Reported AI output without comment	29	70.7%
Rounded with brief justification	7	17.1%
Identified floating-point artifact	5	12.2%

Table 3. Treatment group responses to a deliberately introduced floating-point discrepancy (n = 41).

This finding is pedagogically significant. It demonstrates that access to a computational tool, without explicit instruction in tool literacy and output verification, produces a form of automation complacency in which students treat the software’s output as authoritative rather than as a computation to be checked.

5.3 Qualitative Findings: Student Perceptions of Originality and Integrity

Thematic analysis of the 24 interviews yielded four major themes.

Theme 1: Efficiency versus understanding. Students in the treatment group consistently described the computational tool as “saving time” and “removing the boring parts,” enabling them to focus on “what the numbers mean.” However, several students in the same group expressed anxiety that they might not be able to perform the computations if the tool were unavailable, suggesting an awareness of potential dependency.

Theme 2: Ambiguity about originality. When asked “Is work that uses MathGraph your own original work?”, responses were divided. Approximately half of the treatment-group interviewees considered it original if they wrote the code and interpreted the results. The other half expressed uncertainty, with one student noting: “If the computer does the main work, I am not sure what part is mine.” No student in the control group raised similar concerns, as their computational process was entirely manual.

Theme 3: Desire for institutional guidance. Both groups expressed a desire for clear institutional rules about when and how AI tools may be used. Students reported that different instructors had different expectations, creating confusion. Several students specifically requested a written policy that distinguishes between permitted and prohibited uses of computational tools.

Theme 4: Conjecture formation as a new skill. Treatment-group students who had been required to formulate conjectures based on computed data described this as a novel and challenging experience. Several reported that this was the first time they had been asked to generate mathematical hypotheses rather than verify given statements, and that the experience deepened their engagement with the material.

6. PROPOSED ETHICAL PROTOCOL

Drawing on the quantitative and qualitative findings, and informed by the ethical frameworks reviewed in Section 2.3, the following four-component protocol is proposed for adoption by colleges affiliated with state universities in India. The protocol is designed to be implementable within existing administrative structures and does not require additional financial resources.

6.1 Component 1: Tool Use Declaration

Every assignment submission must include a declaration specifying: (a) which computational tools were used, (b) for which specific tasks, and (c) how the outputs were verified. The declaration should be a standard form appended to all submissions, analogous to the plagiarism declarations already required by most universities. This normalises tool use as a legitimate methodological choice while ensuring transparency.

6.2 Component 2: Proof-Obligation Requirement

When a student uses a computational tool to arrive at a result—such as a conjectured bound or a specific energy value—the assessment rubric must require an accompanying proof sketch, verification argument, or interpretive justification. Submitting a computed output without analytical engagement should receive at most partial credit. This requirement addresses the automation complacency observed in Section 5.2 by making critical evaluation of outputs a graded component of every assignment.

6.3 Component 3: Faculty Verification Checkpoints

At three designated points during the semester (Weeks 5, 10, and 14), the instructor should conduct brief oral or written spot-checks in which students are asked to explain a result they obtained computationally. These checkpoints serve a dual purpose: they deter submission of AI outputs without comprehension, and they provide formative feedback that helps students develop the habit of interpreting and verifying computed results.

6.4 Component 4: Assessment Rubric Redesign

The traditional assessment rubric, which awards marks primarily for correct computation, should be supplemented with explicit marks for: (i) interpretation of spectral properties (e.g., what does the distribution of eigenvalues tell us about the graph's structural symmetry?), (ii) conjecture formation (e.g., how would the energy change if an edge were added?), (iii) comparative analysis (e.g., why is the minimum dominating energy of a crown graph higher than that of a star graph with the same number of vertices?), and (iv) error identification (e.g., recognising when a computational output is likely incorrect). Table 4 proposes a sample mark allocation for a 25-mark assignment.

Assessment Component	Traditional	Proposed
Correct computation / output	20	8
Interpretation of results	3	6
Conjecture / proof sketch	0	5
Comparative analysis	0	3
Tool use declaration & verification	0	2
Presentation / clarity	2	1
Total	**25**	**25**

Table 4. Proposed reallocation of assessment marks from computation to conceptual engagement.

7. DISCUSSION

The central finding of this study—that conceptual retention improves when assessment is redesigned around interpretation rather than calculation, even as procedural fluency modestly declines—aligns with the instrumental genesis framework proposed by Drijvers (2015) and the cognitive partnership model of Kasneci et al. (2023). The large effect size for conceptual understanding ($d = 0.94$) suggests that the trade-off is educationally worthwhile, particularly in a curricular context where the ultimate goal is to prepare students for research or applied work in which computational tools will be ubiquitous.

The automation complacency finding (70.7% of treatment students reporting a floating-point error without comment) underscores that tool access without tool literacy produces a qualitatively different kind of ignorance. Students who compute by hand and make errors are aware that errors can occur; students who receive authoritative-looking decimal outputs from software may not develop this awareness unless it is explicitly cultivated. This finding has implications beyond graph theory: any discipline in which students use computational tools—statistics, physics, engineering—faces the same risk.

The qualitative finding that students desire institutional guidance is consistent with Dawson’s (2020) argument that academic integrity policies must evolve to address AI tools. The current situation, in which individual instructors make ad hoc decisions about permitted tool use, creates inconsistency and student confusion. The four-component protocol proposed in Section 6 is designed to address this gap at the departmental or college level, without requiring legislative action at the university level.

Several limitations should be acknowledged. The quasi-experimental design, while pragmatically necessary, limits causal inference. The sample is drawn from a single government college and may not generalise to elite institutions or private universities. The intervention period (two semesters) may be insufficient to capture long-term effects on learning habits. Finally, the floating-point error test assessed only one specific instance of tool literacy; a more comprehensive battery of tool-literacy assessments would strengthen the findings.

CONCLUSION

This study provides empirical evidence that the introduction of AI-based computational tools in undergraduate graph theory education can enhance conceptual understanding when accompanied by assessment redesign, but simultaneously creates risks of uncritical dependence on automated outputs. The proposed four-component ethical protocol—tool use declaration, proof-obligation requirement, faculty verification checkpoints, and assessment rubric redesign—offers a practical, implementable framework for managing these risks within the Indian higher education system.

The broader implication is that the question facing mathematics educators is not whether to permit AI tools, but how to structure their use so that computation serves understanding rather than substituting for it. The protocol proposed here represents one approach to this structuring; further research should test its effectiveness across multiple institutions, disciplines, and student populations.

As graph theory continues to grow in both pure and applied significance—finding applications in network science, molecular chemistry, and data science—the students we train today must be equipped not only to compute but to interpret, conjecture, and prove. AI tools can accelerate this journey, but only if we redesign the educational architecture to channel their power toward deeper mathematical thinking rather than shallower engagement.

REFERENCES

1. Borji, A. (2023). A categorical archive of ChatGPT failures. arXiv preprint, arXiv:2302.03494.
2. Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. Qualitative Research in Psychology, 3(2), 77–101.
3. Cotton, D. R. E., Cotton, P. A., & Shipway, J. R. (2024). Chatting and cheating: Ensuring academic integrity in the era of ChatGPT. Innovations in Education and Teaching International, 61(2), 228–239.
4. Dawson, P. (2020). Defending assessment security in a digital world: Preventing e-cheating and supporting academic integrity in higher education. Routledge.
5. Drijvers, P. (2015). Digital technology in mathematics education: Why it works (or doesn’t). In S. J. Cho (Ed.), Selected Regular Lectures from the 12th International Congress on Mathematical Education (pp. 135–151). Springer.
6. Gibson, J. P. (2012). Teaching graph algorithms to children of all ages. Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education, 34–39.
7. Gutman, I. (1978). The energy of a graph. Berichte der Mathematisch-Statistischen Sektion im Forschungszentrum Graz, 103, 1–22.
8. Gutman, I., & Trinajstić, N. (1972). Graph theory and molecular orbitals: Total π-electron energy of alternant hydrocarbons. Chemical Physics Letters, 17(4), 535–538.
9. Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., ... & Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. Learning and Individual Differences, 103, 102274.
10. Okonkwo, C. W., & Ade-Ibijola, A. (2021). Chatbots applications in education: A systematic review. Computers and Education: Artificial Intelligence, 2, 100033.
11. Parashivamurthy, H. L., Rajesh Kanna, M. R., & Jagadeesh, R. (2019). Some topological indices and their polynomials of graphene. Oriental Journal of Chemistry, 35(5), 1553–1563.
12. Prinsloo, P., & Slade, S. (2016). Student vulnerability, agency, and learning analytics: An exploration. Journal of Learning Analytics, 3(1), 159–182.
13. Rajesh Kanna, M. R., Dharmendra, B. N., & Sridhara, G. (2013). Minimum dominating energy of a graph. International Journal of Pure and Applied Mathematics, 85(4), 707–718.
14. Rajesh Kanna, M. R., Kumar, R. P., Vijayaraghavan, S., Kumar, A. K., & Nandappa, D. S. (2024). MathGraph: A Python package to compute energy and topological indices of graphs. COMPUSOFT: An International Journal of Advanced Computer Technology, 13(1), 4895–4902.

15. Thomas, M. O. J., & Hong, Y. Y. (2005). Teacher factors in integration of graphic calculators into mathematics learning. In H. L. Chick & J. L. Vincent (Eds.), Proceedings of the 29th Conference of the International Group for the Psychology of Mathematics Education (Vol. 4, pp. 257–264). PME.
16. UNESCO (2021). Recommendation on the Ethics of Artificial Intelligence. United Nations Educational, Scientific and Cultural Organization.