Manime: A Code-Driven Visual Teaching Method for Deep Learning Education

R. Raja Subramanian[1], Bharath Inukurthi[2]
Department of Computer Science and Engineering,
Kalasalingam Academy of Research and Education, Tamil Nadu, India
[1]rajasubramanian.r@klu.ac.in, [2]bharathinukurthi1@gmail.com

***Abstract*—This paper introduces Manime, a code-based animation teaching method for deep learning, and compares it to traditional chalk-and-board instruction using statistical analysis. While chalkboard teaching is static, Manime enables instructors to create repeatable, visually rich, and programmatically generated lessons; however, its effectiveness relative to traditional methods has not been systematically evaluated in classroom settings. We compared Manime-style instruction with chalk-and-board lectures in an introductory deep learning course and collected student feedback on comprehension, retention, engagement, and instructional preference. Using paired t-tests, one-sample t-tests, and chi-square tests on data from 60 students who experienced both formats, we found that comprehension improved by +2.77 points after the Manime animation (large effect size, d = 1.25), retention confidence was high (mean 8.28/10, very large effect size, d = 4.99), and engagement significantly favored Manime (Cramér's V = 0.43), with students also preferring animations for difficult topics (Cramér's V = 0.30). Students unanimously reported that visuals improved recall, and these findings align with multimedia learning theory and Dual Coding Theory, which suggest that combining visual and verbal channels enhances cognitive processing. Overall, the results indicate that Manime provides an effective and complementary teaching style to traditional chalk-and-board instruction for complex deep learning topics.**

***Keywords*—Deep Learning Education; Manim; Multimedia Teaching; Python Animation; Quality Education; Visual Learning**

ICTIEE* Track—*Emerging Technologies and Future Skills
ICTIEE Sub-Track—AI, Machine Learning, and Digital Tools in Education

I. INTRODUCTION

TRADITIONAL chalkboard-based lecturing often struggles to communicate the abstract and dynamic nature of deep learning concepts. Static drawings and verbal explanations require students to mentally visualize processes such as gradient flow, weight updates, and multilayer interactions, which can increase cognitive load and limit active understanding. In contrast, Manime a teaching approach that integrates anime-style visuals with mathematical explanations using the Python Manim library offers dynamic, code-driven animations that make abstract ideas more concrete and intuitive. Originally developed for the 3Blue1Brown channel, Manim enables instructors to construct precisely timed visual sequences of equations, shapes, and transformations, potentially supporting clearer conceptual learning.

Although Manim-based animations have become increasingly popular in online educational content, their effectiveness compared to traditional chalkboard teaching has not been rigorously evaluated in real classroom settings. This gap is particularly relevant in deep learning education, where students often struggle to build accurate mental models of algorithmic behavior. The present study therefore examines whether animation-based instruction using Manime can produce stronger comprehension, higher engagement, and better short-term retention than conventional chalkboard methods when teaching deep learning concepts. To address this, an experiment and student perception survey were conducted in a university deep learning course to directly compare the two instructional approaches.

II. BACKGROUND AND RELATED WORKS

The rapid expansion of artificial intelligence (AI) and multimedia learning technologies has reshaped how educators design and deliver instruction in STEM, computer science, and machine learning (ML) education. A growing body of research demonstrates that well-designed visualizations, animations, and interactive systems can significantly enhance learners' cognitive processing, motivation, and conceptual understanding. Early work in this domain highlights the power of personalization and visual scaffolding. Roozafzai and Zaeri (2024) showed that AI-driven adaptive systems can tailor content in real time to a learner's profile, while animations simultaneously support conceptual clarity.

Similarly, Ji and Zheng (2025) found that visual cues alone improve focus and retention, but when paired with pedagogical agents animated or humanlike guides they enhance engagement, intrinsic motivation, and deeper conceptual understanding. These findings align with Mayer's multimedia learning theory and the dual-channel processing model, which argue that coordinated visual and verbal input strengthens long-term memory formation. Other immersive technologies reinforce these results. Research on virtual reality (VR) demonstrates that richly constructed 3D environments promote constructive learning and long-term memory processes, a finding with direct relevance for immersive pedagogy and transdisciplinary STEM teaching (Ji & Zheng, 2025).

In distance education, Tugtekin and Dursun (2022) compared animated and interactive video formats, revealing that animations are particularly effective for representing complex concepts visually, while interactive videos maintain motivation through embedded quizzes and clickable elements. Their work reinforces the role of variety and interactivity as central to multimedia learning design. Within the machine learning and computer science domain, animation-driven tools have recently gained prominence.

Helbling and Chau (2023) introduced Manim ML, a framework leveraging the Manim engine to visually depict neural network architectures, parameter flows, and layer interactions effectively bridging abstract ML concepts with intuitive visual explanations. Extending this trajectory, Zhou et al. (2024) developed Manimator, a system that integrates large language models with Manim to automatically generate animations from natural-language descriptions. This approach lowers the technical barrier for educators lacking programming expertise and enables rapid production of high-quality animated instructional content. Foundational contributions from the Manim Community Development Team (n.d.) and popular creators such as 3Blue1Brown (2016) demonstrate how animation can transform complex technical subjects into intuitive narratives. The neural network series by 3Blue1Brown exemplifies how thoughtfully designed visual storytelling supports comprehension, offering evidence for the pedagogical power of animation-based content creation. Beyond animation, hands-on and experiential learning approaches further enhance understanding.

Hitron et al. (2018) introduced an experiential model for teaching children fundamental ML ideas through interactive, physical tasks and age-appropriate visual aids. Their findings show that embodied interaction builds intuitive conceptual foundations prior to formal coding or mathematical instruction. Similar strategies are recommended in higher education: Yadav and DeBello (2019) emphasize project-driven learning, visual aids, and interactive notebooks for teaching Python and data science, noting these methods help address diverse learning preferences in graduate-level classrooms. Recent developments continue to enrich the space. Berg et al. (2025) presented Manim-DFA, a Manim-based tool for animating topics in data-flow analysis and abstract interpretation areas traditionally difficult to grasp through text alone. Their results show that automated animations enable clearer understanding of compiler design fundamentals. Likewise, Riyantoko et al. (2025) proposed a self-paced Python-and-statistics framework integrating visual coding tasks and real-world datasets to support active experimentation, directly linking visual outputs with algorithmic behavior. Emerging conceptual perspectives also highlight the philosophical and interdisciplinary value of multimedia approaches.

Yu (2025) introduced the Wisdom Computing Perspective, which situates AI instruction within human-centered, reflective, and visually enriched learning environments. This framework underscores the growing convergence between multimedia learning, AI literacy, and cross-disciplinary education. Parallel research in collaborative learning reinforces these multimedia findings. Jeyanathan et al. (2025) demonstrated that the Student Teams Achievement Division (STAD) method enhances engagement, peer learning, and problem-solving performance in engineering contexts. Similarly, P et al. (2025) found that inquiry-based learning within team-based environments improves conceptual understanding and critical thinking in automata theory courses. When combined with animation-based tools such as Manim, these collaborative strategies promise even greater benefits pairing visual clarity with social learning processes known to strengthen comprehension and retention.

III. Proposed Approach for Effective Teaching

A. Manime

We define Manime (Math + Anime) as the practice of delivering mathematical or technical content through Manim-created animations. This concept is inspired by Grant Sanderson's 3Blue1Brown channel , which popularized Manim for educational videos. Manim is a Python library: users write scripts in a Scene class to construct Mobjects (mathematical objects like Circle(), Text(), or MathTex() for equations) and animate them. The community edition, ManimCE, is a stable fork with extensive documentation . For example, Sanderson's neural networks series uses Manim ("Neural Networks", 3Blue1Brown) to visually explain layer connections and matrix operations. By leveraging code, Manim makes it easy to update animations, highlight steps with color or motion, and synchronize narration. In summary, Manime combines rigorous mathematical content with custom animations to engage students visually.In our study , we developed animated instructional videos using Manim Community Edition (ManimCE), a Python based library that enables programmatic creation of mathematical animations.For each deep learning concept, we wrote custom scripts using Scene classes which contained mathematical objects (Mobjects) like MathTex for the equations we are displaying, Arrow for simulating the signal paths, and Dot objects to indicate where our neuron nodes were within a network. We looked after the animation behaviors with Manim's .animate property and AnimationGroup feature that allow for combined effects with combined movement changes including colors such as red arrows for forward propagation, blue for backpropagation transitions, or in a more general context matrix operations plus smooth timing to showcase the evolution of an object over time and through layers which we felt was useful in conveying concepts that were subject to complicated abstract computations.

B. Architecture of Manime

The architectural design of Manim is grounded in an object-oriented hierarchy that emphasizes modularity, code reuse, and extensibility. The class diagram illustrates the structural relationships among the three core components: Mobjects, Cameras, and Animations. These categories are structured around abstract base classes that define shared interfaces and attributes, which are then extended by specialized subclasses for specific behaviors and visual elements.

1) Mobjects Hierarchy

In Manim, the Mobject class serves as the core building block for every drawable element, offering the fundamental structure needed for visual components. It manages attributes such as points, which define geometric locations, and bounding_box, which specifies the object's spatial limits. This class also provides utility methods like add() for combining multiple objects and arrange() for organizing them within a scene.

Extending this foundation, the VMobject (Vectorized Mobject) class enables vector-based rendering and introduces customizable styling features, including fill_color, stroke_color, and stroke_width, ensuring visual consistency across all vector-rendered shapes. From VMobject, a variety of specialized classes emerge. For example, the Text class supports the creation of text elements with adjustable content and font size, while SVGMobject allows importing scalable vector graphics from file paths, making it ideal for adding intricate illustrations. Another notable subclass, Geometry, provides a template for creating geometric primitives. Derived from Geometry, the Dot class incorporates a radius attribute, and the Line class is defined by start and end coordinate points. This class hierarchy allows all objects to share a unified styling framework while preserving the unique behavior of each type.

2) Cameras Hierarchy

In Manim, the Camera class acts as the central engine for rendering, setting the stage for how every element appears on screen. It defines key parameters like frame_width and frame_height and includes functionality for determining the placement and scale of visual components. While it works well on its own, the class is intentionally built to be extended for more specialized rendering tasks.

Two widely used subclasses build on this foundation. The ThreeDCamera is equipped for creating depth-rich scenes, offering features such as focal distance adjustments to control how three-dimensional objects are perceived. The MovingCamera, on the other hand, enables fluid scene navigation panning across a layout, zooming in for emphasis, or even adjusting automatically with its auto_zoom capability. By structuring these capabilities as extensions of the base Camera, Manim ensures that advanced scene dynamics and perspective changes can be introduced without rewriting the essential rendering logic.

3) Animations Hierarchy

In Manim, all motion effects stem from the Animation base class, which establishes a unified structure for applying changes to Mobjects over time. This class defines core properties such as run_time, which controls how long the effect lasts, and the interpolate() method, which determines how an object transitions from one state to another.

From this foundation, Manim offers a range of built-in animation types. Transform morphs one object into another, reshaping its points and style. FadeIn gradually increases an object's opacity until it is fully visible. Write simulates the process of drawing an object step by step, often used for text or outlines. AnimationGroup allows several animations to be played simultaneously, enabling synchronized visual effects.

One of Manim's most streamlined features is the .animate syntax. Attaching .animate to a Mobject instantly creates the corresponding Animation object, allowing concise commands such as square.animate.shift(RIGHT). This approach keeps code readable and compact while preserving the flexibility and modular design of the underlying animation framework.

4) Design Significance

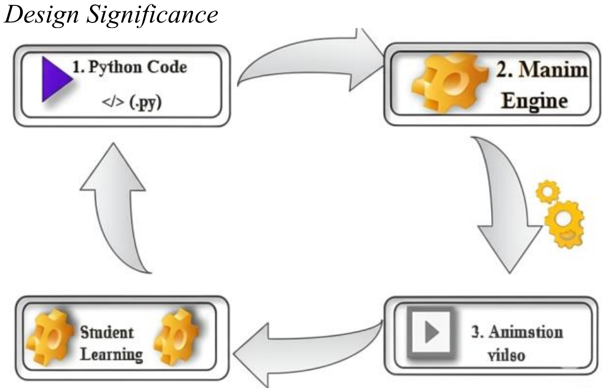
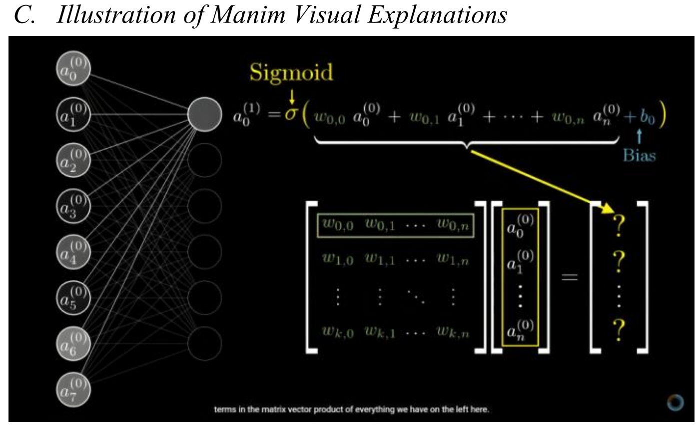


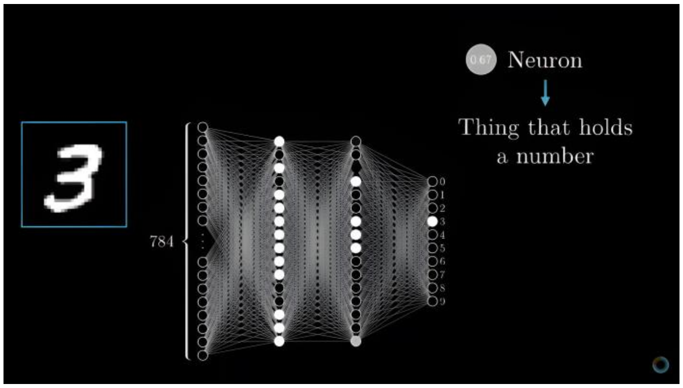
Fig.1. Manim Workflow for Classroom Teaching

Manim's design follows a strongly inheritance-based, extensible architecture, enabling developers to build new visual elements or behaviors simply by subclassing existing components. This approach minimizes repetitive code—for example, every vector-based object inherits styling features directly from VMobject and makes it straightforward to integrate custom elements into projects. By keeping responsibilities clearly divided among Mobjects, Cameras, and Animations, each subsystem can develop and improve independently, yet still operate seamlessly thanks to shared foundational interfaces. This structure is a key factor behind Manim's effectiveness in producing polished, educationally rich visual narratives, particularly in mathematics and computer science, where modular building blocks are combined into intricate, animated scenes with very little redundancy. This (Fig.1) shows how Manim- Based animations are generated from python scripts and then rendered into video files for classroom use. The code-driven nature allows flexible updates, layering and the real- time animation control.

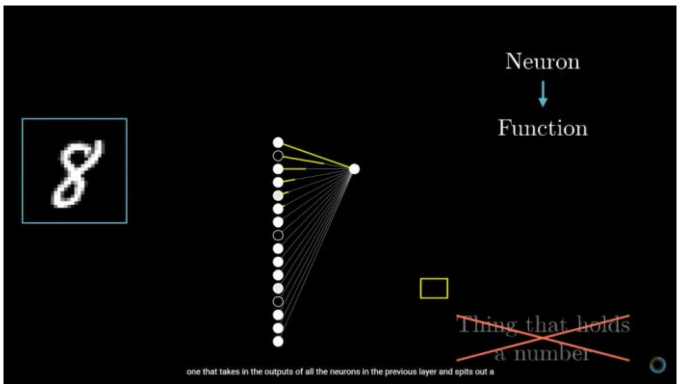
C. Illustration of Manim Visual Explanations



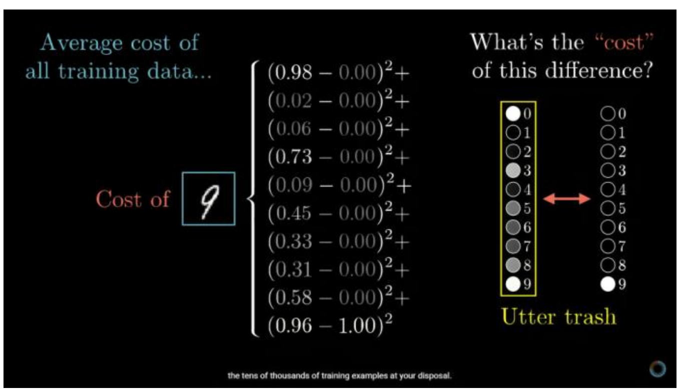
(a) Explaining complex calculations visually



(b) Comparing a neuron to number holder



(c) Explaining neuron as function



(d) Explaining the cost function role and intuition

Fig. 2. Manim Visual Explanation Example video[8]

The fundamental unit of the network, the neuron, is first introduced with a brilliantly simple analogy: "a thing that holds a number" (Fig. 2b). This grounds the concept in basic arithmetic. This initial simplification is later refined to a more accurate, yet still intuitive, definition of a neuron as a "function" that receives inputs from the previous layer and computes an output (Fig. 2c). In the animation, this progression is illustrated through color and motion: incoming connections highlight in yellow as inputs arrive, the neuron brightens briefly to signify computation, and the resulting output is animated as it flows into the next layer. These cues make the transition from "number holder" to functional unit perceptually clear.

To explain the concept of "learning," the video introduces the idea of a cost function as a tangible measure of the network's error on a training example. The animation shows the activations of the output layer for an incorrect guess, contrasts them with the ideal activations, and computes the sum of squared differences. A bracket overlay highlights each term in sequence, and the incorrect output neuron is outlined with a pulsing yellow emphasis to visually mark where the model failed. This makes the cost calculation easy to follow and conceptually grounded. Additional frames, such as the matrix–vector multiplication segment in Fig. 2(a), use animated movement of rows and columns to demonstrate how weights and inputs interact, transforming abstract algebra into a procedural visual story.

The video communicates deep learning concepts through progressive disclosure, beginning with intuitive analogies and gradually introducing mathematical formalism. This design choice makes otherwise abstract neural network mechanisms transparent and accessible, allowing learners to build understanding incrementally through both narrative and animation.

D. Survey Design

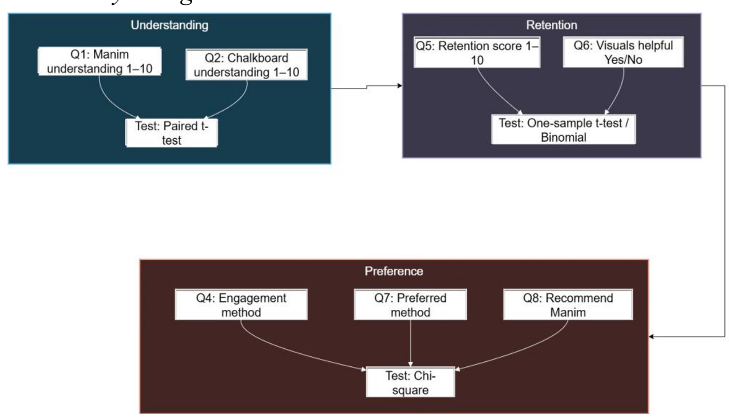


Fig. 3. Survey Questions Structure

To evaluate the effectiveness of Manim compared to traditional chalkboard-based instruction, we designed a structured survey (Fig. 3) focusing on three key dimensions: understanding, retention, and preference. The survey was distributed to participants after they were exposed to both Manim-generated animations and conventional chalkboard explanations of the same mathematical concepts.

The study assessed the clarity of instruction using a 1-10 Likert scale for conceptual comprehension and a traditional chalk and board approach. Long-term retention was measured using both quantitative and qualitative indicators, including self-assessed ability to retain the learned material and the impact of visual aids on memory reinforcement.

Students also expressed their preferred method for future instruction, particularly for abstract or mathematically complex topics. They also expressed their preference for using Manim animations for future teaching endeavors. A chi-square goodness-of-fit test was used to evaluate whether there were statistically significant differences in the students' preferences. The broader goal of the study was to examine how different instructional methods influence students' understanding and ability to retain the material.

IV. Results

All 60 students experienced both instructional methods in the same sequence a chalk-and-board lecture followed by a Manime-based animated explanation. A 20-minute spacing interval was included between the two sessions to minimize carryover effects, and students were not permitted to review materials during this interval. No pre-test was administered due to time constraints, but comprehension was measured through self-reported ratings before and after the Manime session. This within-subject design ensured that differences in responses were attributable to the instructional method rather than differences between student groups.

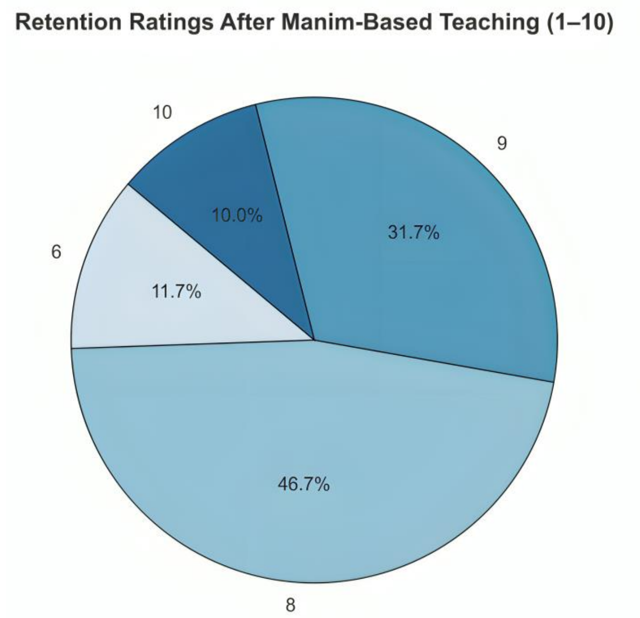
TABLE I
Hypothesis Tests and their Results

Test Type	p-value	Conclusion	Key Statistics
Paired t-test	0.000	Reject H_0: Manime significantly improved comprehension	Cohen's d = 1.2506 (large effect); Mean Difference = 2.7667; 95% CI = [2.1952, 3.3381]
One-sample t-test	0.000	Reject H_0: Retention rating significantly higher than 3	Cohen's d = 4.9884 (very large effect); Mean = 8.2833; 95% CI = [8.0097, 8.5569]
Chi-square goodness-of-fit	0.0013	Reject H_0: Manime was significantly more engaging	Cramér's V = 0.4286 (moderate–strong association)
Chi-square goodness-of-fit	0.0201	Reject H_0: Students preferred Manime for difficult topics	Cramér's V = 0.3 (moderate association)

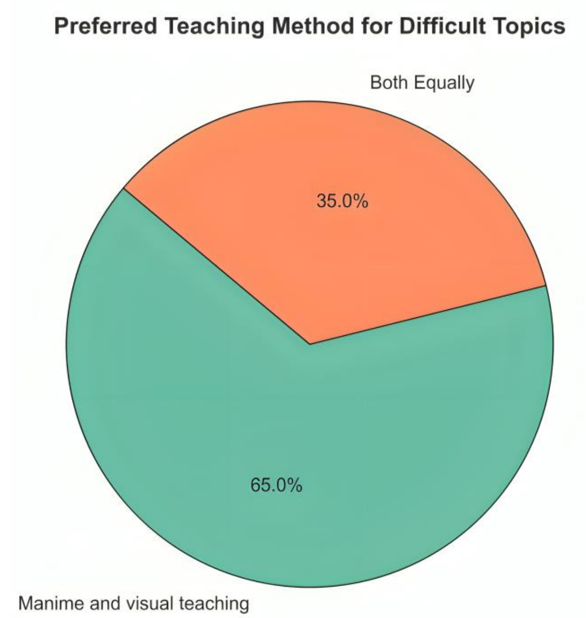
Table I summarizes the hypothesis tests and their results, with all analyses performed at a significance level of $\alpha = 0.05$. A paired t-test was used to compare students' comprehension following the Manim animation with their earlier chalk-and-board ratings. The test yielded a p-value of 0.000, indicating a significant improvement in understanding. The effect size was large (Cohen's d = 1.2506), with a mean difference of 2.7667 and a 95% confidence interval of [2.1952, 3.3381].

Retention confidence was assessed using a one-sample t-test against a neutral value of 3. The p-value of 0.000 showed that retention ratings were significantly higher than neutral. The effect size was extremely large (Cohen's d = 4.9884), with a sample mean of 8.2833 and a 95% confidence interval of [8.0097, 8.5569], indicating high consistency across participants.

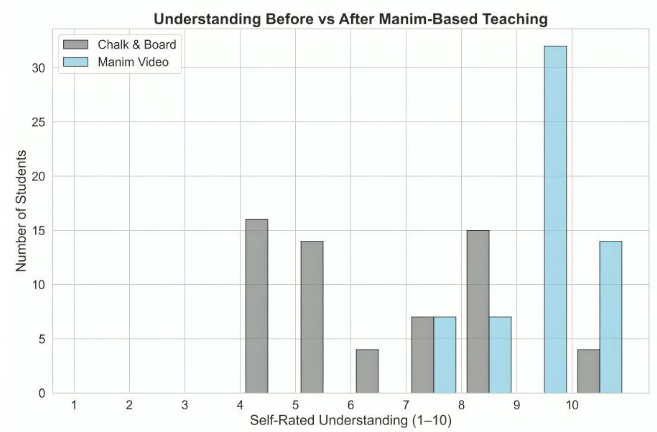
Engagement was evaluated using a chi-square goodness-of-fit test, which returned a p-value of 0.0013. This result indicates that students were not equally engaged by all methods and that Manim-based instruction was significantly more engaging. The effect size (Cramér's V = 0.4286) showed a moderate-to-strong association. A second chi-square test assessed students' preferred method for learning difficult topics and produced a p-value of 0.0201, with a moderate effect size (Cramér's V = 0.3), demonstrating a significant preference for Manim when dealing with challenging or abstract content.



(a) Self-assessed retention ratings after Manime-based teaching



(b) Preferred teaching method when learning Teaching difficult topic



(c) Self-rated understanding before and after Manime-based teaching.

Fig. 4. Survey results

The study conducted on 60 students found that Manim-based visual teaching was more effective than traditional chalk-and-board instruction in terms of instructional preference, comprehension improvement, and retention confidence. The results showed that 65% of students preferred (Fig.4b) Manim and visual teaching for complex or abstract topics, while 35% chose both equally. No student favored traditional chalk-and-board as a stand-alone method, indicating a strong shift towards animated instruction.

Qualitative responses showed that animations helped students connect more effectively with the content, and all respondents recommended Manim-style videos for future lectures. Comprehension levels improved dramatically after the animation, with 50% of students rating their understanding as 9, 21.4% as 10, and no student rated it below 7. The chalk-and-board session yielded a more scattered distribution, mostly between 4 and 7, confirming the statistically significant improvement (Fig 4c).

Retention confidence was also high following the Manim-based session (Fig 4a) , with 46.7% of students rating their ability to retain the topic as 8, 31.7% as 9, and 10% as 10. A one-sample t-test against a neutral retention benchmark score of 3 showed a highly significant result, suggesting that visual explanations had a substantial impact on students' confidence in remembering the material. A binary question asking whether animations aided recall received unanimous "Yes" responses from all 60 students, further supporting the visual method's effectiveness.

The results of this study strongly indicate that Manime-style visual and animated videos are a more effective pedagogical tool than traditional chalk and board lectures for teaching complex topics in deep learning. The most compelling finding is the statistically significant improvement in student understanding. The distribution of responses shifted dramatically from a scattered, moderate level of understanding with the traditional method to a highly concentrated, high level of understanding after the video. This suggests that the visual, step-by-step nature of the animation helped clarify abstract concepts that were previously difficult to grasp. The unanimous agreement that the visuals helped students connect with the topic reinforces this point; the animations are not merely decorative but serve as a crucial bridge to comprehension.

Furthermore, the data on engagement and preference is unequivocal. A significant majority of students found the Manime video more engaging and would prefer it for learning difficult subjects in the future. This is a critical factor, as higher engagement often correlates with better learning outcomes and improved motivation. While nearly a third of students found both methods equally engaging, the strong preference for Manime-style teaching, especially when anticipating difficult material, suggests that students perceive it as a more powerful learning aid.

Finally, the high likelihood of content retention reported by students is another key benefit. The combination of auditory narration and dynamic visuals appears to create a more memorable learning experience. The one-sample t-test confirms that this is not a neutral finding but a strong positive sentiment, suggesting the impact of the video extends beyond immediate understanding to longer-term recall. In addition to students' preference, Manim allows for significant technical advantages when teaching complicated deep learning material. For example, it can showcase the process of matrix multiplication by showing, in an animated and dynamic manner, the rows and columns moving as they multiply. Or, for forward propagation, weights and inputs can be color-coded and animated as they move through each neural layer, representing the dot product being computed with an animated object. In addition, in Manim, you can visualize activation functions by dynamically visualizing the input-output relationship immediately. Through this form of animated content, students can follow these abstract operations intuitively and have the ideas they are observing reinforced - such as weights, biases, and gradient flows in backpropagation. These animations also reduce cognitive load and help students move from pure theoretical understanding to practical understanding.

V. Discussion

The results of this study show that Manime-style programmatic animations offer clear advantages over traditional chalk-and-talk instruction for teaching deep learning concepts. Students demonstrated substantially higher comprehension after viewing the animated materials, suggesting that dynamic visualizations reduce cognitive load and make abstract processes such as gradient flow, forward propagation, and weight updates far more intuitive. Engagement and motivation were also noticeably higher, indicating that visually rich, step-by-step explanations help sustain attention and interest, which is critical for mastering mathematically dense topics. Retention confidence improved as well, supported by unanimous feedback that animations made the material easier to recall, reinforcing the idea that animation serves as a cognitive scaffold rather than a superficial enhancement.

Therefore, visual lessons like Manime increase comprehension, engagement, and long-term memory for students. Students strongly preferred Manime for complex topics and supported its continued use. In practice, Manime lessons can be reused by adjusting the input parameters or modifying small blocks of the animation script rather than recreating entire visual sequences from scratch. For instance, an instructor can adapt the same forward-propagation animation to demonstrate different activation functions by altering only a few lines of code controlling the output curve and color transitions. Network architecture diagrams can also be customized easily by modifying layer sizes, labels, or data-flow animations without redesigning the entire scene. However, this flexibility comes with constraints: developing new animations still requires familiarity with Python and Manim's scene-construction model, and more complex sequences may take considerable time to render. These practical considerations are important for instructors planning to integrate Manime into regular teaching workflows.

This work benefits the educational community by providing empirical evidence that programmable, code-driven animations can significantly strengthen conceptual understanding in advanced computing courses. Manime's reusability, precision, and ability to visualize complex operations offer instructors a modern and effective complement to traditional teaching methods. The major learning is clear: visual, dynamic instruction not only clarifies difficult concepts but also enhances engagement and long-term recall, making it a promising pedagogical tool for education.

These findings also align with multimedia learning theory, which suggests that people learn better from words and pictures than from words alone. This alignment can be further explained through Dual Coding Theory, which proposes that information is processed through complementary visual and verbal channels. In the Manime animations, narrated explanations are paired with synchronized visual cues such as highlighted connections, animated weight flows, and step-wise transformations. These features distribute processing across the visual and auditory channels, reducing the mental effort required to understand complex operations. The animations also support cognitive load theory by externalizing processes that learners would otherwise have to imagine mentally. For instance, matrix multiplication is shown through sliding rows and columns, while gradient flow is depicted through directional color changes. By reducing intrinsic and extraneous cognitive load associated with interpreting abstract equations, these animations help students devote more cognitive resources to building conceptual understanding.

LIMITATIONS

One limitation of this study is its reliance on self-reported ratings, which may introduce bias. Because animations can appear more novel, visually appealing, or entertaining than traditional chalk-and-board instruction, there is a possibility that the observed increases in engagement or interest may reflect a novelty effect rather than a sustained pedagogical advantage. Manime-based animations were new to them, which may also have influenced their responses. Although all participants had similar prior exposure to traditional instruction, their limited experience with animated teaching tools may have created an imbalance in familiarity between the two methods. Future studies should include objective performance measures and repeated exposure to animated instruction to reduce novelty-based bias.

Another limitation of this study is the homogeneity of the participant group. The sample consisted entirely of 60 undergraduate engineering students enrolled in a single deep learning course at one institution. As a result, the findings may not generalize to students from different academic backgrounds, age groups, or levels of prior experience. The subject matter itself neural networks and deep learning is highly mathematical and visually intensive, which may naturally benefit more from animated explanations than other topics. Therefore, the effectiveness of Manime may vary when applied to courses involving less visual structure, such as ethics, theory of computation, or non-technical subjects. Future research should include more diverse participant demographics and evaluate Manime across multiple course types, institutions, and proficiency levels to assess broader applicability.

Although students reported high confidence in their ability to retain the concepts immediately after the Manime-based explanation, this study did not examine long-term retention. The survey captured only short-term, self-reported retention rather than objective, delayed performance. As a result, it remains unclear whether the learning gains associated with Manime persist over time or diminish once the novelty and immediacy of the animation fade. Future research should incorporate delayed post-tests administered days or weeks after instruction to evaluate the durability of retention. Prior learning science studies emphasize that spaced assessments and longitudinal testing provide a more accurate measure of lasting conceptual understanding, and integrating such methods would offer a more reliable evaluation of Manime's long-term instructional impact.

CONCLUSION

The current study demonstrates statistically significant evidence that a code-driven, animation-based teaching approach, Manime, is more effective than traditional chalk and board instruction for teaching concepts pertaining to deep learning. Instructors can create lessons that can be reused and customized by taking advantage of the programmable capabilities of the Manime library. Therefore, visual lessons like Manime increase comprehension, engagement, and long-term memory for students. Students strongly preferred Manime for complex topics and liked its continued use. This study demonstrates the potential benefit of utilizing programmable visual tools in 21st-century technical education because it addresses the disconnect between abstract theory and intuitive understanding.

REFERENCES

3Blue1Brown. (2016). Neural networks [YouTube playlist]. YouTube. https://youtube.com/playlist?list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi

Barut Tugtekin, E., & Dursun, O. O. (2022). Effect of animated and interactive video variations on learners' motivation in distance education. Education and Information Technologies, 27(3), 3247–3276.
Berg, L., Yernaux, G., Vandeloise, M., & Vanhoof, W. (2025). Manim-DFA: Visualising data flow analysis and abstract interpretation algorithms with automated video generation. In Proceedings of the 17th International Conference on Computer Supported Education (CSEDU) (pp. 604–611).
Gijsen, M., Catrysse, L., De Maeyer, S., & Gijbels, D. (2024). Mapping cognitive processes in video-based learning by combining trace and think-aloud data. Learning and Instruction, 90, 101851.
Helbling, A., & Chau, D. H. (2023). ManimML: Communicating machine learning architectures with animation. arXiv Preprint, arXiv:2306.17108.
Helbling, D., & Chau, D. (2023). MLVis: A high-level animation library for machine learning education. arXiv Preprint, arXiv:2305.11099.
Hitron, T., Wald, I., Erel, H., & Zuckerman, O. (2018). Introducing children to machine learning concepts through hands-on experience. In Proceedings of the 17th ACM Conference on Interaction Design and Children (pp. 563–568).
Ji, X., & Zheng, X. (2025). Can pedagogical agents and visual cues enhance students' deep learning in a virtual reality environment? Insights from multiple data analysis. Education and Information Technologies, 1–23.
Jeyanathan, J. S., P, N., & Nair, A. (2025). Assessment of learning: Student Teams Achievement Division technique for empowering students in problem-solving courses. Journal of Engineering Education Transformations, 38(2), 107–113).
Mahardhika, I. E. P., Prayitno, H. J., Indri, I., & Fitriyan, M. R. (2025). Visual and contextual learning for deep learning education: A unified tool for theory–practice integration. Journal of Deep Learning, 69–80.
P, N., M, R., Jeyanathan, J. S., & Muneeswaran, V. (2025). Effect of inquiry-based learning on engineering students as collaborative approach in problem-solving and research for formal language automata course. Journal of Engineering Education Transformations, 38(2), 23–29.
Riyantoko, P. A., Funabiki, N., Brata, K. C., Mentari, M., Damaliana, A. T., & Prasetya, D. A. (2025). A fundamental statistics self-learning method with Python programming for data science implementations. Information, 16(7), 607.
Roozafzai, Z. S., & Zaeri, P. (2024). Digital dynamics: Exploring the intersection of AI, animation, and personalized learning. i-Manager's Journal of Educational Technology, 21(1).
Wunderlich, L., Higgins, A., & Lichtenstein, Y. (2021). Machine learning for business students: An experiential learning approach. In Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1 (pp. 512–518).
Yadav, N., & DeBello, J. E. (2019). Recommended practices for Python pedagogy in graduate data science courses. In 2019 IEEE Frontiers in Education Conference (FIE) (pp. 1–7). IEEE.
Yu, T. (2025). Design description of Wisdom Computing Perspective. arXiv Preprint, arXiv:2505.03800.
Zhou, H., Tang, A., & Chan, S. H. (2024). Manimator: Text-to-animation generation using large language models and Manim. arXiv Preprint, arXiv:2403.06732.
Jeyanathan, J. S., P, N., & Nair, A. (2025). Assessment of Learning: Student Teams Achievement Division Technique for Empowering Students in Problem-Solving Courses. Journal of Engineering Education Transformations, 38(2), 107–113.
P, N., M, R., Jeyanathan, J. S., & Muneeswaran, V. (2025). Effect of Inquiry-based Learning on Engineering Students as Collaborative Approach in Problem-Solving and Research for Formal Language Automata Course. Journal of Engineering Education Transformations, 38(2), 23–29.