Gamified Teaching Scenario Simulation System Built Based on Unity Engine and Particle System

Ling Fan[1], Yang Yang[2]
[1]College of Design and Art, Shanghai Technical Institute of Electronics & Information, Shanghai 200120, China
[2]Digital Media Art Program, College of Art, Shanghai Zhongqiao Vocational and Technical University, Shanghai 201514, China

Corresponding author: Yang Yang (e-mail: FL0514@126.com)

ABSTRACT Real-time semantic visualization plays an increasingly important role in immersive training environments and interactive engineering simulations, where stable dynamic rendering is also valuable for the visualization of electromagnetic field evolution and propagation processes. This study presents a gamified teaching scenario simulation system based on the Unity engine and a particle system, integrating learner behavior modeling, exponentially weighted moving average (EWMA) smoothing, semantic parameter mapping, and GPU-driven parallel rendering into a closed-loop visual feedback framework. A ScriptableObject-based semantic mapping strategy dynamically associates learning states with particle attributes, while ComputeBuffer and Visual Effect Graph enable efficient real-time visualization with consistent temporal behavior. Experimental results demonstrate that the proposed system maintains frame rates between 56.3 and 59.8 fps with an average feedback latency of 45.57 ms, significantly improves semantic mapping consistency, and enhances user immersion and interaction satisfaction across multiple teaching scenarios. By providing temporally coherent and semantically adaptive visual feedback, the proposed framework offers a scalable solution for intelligent visualization systems and provides methodological support for interactive representation and analysis of complex electromagnetic phenomena, virtual engineering environments, and digital training platforms.

INDEX TERMS unity engine, particle system, semantic visualization, real-time rendering, electromagnetic visualization

I. INTRODUCTION

THE intelligent design of gamified teaching scenario simulations promotes the deep integration of education and visual technology. This approach offers a powerful methodology, directly applicable to the industry, by facilitating immersive learning for tasks like design review or complex machine maintenance through the creation of highly realistic, interactive, and visually-driven simulated training environments. As a key mechanism to carry out dynamic visual feedback, particle systems have a decisive effect on learner engagement and scene credibility. However, their accuracy of performance and consistency of interaction still face performance bottlenecks under high-concurrency interaction conditions [1,2]. Improving the coupling degree between semantic mapping and real-time feedback is of great significance for building highly realistic teaching scenarios [3,4]. Improving this coupling requires aligning state changes with parameter updates so that visual responses match user actions without temporal drift. The semantic visualization capability of teaching content has become a key indicator of intelligent instructional design [5,6]. Current research still faces challenges in high-density particle scenes, including response delay and interframe discontinuities. Particle emission mechanisms are constrained by fixed-parameter models, lacking dynamic adjustment of semantic states [7-9]. While most rendering pipelines can improve frame rates under multi-threaded computation, they weaken the temporal stability of particle swarms [10-12]. The semantic and visual offset in real-time feedback leads to a misalignment between the perception of learning behavior and system performance [13-15]. In recent years, particle generation and visual effect control methods have continued to deepen in terms of semantic constraints, temporal mapping, and spatial consistency. Hierarchical particle generation strategies improve the spatial distribution structure of multimodal particles [16,17]; rendering frameworks based on graphics pipeline fusion enhance pixel response accuracy in dynamic scenes [18,19]; the application of dynamic smoothing models improves the temporal transition continuity of particle swarms [20,21]. However, these methods still rely on local parameter adjustment or single-scale control mechanisms, failing to achieve real-time coupling updates between the semantic and visual layers [22-24], making it difficult to maintain highly credible dynamic feedback in gamified teaching scenario simulations.

To address the aforementioned issues, this paper proposes a semantic-based visualization method. This method utilizes the Unity engine and employs a highly parallel rendering structure designed using Visual Effect Graph and ComputeShader. The system tracks learner actions using C# scripts, generates mastery vectors, and smooths semantic state transitions through the EWMA algorithm. This allows for a stable dynamic closed-loop between learning behavior, semantic features, and visual feedback using a ScriptableObject mapping table, enabling dynamic changes in particle parameters. Structurally, this method achieves a consistent temporal rhythm in particle responses, ensuring both semantic accuracy and visual stability. It provides a scalable implementation framework for visualization design in gamified teaching scenarios.

II. SYSTEM METHOD DESIGN

The system approach is to build a dynamic visualization framework based on gamified task-driven learning, which achieves multi-layered coupling of data, semantics, and visual feedback. The task completion status, skill level, scores, and feedback can be mapped in visual space. The overall framework is organized into three layers: data acquisition and behavior modeling, semantic and parameter mapping, and GPU-based dynamic rendering. Visual feedback creates a closed-loop in which the behavioral state of the learner can be visualized semantically in the virtual environment. The overall architecture of the system is illustrated in Figure 1.

This design revolves around a semantically driven gamified particle system, constructing a dynamic closed-loop between the task operational and achievement trigger activity and visual reward feedback. This system measures and models behavior, maps game states to semantic features, and parallel processes through GPUs particle rendering and reward feedback to create a visual loop between operation, challenge, and achievement. The three-layer design retains consistency across the temporal and semantic axes, creating a stable mapping between the temporal characteristics of learning behavior with particle behavior, thus achieving a self-consistent semantic visualization process that can operate effectively in virtual learning spaces.

A. DATA ACQUISITION AND BEHAVIORAL MODELING

Learners' actions in the virtual teaching environment are recorded in time-series format. Let the original event stream be denoted as $E = \{e_t \mid t = 1, 2, \dots, T\}$, and each event consist of three parts: action type, location coordinates, and response duration. To eliminate noise caused by differences in operation frequency and device performance, the event vectors are normalized to obtain a standardized sequence:

$$x_t = \frac{e_t - \mu_E}{\sigma_E} \tag{1}$$

In Formula (1), μ_E is the event mean, and σ_E is the standard deviation, used to stabilize the consistency of the statistical distribution of each sample. The system divides the interaction process into the fixed-length time window ΔT. Let the event set in the k-th window be $E_k = \{x_t \mid t_k \le t < t_k + \Delta T\}$, and the behavior state vector within the window is calculated:

$$b_k = \frac{1}{|E_k|} \sum_{x_t \in E_k} W x_t \tag{2}$$

In Formula (2), W is the task weight matrix. Response weights are assigned according to operation type and task stage, and the comprehensive learning state vector for that time slice is output to describe the changes in integral accumulation, skill improvement, and challenge completion. The state sequence $\{b_k\}$ is mapped to a global state matrix after time-series correlation modeling:

$$S = [b_1, b_2, \dots, b_K]^\top \tag{3}$$

The matrix generated by Formula (3) describes the dynamic trajectory of the learning process in the time dimension, with each row corresponding to an interaction mode in a local learning stage. The system calculates the state vector based on S:

$$s_k = f(W_s b_k + \varepsilon) \tag{4}$$

In Formula (4), $f(\cdot)$ is the nonlinear activation function; W_s is the state projection matrix; ε is the modeling error term. After this mapping, s_k has strong representation consistency and is used for particle parameter generation in the subsequent semantic feature mapping stage [25,26]. The entire modeling process is completed in real time in the C# script layer. By combining event flow and matrix calculation, the temporal features of learner operations have stable numerical behavior in the normalized feature space, thereby eliminating common interactive delays and feedback discontinuities in virtual teaching environments and providing structured input for the dynamic visualization link of the system [27,28]. This ensures that the temporal patterns extracted from the operation sequence remain consistent across different interaction frequencies and device conditions, thus providing reliable input for subsequent parameter mapping.

B. SEMANTIC FEATURE MAPPING AND PARAMETER GENERATION

The learning state vector sequence $\{s_k\}$ obtained from Formula (4) is smoothed in the time dimension to enhance temporal coherence in the transition of semantic states and to prevent instability of particle behavior and visual flickering caused by abrupt fluctuations in the raw sequence. To suppress this noise disturbance and extract persistent semantic trends, an exponentially weighted moving average model is applied to smooth the state sequence [29,30]. This smoothing process introduces a controlled temporal lag that

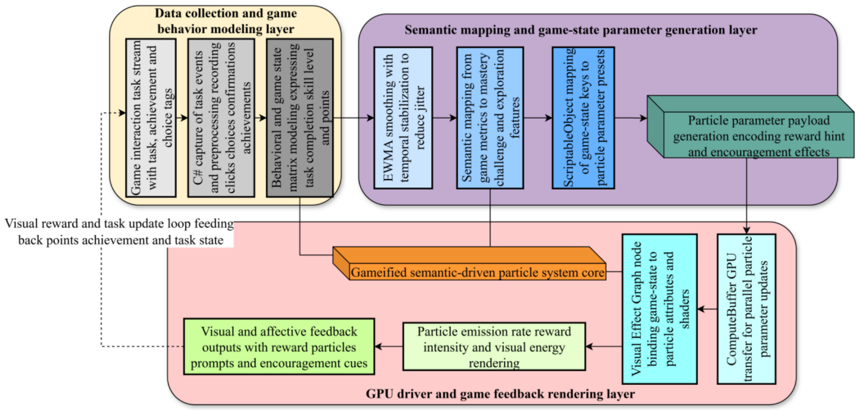

FIGURE 1. Overall architecture diagram of the gamified semantic visualization system method

preserves the continuity of state evolution, ensuring that the visual response reflects a stable semantic progression rather than momentary fluctuations. The smoothed mastery vector is defined as:

$$\hat{s}_k = \alpha s_k + (1-\alpha)\hat{s}_{k-1} \tag{5}$$

In Formula (5), $\alpha \in (0,1]$ is a smoothing factor used to adjust the proportion of influence between immediate behavior and historical task state in the mastery calculation. A suitable selection of α yields a balance between responsiveness and temporal stability, ensuring that the evolution of semantic states aligns with the perceptual expectation of continuous presence within the visualization process. A larger α emphasizes the current operational response, while a smaller α maintains the inertial continuity of the learning process. Through iterative updates, the system obtains a smooth sequence that converges in the time domain and is insensitive to local anomalies [31,32]. To adapt this sequence to the teaching scenario parameters, a semantic mapping function is constructed:

$$p_i = M_i(\hat{s}_k, c) = \beta_i \cdot \tanh(R_i \hat{s}_k + \delta_i c) \tag{6}$$

In Formula (6), p_i represents the attribute parameter of the i-th type of particle; $M_i(\cdot)$ is the corresponding feature mapping relationship; R_i is the projection matrix from semantics to physical parameters; c is the teaching scenario context tensor; β_i and δ_i are the scaling and coupling coefficients, respectively, used to control the magnitude of semantic influence and environmental adaptability. This mapping is registered in the ScriptableObject table in key-value form to achieve efficient indexing and dynamic updating of the parameter set. The teaching scenario context tensor c is defined as a three-dimensional continuous vector derived from task progression, interaction density, and recent rendering load. These values come from the task weight matrix, the normalized event frequency, and the cumulative variation of particle parameters. The tensor c adjusts the projection matrix in Formula (6) so that parameter updates match the operational and visual characteristics of the running scenario. Once all parameter vector sets $P = \{p_i\}_{i=1}^n$ are generated, the particle control set is formed through matrix combination:

$$\Phi = G(P) = \sum_{i=1}^{n} \omega_i p_i \tag{7}$$

In Formula (7), ω_i represents the built-in attribute weight coefficients, which are automatically adjusted by the visual consistency function to maintain energy conservation and color mapping balance among different particle attributes. This parameter set is updated synchronously with $\hat{s}_k$ on the time axis to ensure stable transitions in parameter updates during state smoothing. The tensor c is updated at the same sampling step as $\hat{s}_k$ so that its three components remain aligned with the temporal evolution of the learning state, maintaining consistency in the mapping process without introducing discontinuities.

C. PARTICLE SYSTEM DRIVEN AND DYNAMIC RENDERING

After smoothing is completed, the particle control set Φ generated according to Formula (7) is transferred to the GPU through ComputeBuffer during the rendering stage to achieve parallel computation and maintain the synchronization between semantics and vision. Each particle instance corresponds to a state vector $v_j = [x_j, y_j, z_j, \eta_j]$ in the computation kernel, where the position component (x_j, y_j, z_j) is determined by the dynamic parameters of the control set, and the energy term η_j reflects the spatial mapping of semantic intensity [33,34]. To ensure the smooth evolution of particle distribution, a gradient decay-based velocity update function is adopted:

$$v_j^{t+1} = v_j^t + \lambda_1(\Phi - v_j^t) + \lambda_2 \nabla\Psi(v_j^t) \tag{8}$$

In Formula (8), λ_1 is the semantic attraction coefficient, which controls the convergence rate of particles towards the target parameter field; λ_2 is the weight of the local gradient term, which suppresses particle aggregation in high-density regions through the potential function $\Psi(\cdot)$. This achieves a stable spatial distribution of particle density. As a result, brightness and diffusion remain within predictable ranges, ensuring continuous during rapid updates of particle attributes. The updated state is then processed by parallel normalization and fed into the Visual Effect Graph node network to generate temporally continuous emissivity and color sequences. To maintain inter-frame consistency, the system calculates the particle emission function on the GPU:

$$L_t = \gamma_1 \sum_{j=1}^{N} \eta_j^t + \gamma_2 \left\| v_j^{t+1} - v_j^t \right\|_2 \tag{9}$$

In Formula (9), L_t is the luminous intensity at time t, and γ_1 and γ_2 respectively adjust the weights of semantic energy and motion smoothing terms to maintain physical consistency in brightness and motion in the rendering output. The result calculated by this function directly drives the lighting and velocity attributes of the particle emission module.

During the rendering loop, the system updates the particle attribute matrix $R_t = [v_1^t, v_2^t, \ldots, v_N^t]$ based on the frame synchronization signal and linearly merges it with the smoothed state vector $\hat{s}_k$.

$$F_t = \theta R_t + (1-\theta)\hat{s}_k \mathbf{1}^\top \tag{10}$$

In Formula (10), θ is the inter-frame fusion coefficient, and $\mathbf{1}$ is a unit column vector with consistent dimension, used to extend the semantic state to the particle swarm scale and realize the continuous constraint of global semantics on local rendering. The fusion matrix F_t is transformed into particle instance attributes through the parameter binding layer of Visual Effect Graph to ensure that the particle flow maintains continuity and hierarchy in response to changes in learning state on the time axis. Through this GPU-driven particle dynamics mechanism, the system establishes a real-time closed-loop between semantic mapping and visual feedback, so that gamified states including score growth, level shifts, and challenge completion create corresponding light effect changes and particle rewards across the visual layer contributing to the stable frame-to-frame transitions and coherent visual response during user operations so that the visual output reflects the user's actions with minimal temporal fluctuation, maintaining coherence across successive updates of particle behavior [35].

D. EXPERIMENTAL ENVIRONMENT AND SYSTEM VERIFICATION

The experiment is carried out in a Windows 11 environment using Unity 2022.3 LTS, with a hardware configuration of an Intel i9-13900K, RTX 4090, and 64 GB of memory. This configuration provides a wide performance margin for parallel rendering, and the timing characteristics observed in the experiments reflect the bandwidth and execution range of this platform. Rendering and computation are paired in the parallel, real-time rendering approach applied here to assess rendering performance under gamified task-driven conditions. The research system simultaneously employs the DirectX 12 graphics pipeline to drive the ComputeBuffer and Visual Effect Graph. The script layer employs C# to facilitate event acquisition and parameter updates at a frequency of 120 Hz. The research team has designed seven learning situations: cognitive guidance, operational feedback, concept demonstration, collaborative exploration, scenario simulation, evaluation feedback, and emotion-driven learning. Task points, level advancing, and visual rewards serve as the flow of interaction within each of the learning situations. Performance and constancy are evaluated across the scenarios within increased complexity. The EWMA smoothing method is utilized to examine frame rate, GPU utilization, and feedback delay. The feedback stability of the system in task feedback is evaluated in the gamified scenarios using the plotted correlations between state vectors and particle attributes.

III. RESULTS

A. SYSTEM PERFORMANCE AND REAL-TIME

The experiment focuses on the real-time performance of the particle rendering pipeline, revealing the differences in the temporal structure of parallel computing, semantic constraints, and smoothing strategies in multi-scene interactions by comparing four different rendering mechanisms. The traditional CPU-Shuriken system relies on the main thread to update particle parameters, exhibiting a typical serial bottleneck; the No-Mapping mechanism uses the GPU parallel pipeline but lacks semantic mapping, resulting in unstable state constraints on particle behavior; the system with EWMA disabled already possesses high temporal consistency given semantic projection and GPU acceleration; with EWMA enabled, exponential weights suppress fluctuations in the temporal dimension, making the transition between particle emission and state sequences smoother. Seven teaching scenarios—covering cognitive guidance, operational feedback, concept demonstration, collaborative inquiry, situational simulation, evaluation feedback, and emotion-driven learning—exhibit different interaction densities and parameter update frequencies. These differences place varying pressures on the rendering pipeline and provide a comprehensive testbed for evaluating the temporal stability of each mechanism. The results are shown in Figure 2.

Under the traditional CPU-Shuriken benchmark, the frame rate fell within the 43–48 fps range across multiple scenarios, with the lowest value of 43.5 fps in the scenario simulation, where the feedback latency reached 82 ms, indicating structural blocking of the main thread during high-load interaction phases. The no-Mapping system maintained a frame rate of 50–55 fps, with a latency of 63 ms in the scenario simulation. This performance improvement stemmed from parallel rendering, but due to the lack of semantic constraints, the frequency of particle parameter jumps between adjacent frames increased, causing the GPU to remain in a state of high-amplitude recalculation for extended periods.

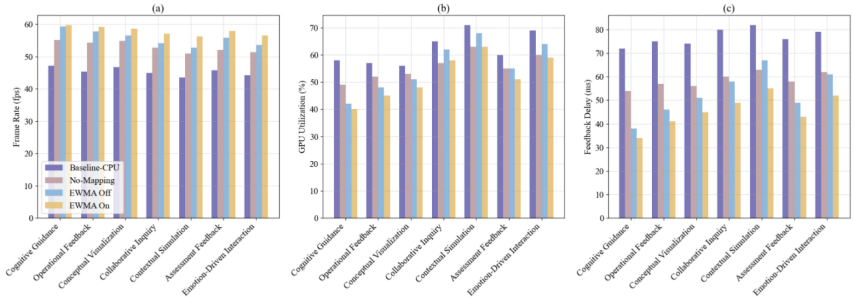

FIGURE 2. Real-time performance comparison chart of multiple scenes under different rendering mechanisms. (a) Average frame rate; (b) GPU utilization; (c) Feedback delay

The semantic mapping framework, with EWMA disabled, stabilizes the parameter update rhythm and constrains delay variations around 67 ms, maintaining a steady distribution of frame rate values across scenes. After enabling EWMA, the feedback delay in the scenario simulation converges around 55 ms with reduced fluctuation, the frame rate reaches 56.3 fps, and GPU utilization settles at 63%. The reduction in utilization is influenced by the computational capacity of the hardware platform, and the smoothing operation interacts with the bandwidth of the parallel pipeline in a way that depends on device-level throughput characteristics. The suppression of abrupt state gradients through exponential smoothing produces a more uniform rhythm when writing parameters to the ComputeBuffer and reduces temporal jitter during rapid state transitions. The comparative results shows that rendering parallelization and semantic constraints are the primary sources of performance improvement, while the smoothing mechanism corrects temporal perturbations and sustains a consistent update pattern across multiple teaching scenarios. The interaction between the smoothing process and the rendering pipeline varies with the processing range of different GPU architectures, and the balance between transfer operations and local updates may shift when the computational margin of the device narrows.

B. SEMANTIC MAPPING ACCURACY

In semantic visualization modeling, the correlation between particle parameters and the learning state dimension directly determines the semantic expression quality of visual feedback. To compare the mapping features under different driving mechanisms, this study constructs two types of systems: the traditional Unity Shuriken particle system, driven directly by Central Processing Unit parameters, lacks smoothness and semantic constraints; the improved system, based on EWMA smoothing and the ScriptableObject mapping table, drives particle behavior in parallel through ComputeBuffer, achieving semantic control of parameter updates. The experiment uses the learning state vector S_1–S_5 (attention, operation frequency, exploration depth, cognitive load, mastery) and particle parameters P_1–P_6 (emissivity, initial velocity, brightness, hue, diffusivity, lifespan) to calculate their correlation coefficient matrix, comparing the semantic mapping accuracy of the two systems. The results are shown in Figure 3.

The traditional system has an average correlation of 0.55, indicating a loose correlation between states and parameters. Low-correlation terms (below 0.50) account for approximately 30.0%, with a correlation coefficient of 0.67 between emissivity and operation frequency, and 0.69 between brightness and mastery, resulting in weak overall consistency. The improved system increases the average correlation to 0.65, and the number of high-correlation terms (above 0.70) increases from 0 to 11 pairs. Operation frequency and emissivity reach a correlation of 0.85, while brightness and mastery rise to 0.83. EWMA smoothing effectively mitigates parameter jumps caused by state fluctuations, and the semantic mapping table constrains the feature projection direction, making the semantic correspondence of particle behavior more stable. The results show that the improved mechanism forms a tighter mapping structure at the state-parameter level, thus significantly improving the accuracy and consistency of semantic mapping.

To further verify the statistical robustness of the improved system at the semantic mapping level, paired-samples t-tests are performed on the average correlation coefficients of the five state dimensions. As highlighted in Table 1, the augmented system significantly outperforms the traditional Shuriken particle mechanism across all state dimensions. The overall average correlation coefficient increases from 0.55 to 0.65, representing an average improvement of 0.10. It is worth noting that the differences between S_2 and S_5 are all extremely significant ($p < 0.01$); however, the differences among S_1, S_3, and S_4 are all significant ($p < 0.05$). Overall, these results indicate that the proposed EWMA smoothing and the structure of the semantic mapping table effectively reduces parameter perturbation arising from fluctuations in state and increases the consistency and stability of the semantic mapping.

C. VERIFICATION OF TEACHING IMMERSION AND USER EXPERIENCE

The semantic visualization mechanism realizes a dynamic coupling between learning behavior and visual feedback using a particle system, which makes cognitive states concrete in the process of interaction. The participants were twenty-eight undergraduate learners enrolled in a training course, with ages ranging from nineteen to twenty-three. All participants had prior exposure to digital design soft-

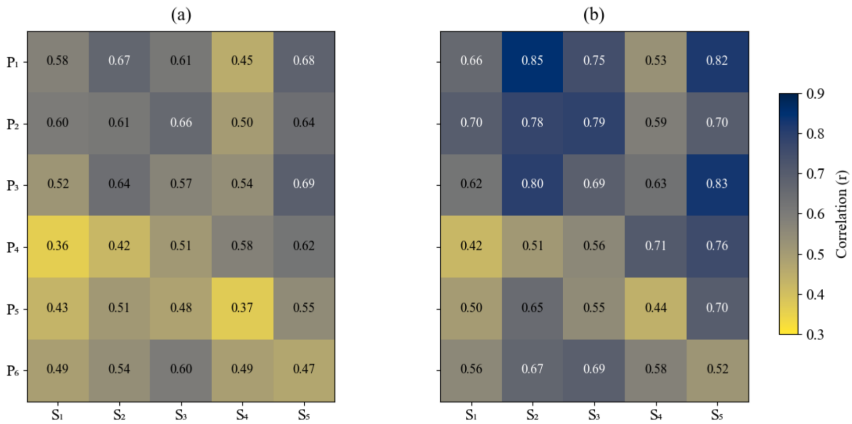

FIGURE 3. Heatmap of semantic mapping accuracy analysis. (a) Traditional Unity Shuriken particle system; (b) The system described in this paper

TABLE 1. Results of the significance test for semantic mapping accuracy

Status dimension	Traditional system (mean)	Improved system (mean)	Improvement	t-value	p-value	Significance
S_1	0.50	0.58	0.08	2.54	0.029	*
S_2	0.56	0.71	0.15	4.12	0.002	**
S_3	0.57	0.67	0.10	3.01	0.012	*
S_4	0.49	0.58	0.09	2.67	0.024	*
S_5	0.61	0.72	0.11	3.58	0.006	**
Average	0.55	0.65	0.10	—	—	—

Note: $p < 0.05$ indicates a significant difference (*), and $p < 0.01$ indicates a highly significant difference (**).

ware used in education, and none had prior experience with particle-based visualization systems. This study was approved by the Ethics Committee of Shanghai Technical Institute of Electronics & Information College of design and art, and was performed in accordance with the principles of the Declaration of Helsinki. All eligible participants signed an informed consent form. The study was conducted during scheduled instructional sessions, and each participant completed all seven learning scenarios under the same device configuration in an independent session. The experiment has established seven teaching scenarios within the Unity platform to explore: cognitive guidance, performance feedback, concept demonstration, collaborative inquiry, situational simulation, assessment feedback, and emotion-driven learning. A five-point scale is used to evaluate learning effectiveness based on immersion and interaction satisfaction. The questionnaire measures differences in user experience when the mechanism is enabled versus disabled and evaluates its adaptability across different teaching tasks. The experimental outcomes are summarized in Figure 4.

The findings indicate that when the mechanism is activated, overall immersion improves by approximately 0.7 points, and interaction satisfaction improves by approximately 0.6 points, with the largest improvements occurring in operational feedback and situational simulation scenarios. Improvements in the immersion metric stem mainly from the system modifying abstract cognitive activity into continuous, observable visual dynamics that allow the learners to maintain stable attention. Interaction satisfaction improved due to reduced feedback delay and greater operational consistency. Collaborative inquiry and emotion-driven scenarios exhibit slower improvement, mainly influenced by the synchronization of group behaviors as well as emotion fluctuations across participants. In sum, the activation of the semantic visualization mechanisms enhances perceptual coherence of the learning process, resulting in a dynamic closed-loop between interaction and cognition.

IV. CONCLUSION

This paper presents a gamified scenario simulation system built with the Unity engine and particle system. This technical foundation offers an adaptable platform for immersive training simulations that accurately model the physical behavior and visual dynamics of fibers and fabrics in complex manufacturing and dyeing processes. An exponentially weighted moving average algorithm is utilized to smooth learning state sequences, and a ScriptableObject mapping table is used to make particle parameters dynamically related to semantic features. This smoothing maintains coherent temporal transitions in semantic features so that particle behaviors reflect a stable trajectory of learner states during dynamic rendering. A ComputeBuffer drives the GPU to render in parallel, generating a real-time feedback loop. Experimental results show that the system improves the average frame rate by approximately 2-3 frames in most scenarios over the baseline, reduces the dispersion of feedback delay during scenario simulation and maintains stable timing in the rendering loop, and increases the average semantic mapping correlation from 0.55 to 0.65 ($p < 0.05$). The work

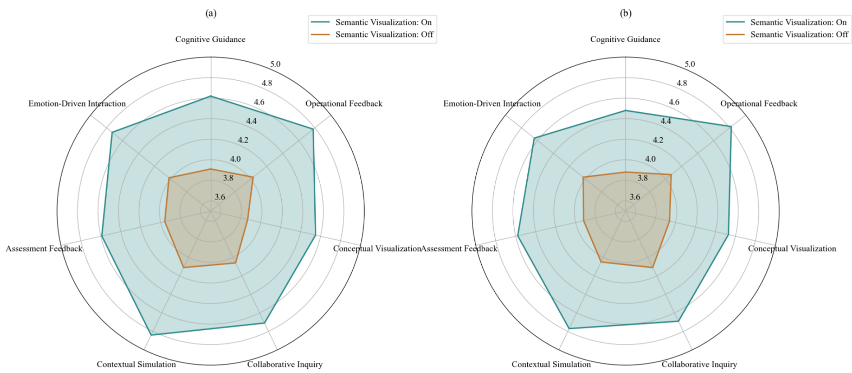

FIGURE 4. Radar chart of learning immersion and interaction satisfaction. (a) Learning immersion; (b) Interaction satisfaction

confirms the capability of semantics-driven visualization mechanisms to enhance both the dynamic stability and semantic consistency of teaching contexts, advancing the evolution of virtual learning experiences toward a unified structure of behavioral state mapping and visual feedback. This validation of using semantic coherence to anchor visual experiences is directly applicable to the industry, affirming the value of such visualizations for professional training and design tasks where the precise semantic mapping of material properties must be consistently reflected in their dynamic visual representation.

AUTHOR CONTRIBUTIONS

Conceptualization – Ling Fan and Yang Yang; methodology – Ling Fan and Yang Yang; investigation – Ling Fan and Yang Yang; writing-original draft preparation – Ling Fan and Yang Yang. All authors have read and agreed to the published version of the manuscript.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

FUNDING

This research received no external funding.

HUMAN RESEARCH SUBJECTS

This study was approved by the Ethics Committee of Shanghai Technical Institute of Electronics & Information College of design and art, and was performed in accordance with the principles of the Declaration of Helsinki. All eligible participants signed an informed consent form.

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

[1] F. Amlasri, "Simulations to teach science subjects: Connections among students; engagement, self-confidence, satisfaction, and learning styles," Education and Information Technologies, vol. 27, no. 5, pp. 7161-7181, 2022, doi: 10.1007/s10639-022-10940-w.

[2] K. Yue, "Design and Application of Virtual Simulation Teaching System under Artificial Intelligence Orientation," Applied Mathematics and Nonlinear Sciences. Erdos Publish House S.L, vol. 9, no. 1, pp. 1-15, 2024, doi: 10.2478/amns-2024-1151.

[3] S. Hong, J. Moon, T. Eom, I. D. Awoyemi, and J. Hwang, "Generative AI-Enhanced Virtual Reality Simulation for Pre-Service Teacher Education: A Mixed-Methods Analysis of Usability and Instructional Utility for Course Integration," Education Sciences, vol. 15, no. 8, pp. 997, 2025, doi: 10.3390/educsci15080997.

[4] I. M. Álvarez, B. Manero, A. Romero-Hernández, M. Cárdenas, and I. Masó, "Virtual reality platform for teacher training on classroom climate management: evaluating user acceptance," Virtual Reality, vol. 28, no. 2, pp. 78-95, 2024, doi: 10.1007/s10055-024-00973-6.

[5] A. Li, J. Li, and L. Nong, "Research on Improving Preschool Teachers' Educational Research Ability Supported by Big Data Technology," International Journal of E-Collaboration (IJeC), vol. 21, no. 1, pp. 1-16, 2025, doi: 10.4018/IJeC.368067.

[6] J. Schoenherr and S. Schukajlow, "Characterizing external visualization in mathematics education research: a scoping review," ZDM-Mathematics Education, vol. 56, no. 1, pp. 73-85, 2024, doi: 10.1007/s11858-023-01494-3.

[7] D. M. Onufriienko, "Using a compute shader for an adaptive particle system," Mathematical Modeling and Computing, vol. 11, no. 1, pp. 1-18, 2024, doi: 10.23939/mmc2024.01.001.

[8] T. Kawamura, Y. Hasegawa, and Y. Idomura, "Interactive steering on in situ particle-based volume rendering framework," Journal of Visualization, vol. 27, no. 1, pp. 89-107, 2024, doi: 10.1007/s12650-023-00945-z.

[9] L. Chu, B. Ai, Y. Wen, Q. Shi, H. Ma, and W. Feng, "A Spatio-Temporal Dynamic Visualization Method of Time-Varying Wind Fields Based on Particle System," ISPRS International Journal of Geo-Information, vol. 12, no. 4, pp. 146, 2023, doi: 10.3390/ijgi12040146.

[10] K. Jaber, E. E. Essel, and P. E. Sullivan, "GPU-native adaptive mesh refinement with application to lattice Boltzmann simulations," Computer Physics Communications, vol. 311, Art. no. 109543, 2025, doi: 10.1016/j.cpc.2025.109543.

[11] M. Faltýnková, O. Meca, T. Brzobohatý, L. Říha, M. Jaroš, and P. Strakoš, "Workflow for high-quality visualisation of large-scale CFD simulations by volume rendering," Advances in Engineering Software, vol. 200, Art. no. 103822, 2025, doi: 10.1016/j.advengsoft.2024.103822.

[12] Y. Zhang, Y. Xu, Y. Xu, Y. Hou, X. Wang, Y. Guo, et al., "Real-time screen space rendering method for particle-based multiphase fluid simulation," Simulation Modelling Practice and Theory, vol. 136, Art. no. 103008, 2024, doi: 10.1016/j.simpat.2024.103008.

[13] V. Cesari, S. D'Aversa, A. Piarulli, F. Melfi, A. Gemignani, and D. Menicucci, "Sense of agency and skills learning in virtual-mediated environment: a systematic review," Brain Sciences, vol. 14, no. 4, pp. 350, 2024, doi: 10.3390/brainsci14040350.

[14] K. Aoyagi, W. Wen, Q. An, S. Hamasaki, H. Yamakawa, Y. Tamura, et al., "Modified sensory feedback enhances the sense of agency during continuous body movements in virtual reality," Scientific Reports, vol. 11, no. 1, pp. 2553, 2021, doi: 10.1038/s41598-021-82154-y.

[15] Z. Alsulaimawi, "Feedback-Enhanced Hallucination-Resistant Vision-Language Model for Real-Time Scene Understanding," 2025. arXiv preprint arXiv:2504.04772.

[16] M. Waseem and M. Hong, "Moving Towards Large-Scale Particle Based Fluid Simulation in Unity 3D," Applied Sciences, vol. 15, no. 17, pp. 9706, 2025, doi: 10.3390/app15179706.

[17] X. Wang, Y. Xu, S. Liu, B. Ren, J. Kosinka, A. C. Telea, et al., "Physics-based fluid simulation in computer graphics: Survey, research trends, and

challenges," Computational Visual Media, vol. 10, no. 5, pp. 803-858, 2024, doi: 10.1007/s41095-023-0368-y.
[18] C. Rao, H. Yu, H. Wan, J. Zhou, Y. Zheng, M. Wu, et al., "Icarus: A specialized architecture for neural radiance fields rendering," ACM Transactions on Graphics (TOG), vol. 41, no. 6, pp. 1-14, 2022, doi: 10.1145/3550454.3555505.
[19] L. Franke, L. Fink, and M. Stamminger, "VR-splatting: Foveated radiance field rendering via 3D Gaussian splatting and neural points," Proceedings of the ACM on Computer Graphics and Interactive Techniques, vol. 8, no. 1, pp. 1-21, 2025, doi: 10.1145/3728302.
[20] P. Jagtap, R. Nasre, V. S. Sanapala, and B. S. V. Patnaik, "Efficient parallelization of SPH algorithm on modern multicore CPUs and massively parallel GPUs," International Journal of Modeling, Simulation, and Scientific Computing, vol. 12, no. 06, Art. no. 2150054, 2021, doi: 10.1142/S1793962321500549.
[21] K. Zheng, Y. Ci, H. Liu, and J. Zhang, "A spatio-temporal process visualization approach for wind features," Computational Geosciences, vol. 25, no. 6, pp. 2055-2067, 2021, doi: 10.1007/s10596-021-10080-z.
[22] L. Chen, Z. Ling, Y. Gao, R. Sun, and S. Jin, "A real-time semantic visual SLAM for dynamic environment based on deep learning and dynamic probabilistic propagation," Complex & Intelligent Systems, vol. 9, no. 5, pp. 5653-5677, 2023, doi: 10.1007/s40747-023-01031-5.
[23] P. Fanta-Jende, D. Steininger, A. Kern, V. Widhalm, J. G. Apud Baca, M. Hofstätter, et al., "Semantic real-time mapping with UAVs," PFG-Journal of Photogrammetry, Remote Sensing and Geoinformation Science, vol. 91, no. 3, pp. 157-170, 2023, doi: 10.1007/s41064-023-00242-2.
[24] X. Liu, Y. He, J. Li, R. Yan, X. Li, and H. Huang, "A comparative review on enhancing visual simultaneous localization and mapping with deep semantic segmentation," Sensors, vol. 24, no. 11, pp. 3388, 2024, doi: 10.3390/s24113388.
[25] S. Wang and B. Xue, "State-space models with layer-wise nonlinearity are universal approximators with exponential decaying memory," Advances in Neural Information Processing Systems, vol. 36, pp. 74021-74038, 2023.
[26] H. Chaudhry, J. Zavatone-Veth, D. Krotov, and C. Pehlevan, "Long sequence hopfield memory," Advances in Neural Information Processing Systems, vol. 36, pp. 54300-54340, 2023.
[27] J. Li and G. Sun, "Design of a virtual simulation interaction system based on enhanced reality," Electronic Research Archive, vol. 31, no. 10, pp. 1-14, 2023, doi: 10.3934/era.2023317.
[28] F. Liarokapis, V. Milata, and F. Skola, "Extended reality educational system with virtual teacher interaction for enhanced learning," Multimodal Technologies and Interaction, vol. 8, no. 9, pp. 83, 2024, doi: 10.3390/mti8090083.
[29] L. Yao, A. Bezerianos, R. Vuillemot, and P. Isenberg, "Visualization in motion: A research agenda and two evaluations," IEEE Transactions on Visualization and Computer Graphics, vol. 28, no. 10, pp. 3546-3562, 2022, doi: 10.1109/TVCG.2022.3184993.
[30] Y. Su, C. Lin, and T. Liu, "Real-Time Trajectory Smoothing and Obstacle Avoidance: A Method Based on Virtual Force Guidance," Sensors, vol. 24, no. 12, pp. 3935, 2024, doi: 10.3390/s24123935.
[31] E. Livieris I, S. Stavroyiannis, and L. Iliadis, "Smoothing and stationarity enforcement framework for deep learning time-series forecasting," Neural Computing and Applications, vol. 33, no. 20, pp. 14021-14035, 2021, doi: 10.1007/s00521-021-06043-1.
[32] T. T. Goh, "Learning management system log analytics: the role of persistence and consistency of engagement behaviour on academic success," Journal of Computers in Education, vol. 1, no. 1, pp. 1-24, 2025, doi: 10.1007/s40692-025-00358-x.
[33] H. Va, M. H. Choi, and M. Hong, "Efficient simulation of volumetric deformable objects in Unity3D: GPU-accelerated position-based dynamics," Electronics, vol. 12, no. 10, pp. 2229, 2023, doi: 10.3390/electronics12102229.
[34] R. Reutov and T. Prikhodko, "Modeling a fluid using a particle system," International Journal of Open Information Technologies, vol. 12, no. 1, pp. 76-82, 2024.
[35] R. A. Girgidov, "Swarm Simulation Modeling Using the Hadamard Product," Moscow University Computational Mathematics and Cybernetics, vol. 47, no. 3, pp. 122-134, 2023, doi: 10.3103/S0278641923030032.