Enhancing Hyper-Personalization in Smart Glasses Design with a Knowledge Graph-Guided GAN Framework

Libo Wang
UCSI University
Nicolaus Copernicus University
326360@o365.stud.umk.pl / free.equality.anyone@gmail.com

Abstract

To build a GAN guided by knowledge graph, this study implements context adaptation and style adjustment for different users' semantic preferences to achieve super-personalized generation capabilities. The system consists of five modules and is simulated and experimentally operated by combining Python 3.13 IDLE with a Python simulator based on custom GPTs. The dataset is two sets of synthetic data that simulate image recognition and text generation respectively. The researcher uses single-group analysis to conduct semantic node perturbation tests and output response observations. The results show that the model is stable under multiple metrics, which proves that the framework has the ability to adjust the output of sentences and patterns in real time according to conditional nodes. This framework provides a design paradigm with closed-verification logic and semantic mapping consistency for the generation system of ultra-personalized smart glasses. And it can be extended to personalized dialogue and context-guided interface in the future, focusing on high-precision human-computer interaction adaptation at the semantic level.

1. Introduction

The current smart glasses market based on artificial intelligence (AI) is gradually showing a diversified development trend, forming a competitive landscape in which the United States leads high-end technology, China promotes consumption popularization, and Europe focuses on industrial applications (Kim & Choi, 2021; Yutong et al., 2021). In the U.S. market, technology companies emphasize the task processing capabilities and software and hardware integration of their products (Zhang et al., 2024). The Solos AirGo Vision series of smart glasses launched by OpenAI achieve instant conversation, speech understanding and image recognition (Solos, 2024). In contrast, China's AI smart glasses market places more emphasis on product implementation and AIoT ecological integration through product development around smart translation, motion monitoring and mixed reality applications (Qi et al., 2024).

However, current smart glasses products lack dynamic and sophisticated personalized design mechanisms, making it difficult to meet diverse market demands. It can be seen from the technology integration and product decision-making of this industry that the core technology of this product involves visual sensing, computer vision and human-computer interaction (Zuidhof et al., 2021; Gao et al., 2024). Although the development of these technologies has improved the computing power and interaction accuracy of devices, it has not formed a design architecture that can adapt to market needs in real time (Lv et al., 2022). Current device manufacturers focus more on hardware performance and sensing accuracy, but ignore how to convert device sensing data into feasible product optimization solutions, making it difficult for user experience to break through universal patterns (Rossos et al., 2024). In addition, the current product management of smart glasses still tends to rely on ex-ante market forecasts and historical data for decision-making and lacks deep learning of users' immediate behaviors, which results in the inability to flexibly adjust the product development process (Nahavandi et al., 2022).

The gap that makes the problem difficult to solve is that existing smart glasses technology still relies on static preset rules and lacks a data-driven dynamic adaptation mechanism, making it difficult for product design to meet the individual needs of users. Because current AI interaction design is still based on preset parameters and fixed rules, it lacks the ability to adapt to user behavior in real time, which results in personalized design being limited to superficial adjustments. For example, although GAN and image generation technology can provide personalized content, they lack controllability and structured learning mechanisms, making the generated results unable to effectively match the interactive logic of smart glasses. Gaze tracking technology can detect the user's viewing area, but the UI configuration is still based on static threshold changes rather than dynamic optimization based on real-time data. This makes product personalized adjustments still rely on manual settings, and it is actually difficult to implement AI autonomous optimization through real implementation.

In addition, most current smart glasses product developers lack a product decision-making mechanism based on real-time data drive, resulting in the personalized design of the product being unable to quickly adapt to the constantly updated market demand. Because its development and management still rely on static market analysis and predictive demand models, this means that product management lacks a mechanism to transform users' real-time behavioral data into product adjustment strategies. In fact, the technical complexity of the industry means that the product management process within the company has long been highly divided, with design, engineering and marketing teams running independently. It makes it difficult to transform the personalized data generated by AI into specific product decision-making indicators due to the lack of cross-department data integration mechanism.

2. Related Work

In the current academic literature, there is no significant documentary evidence for directly applying GAN guided by knowledge graphs to the hyper-personalization of smart glasses design, but research in related fields provides indirect inspiration and evidence. Some studies have proven that knowledge graphs can effectively supplement the data-driven model of GAN in the field of natural language processing by improving the semantic rationality of generated results. However, it is technically feasible from the perspective of the application of GAN in integrating generative AI in personalized hardware devices and smart wearable devices.

2.1. Knowledge Graphs and GANs

Knowledge graph is a semantic network based on a graph structure, and its core operating mechanism involves knowledge representation, relational reasoning and structure learning (Peng et al., 2023). According to research by scholars such as Hogan et al. (2021) and Wang et al. (2017), knowledge graphs are usually constructed in the form of RDF triples "subject-predicate-object", which convert symbolic entities into low-dimensional vector representations through embedding technologies such as TransE, RotatE or DistMult to facilitate machine learning model processing and reasoning (Khan, 2023). As can be seen from the previous description, its core advantage lies in its semantic expression ability and associative reasoning ability (Liu et al., 2024). In recent years, researchers improved the automatic construction and dynamic update capabilities of knowledge graphs by combining deep learning, knowledge distillation and other methods. For example, Microsoft has deployed expansion technology in large language models (LLMs) search engines to greatly improve the accuracy and relevance of search results (Zhong et al., 2023; Pan et al., 2024).

Generative adversarial networks made breakthrough progress in the fields of image, speech and text generation in recent years, making the generated data gradually approach the real distribution (De Rosa & Papa, 2021; Liu et al., 2021; Wali et al., 2022). However, traditional GANs have challenges in learning semantic associations and structural consistency, which makes it difficult for GANs that rely solely on data-driven to be applied to scenarios that require logical reasoning and knowledge constraints (Saxena & Cao, 2021). In this regard, researchers are gradually exploring the combination of knowledge graphs and GANs to improve the semantic rationality, structural integrity and logical consistency of the generated content (Dai et al., 2020). In addition, the knowledge graph provides product design constraints for GAN and avoid invalid samples caused by unconstrained generation (Lembono et al., 2021).

2.2. AI-Driven Personalization in Smart Glasses

The personalized design of smart glasses faces challenges such as product structure complexity, difficulty in data acquisition, and changing market demand (Lee et al., 2021). As mentioned before, the industry mainly relies on static preset parameters for adaptation. For example, according to research by Wang et al. (2023), the conceptual design of smart glasses is based on big data and knowledge graph reasoning to adapt to different scene needs. However, these solutions lack data-driven dynamic adaptation mechanisms and are difficult to meet users' personalized needs. Recent researchers have begun to use deep learning and computer vision technology for personalized optimization (Incel & Bursa, 2023). In view of Dong et al. (2021) research, introducing human prior knowledge into network structure design can make the model have faster convergence speed and lower computational cost.

The core mechanism of Adversarial learning first proposed by Goodfellow et al. (2020) is generative adversarial networks (GAN). As a learning mechanism based on game theory, it approximates the real data distribution through confrontation training of the generator and the discriminator (Durgadevi, 2021). Previous research has shown that knowledge graphs can provide structured prior knowledge, allowing GANs to not only rely on statistical distributions but also follow domain constraints when generating (Dong et al., 2021; Dash et al., 2022). However, the personalized design of smart glasses involves multi-dimensional restrictions such as optical parameters, wearing comfort, and interaction methods. Currently, there is no literature that clearly verifies that GAN guided by knowledge graphs can effectively solve these problems. Therefore, the application of this technology still needs to be experimentally verified to determine its feasibility and advantages in smart glasses products.

3. Knowledge Graph-Guided GAN

In view of the above gaps, this research proposes knowledge graph-guided GAN to fundamentally address the problem of static preset rules through structured semantic constraints and generative adversarial learning. As a decision-making layer, the knowledge graph transforms the visual sensing, behavioral data and device specifications of smart glasses into an interpretable knowledge relationship network, thereby providing constraints for the GAN generation mechanism to meet the user's actual application needs. During the generation process, the content adaptability is dynamically adjusted through the GAN discriminator to ensure the best adaptability between UI layout, interaction model and hardware resources, and avoid disorderly generation.

3.1. Propose Frameworks & Algorithms

Knowledge graph-guided GAN improves the semantic consistency and constraint adaptability of generated data by integrating structured semantic knowledge into the adversarial learning mechanism. The knowledge graph serves as a conditional constraint. Through relational reasoning and node embedding, GAN can dynamically adjust the generated content based on domain knowledge to ensure that the personalized design of smart glasses meets technical standards and user needs (Figure 1).

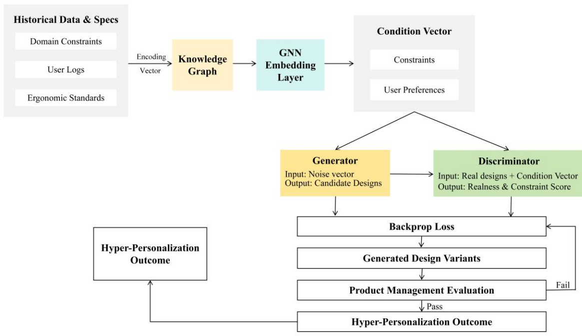

Figure 1. Knowledge Graph-Guided GAN Framework

In this framework, historical data and specs serve as the basic data source of knowledge graph-oriented GAN technology and are responsible for constructing the core parameters and decision-making constraints of intelligent product design. The domain constraints cover industrial standards, technical specifications and market regulatory requirements to ensure that the generated design complies with current technical specifications and market access standards. The user logs come from the interactive behaviors of end users, including frequency of use, scene preferences, adjustment habits, etc., and use deep learning to extract personalized patterns so that the product can better adapt to the evolving needs of different users. The ergonomic standards are based on the wearing behavior and anthropometric data of different groups to ensure the wearing comfort and long-term use adaptability of smart glasses. These heterogeneous data undergo multi-layer feature learning and weight mapping through encoding vectors, and are finally fed into the knowledge graph to combine semantic association and data drive.

$$h_v^0 = \text{initial feature vector of node } v$$

$$h_v^k = \sigma(W^{(k)} \cdot \sum_{u \in N(v)} \alpha_{uv} \cdot h_u^{(k\text{-}1)})$$

where $N(v)$ is the neighborhood of node v; α_{uv} is a weighting factor from adjacency relationship; $W^{(k)}$ is a learnable parameter matrix for layer k; σ is an activation (e.g., ReLU); After K layers, it is $h_v^{(k)}$.

The knowledge graph plays the role of structured data storage and associated reasoning. Its essence is a graph structure data model that is composed of nodes and edges. Among them, nodes represent key attributes related to product design (such as optical quality, battery performance, wearing comfort, etc.), and edges define the semantic relationships and weights between these attributes. In industrial circles, such as Google's Knowledge Vault or Facebook's DeepText, they all construct semantic understanding mechanisms through knowledge graphs (Hubert et al., 2023). In this research architecture, the knowledge graph not only serves as a storage of static information, but also integrates user interaction data and market feedback through a dynamic update mechanism, so that the design elements of smart glasses can be adaptively adjusted as the data changes, providing more semantically relevant input features for downstream deep learning modules.

The GNN embedding layer is responsible for converting structured data in the knowledge graph into numerical vectors that can be used for neural network training. Traditional embedding methods such as word2vec or TF-IDF are difficult to capture complex relationships, while graph neural networks (GNN) can recursively aggregate the information of neighbor nodes to the target node through a message passing mechanism, so that the embedding vector reflects the global structure. In this system, the GNN embedding layer performs multi-layer neighbor aggregation operations, so that different design attributes of smart glasses obtain high-dimensional feature representation in the embedding space, and finally generates a condition vector as the core input of the GAN model, ensuring that the design generation process meets product constraints and personalized needs.

In the design of the condition vector, it is responsible for integrating structured constraints and personalized requirements into data input that can be interpreted by the model. Among them, the constraints cover engineering specifications, ergonomic standards and knowledge graph reasoning results to ensure that the generated design meets technical and market requirements. The user preferences dynamically adjust personalized weights based on historical behavioral data and real-time feedback. After vector encoding, it is input into the generator together with random noise to guide the GAN to produce highly personalized design variants that comply with constraints. The mathematical representation of the algorithm is as follows:

$$\text{Condition} = [E_{KG}, p, d]$$

where E_{KG} is the knowledge graph embedding; p is the user preference vector; d is the historical design data.

This part refers to the technical principles of GAN and applies them to this framework. The core task of the generator is to generate candidate designs based on conditional vectors and random noise. But unlike traditional GAN, it provides product semantic constraints for the generator through knowledge graph embedding, so that the output meets engineering standards and personalized needs. The generator receives the condition vector and generates high-dimensional features through non-linear transformation to ensure that the design solution meets the basic structural conditions while adapting to market changes and user needs. The discriminator is responsible for evaluating whether the design comes from real data and checking whether it violates knowledge graph constraints. Different from traditional GAN, it not only performs binary classification, but also calculates the matching degree based on conditional constraints to ensure that the design results are feasible. The algorithm is as follows:

$$x_{gen} = G\,(z,\text{Condition};\theta_g) = f_g\,(W_g\,[z \oplus \text{Condition}] + b_g)$$

$$p_{\text{real}} = \sigma\,(W_d \cdot [x \oplus \text{Condition}] + b_d)$$

Where x_{gen} is the generated design variant, output from the generator; G is the generator function, parameterized by θ_g; z: is the noise vector, sampled from a probability distribution; W_g is the weight matrix in the generator's neural network; $\oplus$ is concatenation operation, merging noise z with the condition vector; f_g is A nonlinear mapping function; b_g is the bias term in the generator. p_{real} represents probability that a design is "real" or aligns with constraints; D $(x,\text{Condition};\theta_d)$ represents the discriminator function, parameterized by θ_d; W_d is the weight matrix in the discriminator's neural network; b_d is the bias term in the discriminator; σ represents the sigmoid activation function, which maps outputs to a probability range (0,1); x is the input design, which can be either real historical data or a generated design;

The backprop loss, as the core of learning, dynamically adjusts the weight according to the difference between the generated results and the real samples. If the product constraints are violated, a penalty signal is generated to optimize the generator so that the design gradually approaches the optimal solution. The following algorithms are supported:

$$L_D = -\mathbf{E}x_{\text{real}}, \text{Condition}\,[\log D\,(x_{\text{real}}, \text{Condition})] - \mathbf{E}z\,[\log\,(1-D\,(G(z, \text{Condition}), \text{Condition}))]$$

$$L_{\text{constraint}} = (x_{\text{gen}}, E_{KG}) = \sum_i \Phi(x_{\text{gen}}, \text{constraint}_i)$$

Where L_D represents the discriminator loss, designed to maximize correct classification; $L_{\text{constraint}}$ represents the constraint penalty, applied when generated designs violate predefined conditions; Φ $(x_{\text{gen}}, \text{constraint}_i)$ represents A distance function measuring the violation severity for each constraint i.

The generated design variants are the product solutions output by the GAN generator. After conditional vector constraints, a smart glasses design with market adaptability and personalized characteristics is formed. These designs will enter the product management evaluation stage and be screened based on industry standards, ergonomic requirements and consumer preferences. If the design fails to pass the review, it will be marked as "Fail" and return to backprop loss. Then it optimizes the GAN model through gradient adjustment so that the generator learns a design direction that is more in line with market demand. If it passes the evaluation, it will be marked as "Pass" and enter the hyper-personalization outcome stage to ensure that the final design meets highly customized needs.

4. Experiment

Given that the knowledge graph is a connected semantic conditional field rather than an auxiliary label or classification basis, it guides the direction of generating logic and discriminating semantics through node edge weights and context configuration. Embedding nodes from the knowledge graph into the condition generator does not input a single classification vector, but rather structural information that links multiple nodes, varies in weight, and has semantic dependencies. In addition, feedback from the discriminator to the graph forms a closed-loop learning chain that cannot be captured by traditional observation methods, which includes three levels: condition recoding, node semantic superposition, and pattern rewriting. Static design may cut off the semantic diffusion path, making it difficult to adjust the generation behavior in response to slight changes in the knowledge structure (Po et al., 2024). Therefore, this experiment uses a simulation experiment that can control node dynamics and observe the semantic transmission process to reflect the verifiability and reproducibility of the overall technical logic (Kleijnen, 2015; Dai et al., 2024).

The graph embedding vector received by the conditional generator has the characteristics of node semantic superposition, edge weight transfer, and upstream and downstream node linkage, which makes the generation result highly dependent on the graph structure configuration (Yang et al., 2019). The simulation experiment slightly perturbed the weights of specific nodes in the graph to observe whether the generated style accurately corresponds to context changes (Kleijnen, 2015; Xu & Bao, 2024). The multiple discrimination tasks undertaken by the graph discriminator must also be tested one by one under simulation conditions to verify its recognition accuracy of semantic boundaries. The most critical test is the test of the semantic rewrite module, which requires continuous simulation to generate mismatch situations to observe whether the graph node adjustment can converge round by round. All of the above require the synchronous control of graph structure, sample generation, and semantic evaluation in a simulation environment to fully verify the effectiveness and adaptability of the closed-loop graph-guided generation mechanism.

4.1. Setup

The researcher used single-group analysis as the core experimental design, because there is no equivalent reference system for knowledge graph-guided GAN in the industry and academia. The field of smart glasses has not yet demonstrated a multi-module generative network that combines graph neural representation, knowledge graph constraints, and semantic generation closed loops, which leads to the establishment of a control group with the same performance foundation and structural logic. The use of single-group analysis can instead focus on the generation style variation, semantic retention, and sentence matching performance of this system under conditional node perturbations (Marsden & Torgerson, 2012). And it verifies whether the model has the ability to accurately super-personalize across contexts through step-by-step node adjustment in a simulated environment, ensuring that super-personalized generation is structurally fully verifiable.

By using Python 3.13 IDLE combined with a Python simulator based on the Custom GPTs, the combination not only provides choices for the experimental environment, but also directly corresponds to the execution process and data generation logic. This is because the experiment needs to dynamically adjust the input conditions based on the knowledge graph node combination, and generate corresponding images and semantic samples in real time. The Python simulator provides a modeled semantic condition input mechanism that performs alternating simulations for node weights, graph topology, and constraint types to make up for the lack of real user data (Figure 2). The simulator's built-in structured output mechanism also facilitates the instant export of documents in multiple formats, and combines custom semantic criteria for repeated trials.

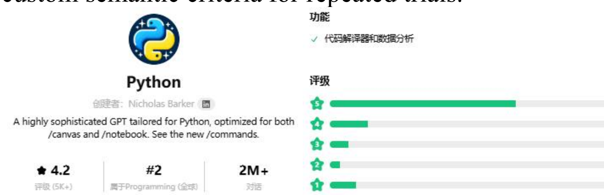

Figure 2. Python simulator based on GPTs

Python 3.13 IDLE maintains the highest environment compatibility to facilitate the initialization and interoperability of the multi-module framework system. It ensures that the graph embedder, generator, and writeback modules can operate in conjunction in a single environment.

4.2. Dataset

Due to the strict requirements of semantic control level and structural correspondence logic, this study uses a synthetic dataset as training and validation samples, including 1,000 images and 50,000 user log texts in JSON format. Because the knowledge graph-guided GAN core includes a closed-loop mechanism of graph node condition generator, semantic identifier and write-back module. Node input must be adjustable, reconfigurable and semantically calibrated to simulate the user's semantic preferences in different scenarios. However, most of the existing public data on smart glasses lack dynamic user logs and cannot provide diverse and structured node semantic combination inputs. The lack of style tracking data under continuous semantic shift also makes it difficult to support the node response test of the generator.

The researcher constructed two types of synthetic data sets: image-based diary data to simulate visual preference transfer; text-based semantic data to reorganize the user's target statement and the corresponding vocabulary of the graph nodes to generate simulated context output. The scheme explicitly corresponds to the semantic mapping structure between the generator and the graph nodes. It adjusts the input nodes for each set of samples and tests the response accuracy and semantic consistency.

4.3. Implementation

During the execution, the researcher used Python 3.13 IDLE to write the main program logic and initialize the knowledge graph-guided GAN module. This stage will preset the node dimension, edge weight range and semantic tag format, and configure the JSON output format recognizable by the simulator as a connection point for subsequent model intercommunication and semantic monitoring. After completing the initialization, enter the Python simulator under the Custom GPTs framework. Enter the first round of semantic training instructions KGG-GAN code, and set the first set of synthetic data to the image log class. The researcher observes the stability of the generator's style transfer behavior under different edge weight inputs, and the discriminator outputs the semantic correspondence score.

After constructing the image dataset in the Python simulator with prompt, the researcher switched to the second set of synthetic data, namely text-based semantic input samples. The simulator generates training input sequences with graph node vocabulary combinations for sentence generation and semantic strength testing. The focus of this stage is not on language fluency, but on the correspondence completeness of semantic nodes and the constraint coverage of sentence content. The image identifier will be started in parallel with the semantic indicator module to execute and calculate the six metrics of Frechet Inception Distance (FID), Inception Score (IS), Precision, Recall, Structural Similarity Index Measure (SSIM), and NDCG@K; for text-based user logs, it executes and calculates the six metrics of BLEU, METEOR, BERTScore, Precision, Recall, and NDCG@K. All the codes and experimental records required for the experiment have been uploaded to the Github repository to ensure that they are publicly available.

5. Result

The experimental results are all stable outputs selected after multiple rounds of training. Due to occasional calculation deviations during the simulator execution, the researcher had to repeat sample generation and evaluation. It excludes semantic alignment anomalies or node writeback errors, and the data presented are all optimal state records.

The essence of hyper-personalization lies in whether the generated results can accurately respond to user conditions and maintain semantic consistency. In image recognition, FID and IS measure style differences and generation diversity respectively; Precision and Recall test whether the generation is close to the target style range; SSIM evaluates the accuracy of visual structure restoration; NDCG@K reflects the correspondence between the generated samples and the user's preference ranking. In text generation, BLEU and METEOR evaluate word overlap and semantic segment alignment; BERTScore emphasizes semantic level similarity; Precision and Recall check the completeness of semantic correspondence; and NDCG@K again corresponds to whether the sentence generation conforms to the node preference arrangement. The following table 1 shows the quantitative metrics of graph items and text logs obtained through 8 group of repeated experiments based on Python simulator calculations.

Table 1. Metrics for image recognition

Group	FID	IS	Precision	Recall	SSIM	NDCG@K
1	11.017	7.011	0.656	0.517	0.711	0.769
2	13.689	7.885	0.679	0.537	0.743	0.720
3	14.088	7.963	0.693	0.544	0.733	0.714
4	13 126	7.253	0.617	0.476	0.778	0.753
5	15.784	7.736	0.653	0.534	0.766	0.718
6	12.062	7.791	0.673	0.493	0.798	0.764
7	12.829	7.998	0.697	0.473	0.725	0.713
8	13.051	7.344	0.632	0.543	0.749	0.692

The overall range was 10 to 16, which showed that the difference between the generated image and the true distribution was relatively acceptable. The IS value was concentrated in the range of 7.011 to 7.998, indicating that the sample diversity and recognizability were at a basic level. The Precision and Recall data showed a certain gap, with the highest Precision appearing in Group 7 (0.697); and the highest Recall was in Group 8 (0.543), indicating that the model retained features well when generating patterns, but the coverage range still had an upper limit; SSIM reached a high point of 0.798 in the 6th group; most samples fell steadily between 0.711 and 0.778, indicating good consistency in visual structure preservation; in terms of NDCG@K, the 1st and 6th groups were the best performing samples, at 0.769 and 0.764 respectively, and the overall range was maintained at 0.692 to 0.769. This shows that the generated style rankings are well consistent with the correlation between user preferences. The evaluation of text generation is shown in table 2.

Table 2. Metrics for text generation

Group	BLEU	METEOR	BERTScore	Precision	Recall	NDCG@K
1	0.231	0.381	0.741	0.631	0.526	0.761
2	0.287	0.369	0.751	0.663	0.537	0.739
3	0.243	0.372	0.739	0.617	0.558	0.721
4	0.278	0.355	0.765	0.684	0.547	0.722
5	0.254	0.298	0.721	0.634	0.592	0.699
6	0.266	0.318	0.775	0.609	0.567	0.742
7	0.271	0.337	0.718	0.656	0.581	0.703
8	0.259	0.347	0.738	0.678	0.556	0.719

The text generation data showed that the BLEU index performance was concentrated between 0.231 and 0.287, with Group 2 being the highest (0.287). Although there is still a gap with the top language models, it has met the semantic transcription goals set in this study. The highest

METEOR performance was in Group 1 (0.381), and the lowest was 0.298 in Group 5, indicating that there is still room for improvement in the coverage of word variation and the stability of semantic segment matching. In terms of BERTScore, Group 6 achieved the best performance of 0.775, and the vast majority were distributed in the range of 0.718 to 0.765, proving that the overall correspondence of the semantic level remained at a good level. The highest Precision was in Group 4 (0.684), and the highest Recall was in Group 5 (0.592). The ability to accurately present key words in the explanation sentences and the breadth of semantic coverage varied among different samples. In terms of NDCG@K, Group 1 had the highest score (0.761), and Group 5 had the lowest score (0.699). The overall distribution still showed that the model output sentences and the user's semantic node preferences were clearly aligned, showing a stable and hyper-personalized semantic generation tendency.

6. Limitation

In this experiment, the synthetic data input to the Python simulator are JSON format samples that are carried by computer language, and are not composed of real pictures and texts obtained by camera lenses. It means that the semantic density and color texture representation of the visual style space learned by the generator are still different from the real-world input. Although this limitation does not affect the mechanism verification and logical accuracy of the simulation stage, the hyper-personalization effect of the knowledge graph-guided GAN in the practice of smart glasses systems may be different from this result.

7. Conclusion

Using the knowledge graph as a conditional guide for the GAN framework, this research conducts module interaction between node semantic embedding and graph structure logic. It builds four layers of dynamic links, including image recognition, semantic generation, closed-loop adjustment, and sentence correspondence, to achieve ultra-personalization of smart glasses. Node weights, boundary relationships, and upstream and downstream semantic correspondence together form the input basis of the conditional generator. The conditional generator processes the combination of graph node weights and then passes it into the semantic variation logic. The image identifier then performs a double-layer evaluation for style and context consistency, and the writeback module adjusts the node parameters to enhance semantic convergence. After repeatedly perturbing the node structure through the Python simulation environment and importing the dual dataset test, all indicators show stable results based on the data results. In the future, if the adaptive ability of the node graph is further expanded, the dynamic shaping of the graph structure can be realized by fusing the sensor scene input, which will further expand the system to complex fields for semantic self-rule establishment and output convergence.

Code Availability

The relevant architecture code and some experimental records have been uploaded to the GitHub repository for sharing:
https://github.com/brucewang123456789/GeniusTrail/tree/main/Knowledge%20Graph-Guided%20GAN

References

[1] Yung-Hung Chen, Chih-Hsuan Huang, Sin-Wun Syu, Tien-Ying Kuo, and Po-Chyi Su. Egocentric-view fingertip detection for air writing based on convolutional neural networks. *Sensors*, 21(13):4382, 2021.

[2] Chuan-Pu Dai, Feng Ke, Yufei Pan, Jaehyun Moon, and Zhen Liu. Effects of artificial intelligence-powered virtual agents on learning outcomes in computer-based simulations: A meta-analysis. *Educational Psychology Review*, 36(1):31, 2024.

[3] Yujie Dai, Shuai Wang, Xiaowei Chen, Chuan Xu, and Wen Guo. Generative adversarial networks based on Wasserstein distance for knowledge graph embeddings. *Knowledge-Based Systems*, 190:105165, 2020.

[4] Tanmay Dash, Saurabh Chitlangia, Ankur Ahuja, and Anirban Srinivasan. A review of some techniques for inclusion of domain-knowledge into deep neural networks. *Scientific Reports*, 12(1):1040, 2022.

[5] Gustavo Henrique De Rosa and João Paulo Papa. A survey on text generation using generative adversarial networks. *Pattern Recognition*, 119:108098, 2021.

[6] Tao Dong, Qiang Qi, Jing Wang, Aoxiang Liu, Hao Sun, Zhiqiang Zhuang, and Jianwei Liao. Generative adversarial network-based transfer reinforcement learning for routing with prior knowledge. *IEEE Transactions on Network and Service Management*, 18(2):1673–1689, 2021.

[7] M. Durgadevi. Generative adversarial network (GAN): A general review on different variants of GAN and applications. In *Proceedings of the 2021 6th International Conference on Communication and Electronics Systems (ICCES)*, pages 1–8, 2021.

[8] Li Gao, Chenglong Wang, and Guoyin Wu. Wearable biosensor smart glasses based on augmented reality and eye tracking. *Sensors*, 24(20):6740, 2024.

[9] Axel Hogan, Eva Blomqvist, Maarten Cochez, Claudia d' Amato, Gerard de Melo, Claudio Gutierrez, and Axel Zimmermann. Knowledge graphs. *ACM Computing Surveys*, 54(4):1–37, 2021.

[10] Yu-Jie Huang, Jiun-Cheng Lin, Sheng-Shiun Lee, and Bo-Jun Wu. Reading and walking with smart glasses: Effects of display and control modes on safety. *International Journal of Human – Computer Interaction*, 40(23):7875–7891, 2024.

[11] Nicolas Hubert, Philippe Monnin, Alexandre Brun, and Denis Monticolo. Sem@K: Is my knowledge graph embedding model semantic-aware? *Semantic Web*, preprint:1–37, 2023.

[12] Ozlem Durmaz Incel and Sibel Özlem Bursa. On-device deep learning for mobile and wearable sensing applications: A review. *IEEE Sensors Journal*, 23(6):5501–5512, 2023.

[13] Aftab Khan. Knowledge graphs querying. *ACM SIGMOD Record*, 52(2):18–29, 2023.

[14] Donghyun Kim and Yong Choi. Applications of smart glasses in *applied sciences*: A systematic review. Applied Sciences, 11(11):4956, 2021.

[15] Jack P. C. Kleijnen. Design and analysis of simulation experiments. In *Proceedings of the International Workshop on Simulation*, pages 3–22, 2015.

[16] Chi Kin M. Lee, Lok Lui, and Yiu P. Tsang. Formulation and prioritization of sustainable new product design in smart glasses development. *Sustainability*, 13(18):10323, 2021.

[17] Teguh Santoso Lembono, Enrico Pignat, Jakub Jankowski, and Sylvain Calinon. Learning constrained distributions of robot configurations with generative adversarial network. *IEEE Robotics and Automation Letters*, 6(2):4233 – 4240, 2021.

[18] Anran Liu, Dongyun Zhang, Yizhe Wang, and Xiaowei Xu. Knowledge graph with machine learning for product design. *CIRP Annals*, 71(1):117–120, 2022.

[19] Ming-Yu Liu, Xiaodong Huang, Jun-Yan Zhu, Ting-Chun Wang, and Arjun Mallya. Generative adversarial networks for image and video synthesis: Algorithms and applications. *Proceedings of the IEEE*, 109(5):839–862, 2021.

[20] Xiaolong Liu, Tianze Mao, Yanchao Shi, and Yuxuan Ren. Overview of knowledge reasoning for knowledge graph. *Neurocomputing*, 127571, 2024.

[21] Zhengyu Lv, Federico Poiesi, Qiang Dong, Jaime Lloret, and Haibin Song. Deep learning for intelligent human – computer interaction. *Applied Sciences*, 12(22):11457, 2022.

[22] Emma Marsden and Carole J. Torgerson. Single group, pre- and post-test research designs: Some methodological concerns. *Oxford Review of Education*, 38(5):583 – 616, 2012.

[23] Dariush Nahavandi, Roohallah Alizadehsani, Abbas Khosravi, and U. Rajendra Acharya. Application of artificial intelligence in wearable devices: Opportunities and challenges. *Computer Methods and Programs in Biomedicine*, 213:106541, 2022.

[24] Lazzat Orynbay, Bibigul Razakhova, Peter Peer, Bostjan Meden, and Ziga Emersic. Recent advances in synthesis and interaction of speech, text, and vision. *Electronics*, 13(9):1726, 2024.

[25] Shuai Pan, Lian Luo, Yizhe Wang, Chao Chen, Jie Wang, and Xiaofang Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[26] Chenxi Peng, Feng Xia, Mehran Naseriparsa, and Francesco Osborne. Knowledge graphs: Opportunities and challenges. *Artificial Intelligence Review*, 56(11):13071 – 13102, 2023.

[27] Ruofan Po, Yifan Wang, Vladislav Golyanik, Kevin Aberman, Jonathan T. Barron, Ariel Bermano, and Gordon Wetzstein. State of the art on diffusion models for visual computing. *Computer Graphics Forum*, 43(2):e15063, 2024.

[28] Wei Qi, Xiaowei Xu, Kai Qian, Björn W. Schuller, Gennaro Fortino, and Andrea Aliverti. A review of AIoT-based human activity recognition: From application to technique. *IEEE Journal of Biomedical and Health Informatics*, 2024.

[29] Dimitrios Rossos, Alex Mihailidis, and Brandon Laschowski. AI-powered smart glasses for sensing and recognition of human – robot walking environments. In *Proceedings of the 2024 10th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*, pages 62–67, 2024.

[30] Divya Saxena and Junfeng Cao. Generative adversarial networks (GANs): Challenges, solutions, and future directions. *ACM Computing Surveys*, 54(3):1–42, 2021.

[31] Solos. The Verge reports Solos's upcoming AirGo V smartglasses with ChatGPT-4o and a camera, 2024.

[32] Matthias Tretter, Michael Hahn, and Peter Dabrock. Towards a smart glasses society? Ethical perspectives on extended realities and augmenting technologies. *Frontiers in Virtual Reality*, 5:1404890, 2024.

[33] Ali Wali, Zunaira Alamgir, Shaista Karim, Ahmad Fawaz, Muhammad Bilal Ali, Mohammad Adan, and Muhammad Mujtaba. Generative adversarial networks for speech processing: A review. *Computer Speech & Language*, 72:101308, 2022.

[34] Jian Wang, Tong Shi, Meng Liu, and Kaikai Jiang. Knowledge graph based augmented reality work instruction for wire harness final assembly on formboard. In *Proceedings of the 2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, pages 1–6, 2023.

[35] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.

[36] Heng Xu and Jia Bao. Dynamic knowledge graph-based dialogue generation with improved adversarial meta-learning. In *Proceedings of Artificial Intelligence and Human-Computer Interaction,* pages 13–20, 2024.

[37] Chao Yang, Pengfei Zhuang, Wei Shi, Anh Luu, and Ping Li. Conditional structure generation through graph variational generative adversarial nets. *Advances in Neural Information Processing Systems*, 32, 2019.

[38] Yutong Qiu, Ji Hang, Jun-Rong Chen, and P. J. Ng. The impact of smart glasses on a new generation of users. *International Journal of Business Strategy and Automation*, 2(4):1–25, 2021.

[39] Dongyang Zhang, Yiming Li, Zeyu He, and Xiaolong Li. Empowering smart glasses with large language models: Towards ubiquitous AGI. In *Companion of the 2024 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 631–633, 2024.