ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 1 - 2026

Poster Layout Optimization Driven by Deep Reinforcement Learning

Yiqiu Yu[1,*] *and Qiulin Xu*[2]

[1] College of Art and Design,Ningbo University of Finance & Economics, Ningbo 315175, Zhejiang, China

[2] School of design and communication, Zhejiang Fashion Institute of Technology, Ningbo 315000, Zhejiang, China

SUMMARY: *Poster advertising layout optimization requires collaborative control of visual hierarchy, semantic primary and secondary, and spatial balance. However, manual design is costly and unstable across scenes. We propose a deep reinforcement learning framework that models layout generation as a sequential decision process of title, body, image, logo, price, button, and whitespace. The multi-branch state encoder integrates geometric properties, saliency cues, alignment relationships, and semantic importance, and the reward function jointly evaluates readability, balance, overlap penalty, information priority, and aesthetic consistency. Experimental results on 12480 poster samples show that the layout quality score of the model is 91.3, and the number of labeled elements is 74860. Compared with the rule template method, the accuracy of element alignment is improved from 84.7% to 92.6%, the overlap rate is reduced from 6.8% to 1.9%, and the average inference time is 0.18 s per sample. The framework provides a computable and scalable solution for intelligent layout optimization of posters in digital design systems for commercial scenes.*

KEYWORDS: *Deep reinforcement learning; Poster advertisement; Layout optimization; Intelligent visual computing*

1 Introduction

Poster advertisement layout is the core computing task in digital visual communication system, and the layout result directly affects the information recognition sequence, visual stay path and the efficiency of brand semantic communication. Computer-generated typography needs to deal with the geometric relationships among headings, text, images, logos, decorative blocks and white space simultaneously. It also needs to maintain alignment order, hierarchical clarity and local visual tension within a limited frame. In the poster task, where the screen size is fixed and the content combination is changeable, the layout algorithm not only has to decide the placement of elements, but also synchronously control the size proportion, boundary distance and reading rhythm. This task has clear engineering computational properties. Li et al. focused on text layout generation on natural images, and introduced deep aesthetic learning into the text layout process, so that the layout results gave consideration to both content embedding and visual harmony [1]. Chen et al. proposed a graphical layout generation method under element condition constraints, in which element attributes were used as generation control variables to improve the responsiveness of layout output to content conditions [2]. Chen et al. further constructed a multi-constrained graphic layout generation

*yuyiqiu@nbufe.edu.cn
https://doi.org/10.65102/is2026321
Published: 30 April, 2026

system, which incorporated factors such as location, scale and color matching into a unified framework and promoted the collaborative evolution of layout generation to comprehensive constraints [3]. Wu et al. studied the process of customized magazine layout generation and showed that there was a learnable mapping relationship between content organization and page structure [4]. Cheng et al. discussed the automatic design of poster text based on visual perception mechanism, which made the distribution of sight lines and information saliency in poster scenes obtain more explicit computational expression [5].

As generative models and reinforcement learning methods enter the field of design computing, the focus of layout research shifts from rule choreography to learnable decisions. Kakooee et al. applied deep reinforcement learning to spatial layout design and proved that the sequential decision-making mechanism could continuously correct layout actions in a complex constrained environment [6]. Wang et al. combined deep learning and graph algorithms to realize automatic building layout generation, which strengthened the role of node relationship and spatial adjacency in layout modeling [7]. Aalaei et al. proposed graph constraint GAN to stabilize architectural layout output with structural constraints, providing a transferable idea for the expression of relations between layout elements [8]. Jiang et al. processed the layout construction task through the site embedded generation model, and showed that the external environment features can significantly change the layout distribution pattern [9]. Cheng et al. introduced a learning strategy into virtual world layout generation and demonstrated the expansion ability of multi-scene layout modeling [10]. In order to make the related research vein clearer, Table 1 summarizes some representative works.

Table 1: Summary of related studies

Reference	Research Object	Method	Findings and Implications
[11]	Conditional room layout generation	Graph neural networks for spatial relationship modeling	Relationship encoding can improve layout controllability.
[12]	Automated architectural layout systems	Comparison of different automatic configuration methods	The system implementation path provides engineering reference value.
[13]	Automatic indoor space layout	Convolutional neural networks for layout feature extraction	Visual features can support layout decision-making.
[14]	Research on intelligent layout generation	Review and analysis of deep generative models	Generative methods have become the mainstream direction.
[15]	Document layout analysis	Semi-supervised learning framework for segmentation	Layout understanding can in turn support layout generation.

Yao et al. built a graph neural network model around conditional room layout generation, emphasizing the supporting role of topological relationships between objects on layout quality [11]. Zhong et al. compared the implementation processes of two kinds of automatic building layout systems and showed that the layout algorithm depends not only on model performance, but also on system-level configuration mechanism [12]. Wu et al. discussed the automatic layout method of indoor space based on convolutional neural network, and further showed that visual feature learning had fundamental value for layout decision-making [13]. Shi et al. summarized the evolution route of deep generative models in intelligent layout generation,

pointing out that layout learning is shifting from local arrangement to global collaborative modeling [14]. Banerjee et al. verified the structure recognition ability under the condition of weak annotation through the semi-supervised document layout analysis framework, which provided a method support for the utilization of complex layout data [15]. Gemelli et al. organized scientific document layout analysis dataset and annotation system, and promoted the data standardization of layout tasks [16]. Pena et al. proposed a human-machine collaborative data collation and continuous layout analysis mechanism to provide a more stable iterative basis for layout evaluation [17]. Vesalainen et al. proposed the document layout error rate index, which provided a more fine-grained measurement tool for the evaluation of layout structure [18]. Wu et al. proposed a unified layout analysis framework for complex documents to enhance the recognition consistency of multi-region and multi-level layout structure [19]. Wu et al. also used document style guide to carry out cross-domain layout analysis, indicating that layout model has begun to pay attention to style transfer and inter-domain generalization ability [20].

In the context of the existing layout generation research gradually shifting from static rules to learning strategies, the requirements for the model in the poster advertising scene are no longer limited to the feasible arrangement of elements, but further to the joint modeling of semantic primary and secondary, visual rhythm and spatial coordination. Based on this technical orientation, we incorporate the title, image, logo, text and white space into a unified state representation, construct a deep reinforcement learning optimization framework for typography decision, and carry out experimental analysis combining alignment accuracy, overlap control, hierarchical consistency and reasoning efficiency. The following contents will introduce the task description and state modeling method, action decision and reward feedback mechanism, model implementation process, data construction and experimental results in turn, so as to form a complete calculation path of intelligent poster advertisement layout.

2 Research methods and models

2.1 Poster advertisement layout task description and state space modeling

The poster layout task can be represented as a sequence layout process on a constrained canvas. The system input contains the title, body text, product image, brand logo, price tag, decorative graphics, and background size information, and the output is the position, scale, hierarchy, and alignment of each element on a two-dimensional canvas. Different from ordinary mixed layout of graphics and text, poster ads assume the functions of information transmission and visual transformation at the same time. The layout results not only determine the reading path, but also affect the recognition order of users 'primary information, secondary information and action prompts. Therefore, task modeling cannot only use coordinates and dimensions, but also write semantic levels, salient regions, adjacent spacing, and local whitespace into a unified state space.

As shown in Fig. 1, the original template samples are firstly segmented by text blocks, detected by image subject, extracted by logo area and normalized by canvas size, and then the elements from different sources are mapped into a unified description space. The text element records the line number, font size difference and keyword density, the image element records the main body coverage, edge safety zone and color contrast, and the function element records the semantic weight and the minimum visible threshold. In this way, the state space is no longer just an isolated geometric set, but a structured input that can express both content

attributes and spatial order.

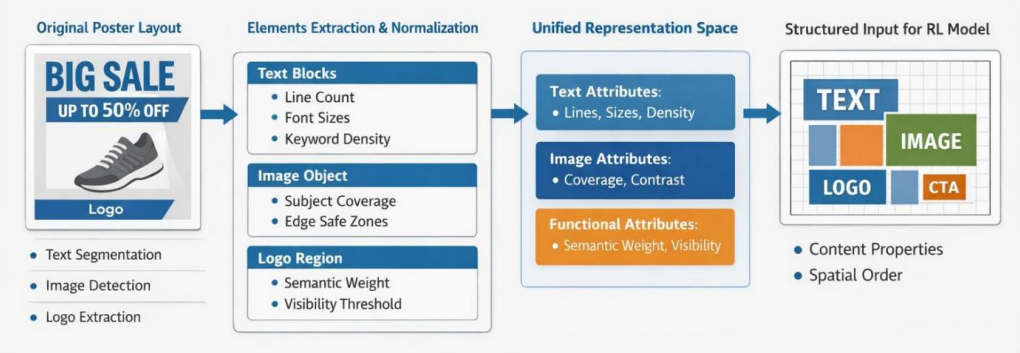

Figure 1: Flowchart of input parsing and canvas mapping for the poster AD layout task

In order to compress single-element attributes into a computable representation, we construct the following element state encoding formula:

$$z_i = \tanh\left(A_p g_i + A_q c_i + A_v r_i + b_z\right) \tag{1}$$

where z_i represents the base state vector of the i layout element, g_i represents the geometric feature composed of the center coordinate, width, height, and area proportion, c_i represents the embedding of the element category, r_i represents the visual feature composed of salient response, font intensity, and color difference, A_p, A_q, A_v are the mapping matrices, and b_z is the bias term. The function of this formula is to map heterogeneous elements into the same latent space, which is convenient for subsequent relationship modeling and policy updating.

A single element vector does not fully represent the layout structure. There are obvious relative constraints between headers and images, prices and buttons, logos and whitespace, and there is a need to further describe the connections between elements.

As shown in Fig. 2, the state space consists of a local description layer, a relational constraint layer, and a global sink layer. The local description layer records the center coordinates, width/height ratio, font level and color density of each element. The relationship constraint layer records the alignment, spacing and occlusion relationship between elements.

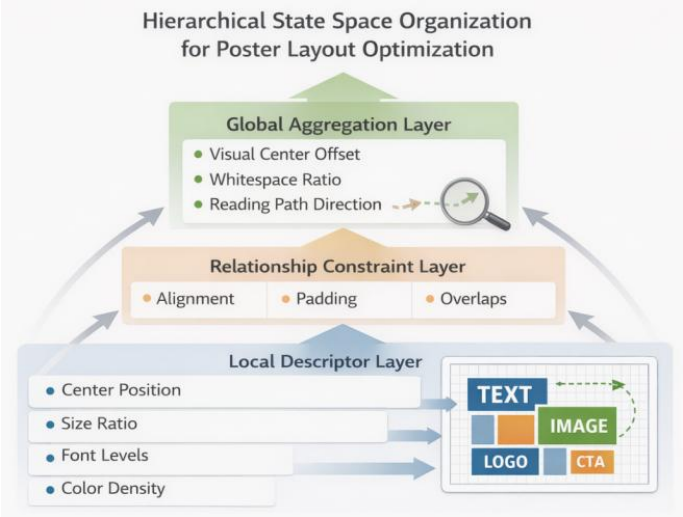

Figure 2: Hierarchical organization diagram of the layout state space

In order to explicitly describe the structural connections between elements, this paper defines the following relational coupling function:

$$\chi_{ij}=\sigma\left(\mu_1 u_{ij}+\mu_2 d_{ij}-\mu_3 o_{ij}+\mu_4 s_{ij}\right) \tag{2}$$

Here, χ_{ij} represents the relationship strength between element i and element j, u_{ij} represents the alignment consistency, d_{ij} represents the attenuation term of the normalized adjacency distance, o_{ij} represents the occlusion ratio, s_{ij} represents the semantic coupling strength, μ_1 to μ_4 are the learnable weights, and $\sigma(\cdot)$ is the activation function. This expression is used to reflect the structural dependencies between elements and avoid the model from only starting from the layout of a single object and ignoring the overall order.

After the relationship is computed, the system proceeds to build the global layout state, which is used to assess whether the current canvas is close to the target distribution of balance, clarity, and readability. In order to enable the policy network to read the current layout environment at the overall level, we define the following form of global state aggregation:

$$H_t=\sum_{i=1}^{N}\alpha_i z_i+\sum_{i=1}^{N}\sum_{j\neq i}\beta_{ij}\chi_{ij} \tag{3}$$

Here, H_t denotes the global layout state at time t, N denotes the number of elements in the canvas, α_i denotes the element-level aggregation weights, and β_{ij} denotes the relation-level aggregation weights. This formula is used to compress local features and relationship features into a unified representation of the environment, so that subsequent strategy decisions can perceive single element changes and the overall layout trend at the same time.

In the actual calculation, this paper uses the hybrid state representation of discrete grid and continuous coordinate parallel. The discrete grid is used to quickly judge collision, out-of-bounds and crowded intervals, and the continuous coordinates are used to maintain fine-grained displacement information during the fine-tuning process. The text block additionally records line breaking potential and character density, the image area additionally records body offset, and the price and button additionally records visual priority and minimum safety margin. The state space constructed in this way can cover the main layout factors in the poster advertisement format, and also provides a stable input for the subsequent visual hierarchy constraint and reinforcement learning strategy optimization.

2.2 Deep Reinforcement Learning Optimization techniques for Layout Visual Hierarchy Constraints

In a poster advertising scenario, the layout is not only determined by the placement of elements, but also by the formation of a clear visual hierarchy between the main title, main image, price information, auxiliary instructions, and action entry. Deep reinforcement learning can learn layout strategy through continuous trial and feedback update. However, without hierarchical constraints, the model is easy to obtain locally neat layout results with unclear primary and secondary. In this paper, we introduce visual hierarchy constraints in the policy optimization stage, so that the agent can pay attention to the saliency distribution, semantic level and spatial order simultaneously during the training process, thus promoting the layout optimization from pure spatial search to layer-aware policy learning.

As shown in Fig. 3, the hierarchical constraint is composed of saliency branch, semantic

branch and geometric branch. The saliency branch extracts attention cues based on regional contrast, subject coverage, and font intensity, the semantic branch extracts functional levels based on element categories and information priorities, and the geometric branch extracts spatial order based on center position, alignment direction, and spacing ratio. The three branches converge at the fusion layer to form a hierarchical constraint vector, which is fed into the policy network and the value network. After this processing, the model is able to determine whether an action disrupts the order between primary vision and secondary information. In order to ensure that the layout learning process has persistent memory ability for layout rules, the system adopts off-policy reinforcement learning framework and retains high-quality layout clips in experience playback.

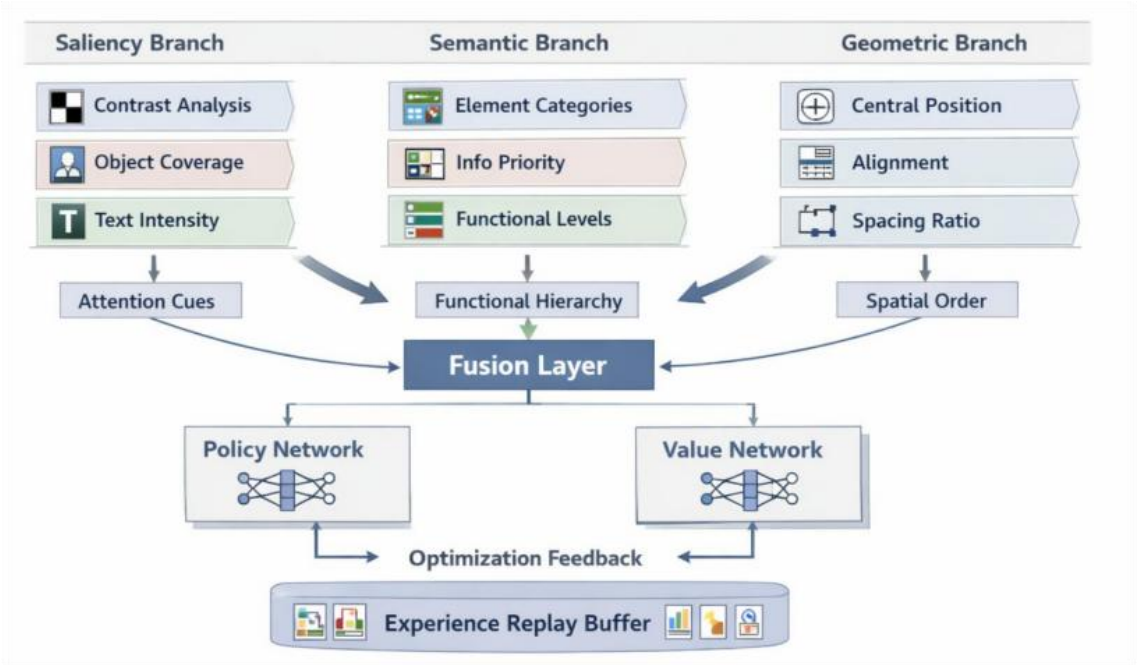

Figure 3: The encoding flowchart of the visual hierarchy constraints for typography

In order to write three types of heterogeneous information into the hierarchical representation space, this paper defines the following hierarchical fusion form:

$$y_t=\mathrm{Softmax}\left(B_s m_t+B_e n_t+B_g k_t\right) \tag{4}$$

where y_t represents the hierarchical distribution vector at time t, m_t represents the saliency branch output, n_t represents the semantic branch output, k_t represents the geometric branch output, and B_s, B_e, and B_g are mapping matrices. The function of this formula is to compress the hierarchical cues from different sources into a unified probability distribution, so that the model can directly judge whether the attention ordering of various elements in the current layout is reasonable.

As shown in Fig. 4, the training process includes six links: state sampling, action execution, reward calculation, sample caching, parameter update and policy correction. After each iteration, the model will calculate the alignment deviation, overlap area, hierarchical confusion rate and blank imbalance degree, and return them as constraint signals. Different from the training method that only aims at the final score, this paper adds a local hierarchical constraint reward in the intermediate step, so that the policy network can avoid putting high-priority elements into low-concern regions at the early stage of arrangement.

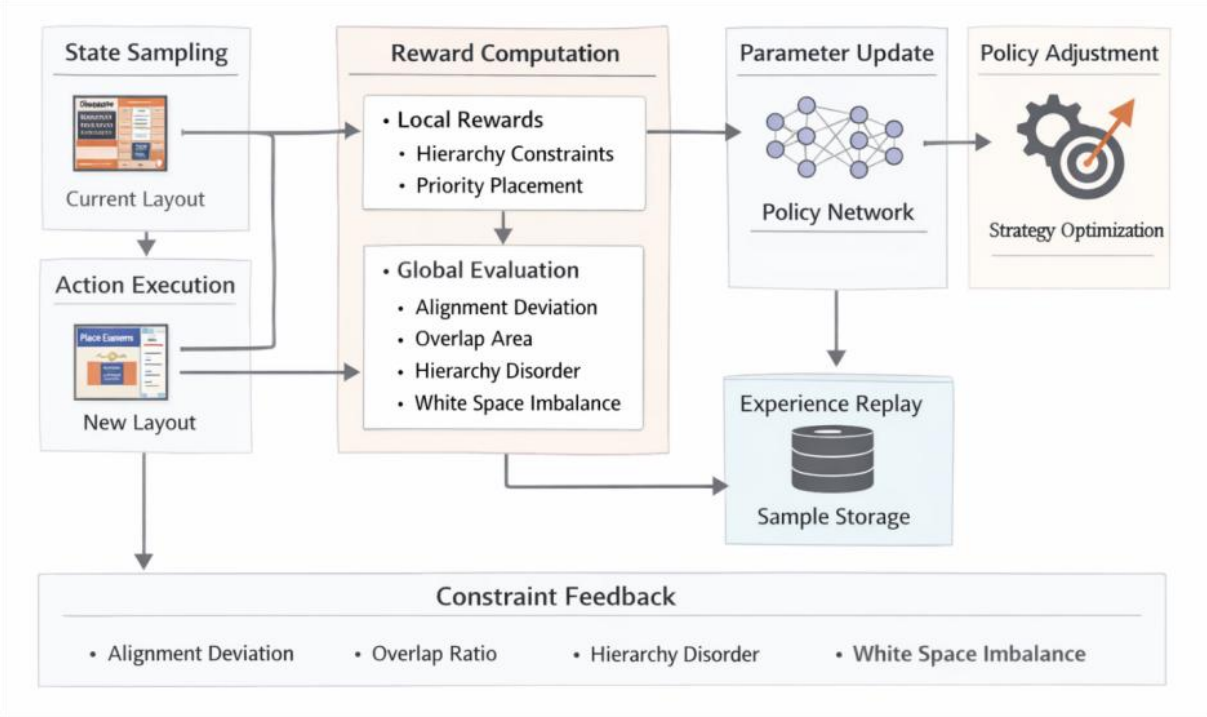

Figure 4: Policy training flowchart for layout optimization in Deep reinforcement learning

In order to constrain the current hierarchical distribution to constantly approach the target template, we construct the following hierarchical consistency loss function:

$$L_c=\sum_{h=1}^{P}\lambda_h\left|y_t^{(h)}-\pi^{(h)}\right|+\eta\sum_{h=1}^{P}y_t^{(h)}\log\frac{y_t^{(h)}}{\pi^{(h)}} \tag{5}$$

Here, L_c represents the level consistency loss, $y_t^{(h)}$ represents the response value of the h level channel at the current time, $\pi^{(h)}$ represents the reference distribution of the corresponding level in the target template, λ_h represents the channel weight, η is the distribution shift penalty coefficient, and P represents the number of level channels. This formula is used to constrain the relative attention order of key elements such as title, main image, price, and button, preventing the model from getting semantically confusing layouts.

Based on the hierarchical loss, we continue to jointly write the hierarchical constraint, aesthetic stability term and policy benefit into the optimization objective, so that the training process not only retains the exploration ability, but also does not deviate from the structural law of the layout task. To this end, this paper adopts the following joint optimization objective:

$$J_p=E\left[\min\left(\rho_t\hat{A}_t,\text{clip}\left(\rho_t,1-\epsilon,1+\epsilon\right)\hat{A}_t\right)-\omega_c L_c+\omega_f F_t\right] \tag{6}$$

where J_p represents the strategy optimization objective, ρ_t represents the action probability ratio between the old and new strategies, $\hat{A}_t$ represents the estimated advantage, ϵ represents the truncation threshold, ω_c represents the hierarchical loss weight, F_t represents the consistency score of typographic style, and ω_f represents the style item coefficient. This formula is used to balance the action benefit, hierarchical order and style stability synchronically when the strategy is updated, so that the training results are more in line with

the layout requirements of commercial posters.

At the implementation level, we introduce conditional embeddings for different style templates to distinguish the hierarchical differences between promotional, brand and event posters. For the samples with a high proportion of the main image, the system increases the weight of the significant branch. For the samples with dense text information, the system increases the weight of semantic branch and geometric branch. In this way, the reinforcement learning model is able to adaptively adjust the hierarchy preference according to the specific scenario, rather than relying on a single rule.

2.3 Analysis of layout action decision and reward feedback mechanism

During layout optimization in reinforcement learning, the action space and reward feedback determine whether the agent can transform the abstract goal into a stable layout step. Poster layout is not a one-time result, but an iterative process around element placement, scale correction, alignment attachment, whitespace adjustment, and level switching. If the action definition is too coarse, the model is prone to search shock. If the reward feedback is too weak, it can be difficult for the model to discern the value of local adjustments. Therefore, the action decision is designed as a hierarchical combination process, and the reward feedback is designed as a multiple joint process, so that the policy network can not only fine-grained control the single step action, but also learn the overall pattern law from the cumulative reward.

As shown in Fig. 5, the agent first reads the current state with the candidate element index, then generates three sub-decisions in the action head: position displacement, scale adjustment and alignment selection, and finally combines them into a single-step layout action. This design is able to reduce the discrete explosion caused by large action search and maintain the interpretability of the policy output. In order to keep the model approaching the target layout during the layout process, the reward function uses multiple joint feedback instead of giving the overall score only at the end.

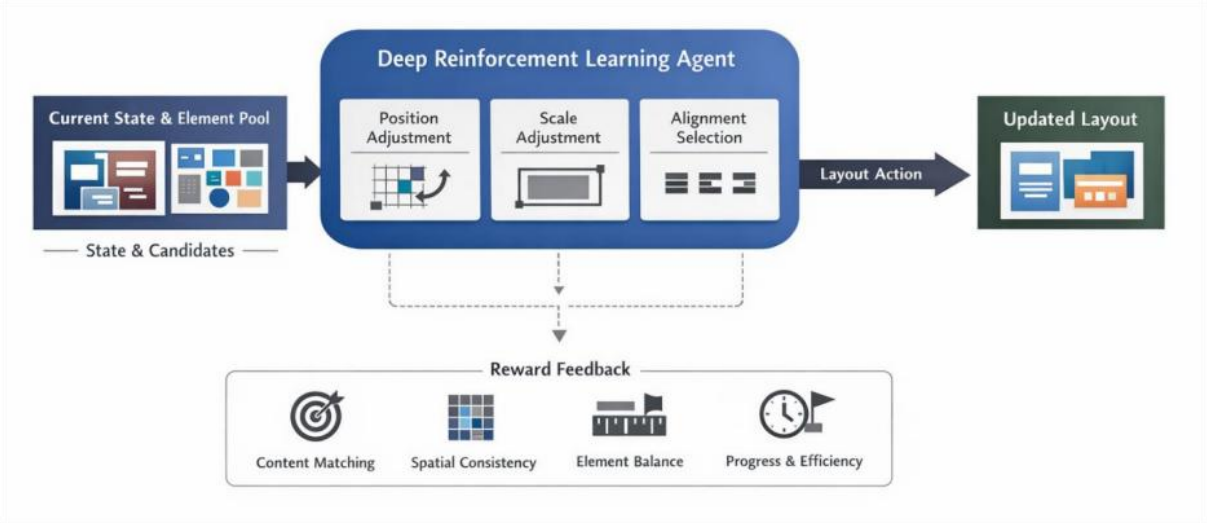

Figure 5: The execution flow diagram of the layout action decision

In order to describe the action selection process, this paper gives the following decomposition formula:

$$P(a_t)=P(l_t \mid s_t)\cdot P(q_t \mid s_t,l_t)\cdot P(b_t \mid s_t,l_t,q_t) \tag{7}$$

where $P(a_t)$ represents the combined action probability at time t, s_t represents the current layout state, l_t represents the position displacement action, q_t represents the scale adjustment action, and b_t represents the alignment selection action. This formula is used to split complex actions into conditional probability chains, thereby reducing the search difficulty of the joint action space and improving the convergence stability of the policy network.

Immediately after an action is executed, the system reads local text readability, visual balance, priority of key elements, and border security, and aggregates these metrics into immediate rewards.

As shown in Fig. 6, the feedback module consists of a readability reward, a balance reward, an overlap penalty, a priority consistency reward, and a boundary security penalty. The readability reward measures the minimum visual size and line spacing adaptation of the text region, the balance reward measures the offset between the visual center and the canvas center, and the priority consistency reward measures whether the element with high semantic weight is located in the high attention region. By writing these feedbacks back after each action step, the model is able to correct layout trajectories with large offsets in time. In the action decision stage, an illegal action shielding mechanism is added. Any actions that cause out-of-bounds, heavy overlap, or occlusion of key elements are filtered by the constraint mask before sampling, thus relieving the policy network of the burden of invalid exploration.

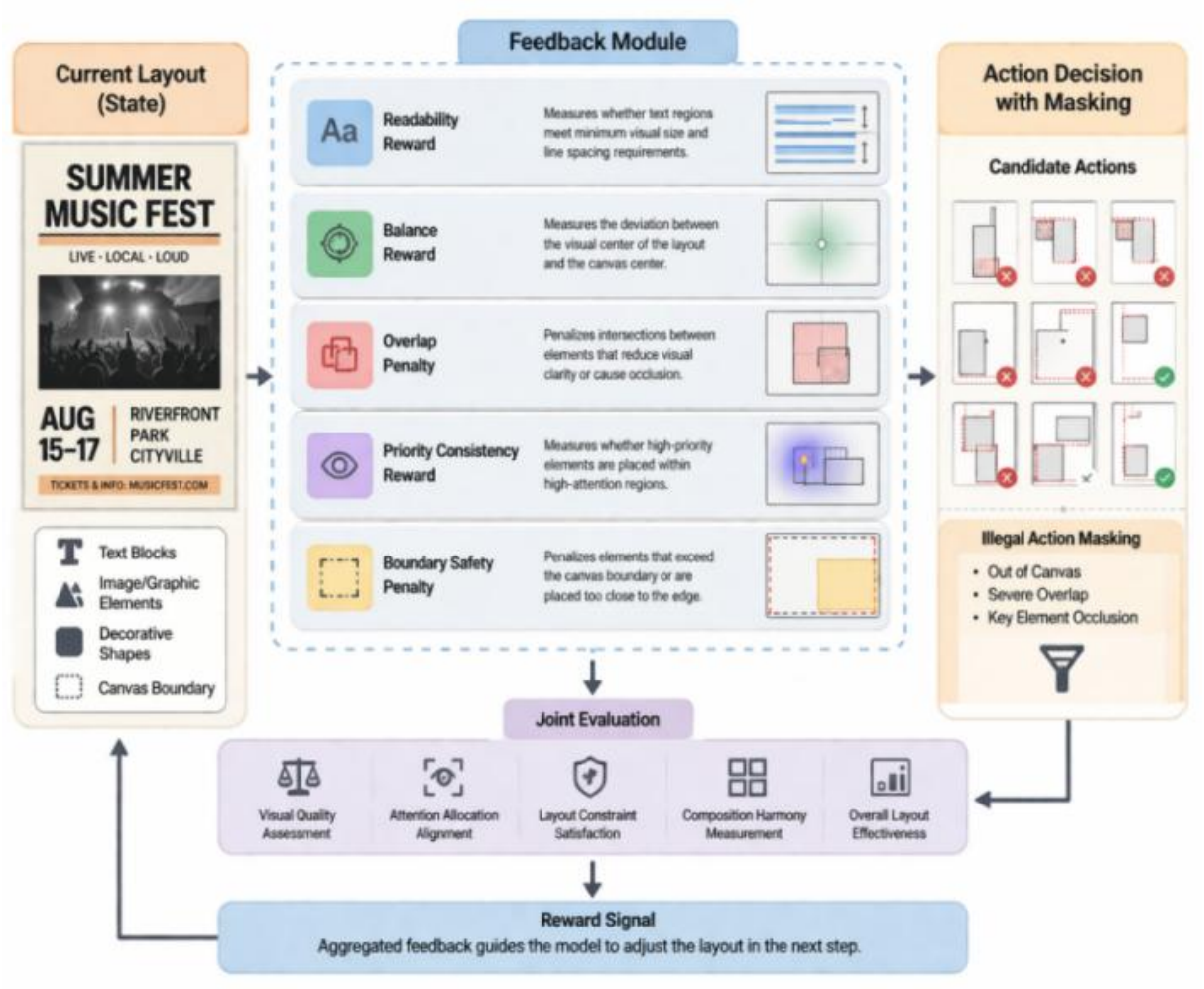

Figure 6: Joint evaluation structure diagram for typographic layout reward feedback

In order to unify the single-step reward calculation, we use the following immediate reward function:

$$R_t=\theta_1 u_t^{read}+\theta_2 u_t^{bal}+\theta_3 u_t^{prio}-\theta_4 v_t^{over}-\theta_5 v_t^{edge}-\theta_6 v_t^{jit} \tag{8}$$

Here, R_t represents the immediate reward at time t, u_t^{bal} represents the readability gain, u_t^{bal} represents the balance gain, u_t^{prio} represents the priority consistency gain, v_t^{over} represents the overlap penalty, v_t^{edge} represents the out-of-bounds penalty, v_t^{jit} represents the local jitter penalty, and θ_1 to θ_6 are the corresponding weights. This formula is used to compress the multi-dimensional layout target into a single scalar reward, so that the reinforcement learning model can obtain an update signal with clear direction.

Single-step rewards can evaluate local adjustments, but poster format focuses more on the overall result accumulated over several steps. Therefore, the system also needs to build cumulative returns, so that the period of quality improvement is continuously written into the long-term evaluation. In order to retain the contribution of small corrections, this paper adds a quality increment term to the traditional discounted return, and forms the following cumulative return expression:

$$G_t=\sum_{\tau=t}^{T}\gamma^{\tau-t}\left(R_\tau+\Delta Q_\tau\right) \tag{9}$$

Here, G_t represents the cumulative return from time t, γ represents the discount factor, T represents the end point of the current trajectory, and ΔQ_τ represents the incremental value of the typographic quality before and after the execution of step τ. This formula is used to enforce small but effective layout corrections, so that the model does not ignore actions that are beneficial to the overall structure but have low single-step scores.

In order to reduce invalid search, this paper adds illegal action mask before action sampling, and directly masks the actions that cause serious occlusion, cross the boundary or the key element is not visible. For the elements that have reached a stable arrangement, the system uses a local freezing mechanism to avoid repeatedly adjusting the same area. For highly sensitive elements such as prices, buttons, and brand slogans, the rewards module additionally sets conversion oriented weights to make it easier to get into the high attention zone. Through the cooperation of action decomposition, immediate reward and cumulative reward, the format optimization process forms a complete closed loop.

3 A poster advertisement layout optimization model based on deep reinforcement learning algorithm

The poster layout optimization model based on deep reinforcement learning algorithm organizes the poster generation process as an iterative policy search task based on the state space, hierarchical constraints and reward feedback established in Chapter 2. Instead of directly output a one-time static layout, the model gradually selects the operation object from the set of elements to be placed, predicts the layout action, and modifies the global composition, so that the main title, main image, price information, description text, and brand logo form a readable, balanced spatial structure with commercial communication direction in the same canvas. Compared with layout methods that only rely on generative sampling, the proposed model emphasizes continuous evaluation and parameter update during action, so it

is easier to maintain structural stability and reasoning consistency under complex samples.

In order to enable the policy network to extract effective deep representations for layout actions from the current layout environment and reduce the offset effect of single-step observation noise on the decision distribution, we define the following layer-aware state projection function.

$$m_{\tau}=\mathrm{LayerNorm}(\Lambda_1\kappa_{\tau}+\Lambda_2\nu_{\tau}+\Lambda_3\varpi_{\tau}+\delta) \tag{10}$$

Here, m_{τ} denotes the main state embedding at time τ, κ_{τ} denotes the geometric compression vector, ν_{τ} denotes the semantic rank vector, ϖ_{τ} denotes the visual saliency response, Λ_1 to Λ_3 are projection matrices, and δ is the bias term. The function of formula (10) is to map the layout information scattered in different channels into a unified expression space and provide stable input for subsequent action search.

In order to continuously write the competition, adsorption and avoidance relations between the placed elements and the elements to be placed into the policy memory, and make the action generation retain the layout context formed by the preorder decision, we further construct the following relational memory update function.

$$n_{\tau}=(1-\upsilon_{\tau})\odot n_{\tau-1}+\upsilon_{\tau}\odot \tanh\left(\Pi_1 m_{\tau}+\Pi_2 \bar{g}_{\tau}\right) \tag{11}$$

Here, n_{τ} represents the relational memory vector, υ_{τ} represents the update gating, Π_1 and Π_2 are transformation matrices, and $\bar{g}_{\tau}$ represents the neighborhood sink features of candidate elements. The function of formula (11) is to write the competition and cooperation between adjacent elements into the continuous memory, so as to avoid losing the structural clues formed by the previous actions in the multi-step arrangement of the strategy.

After the state projection and relation memory are established, the model enters the policy estimation phase. In this stage, the final position is not directly output, but the corresponding layout gain of each candidate action is predicted, and then the actions with large boundary crossing, overlapping and hierarchical dislocation are reduced according to constraint gating.

As shown in Fig. 7, in the training phase, the online policy network firstly generates action candidates, and then the value evaluation branch calculates the proximity of the current layout to the target layout, and then the immediate reward, cumulative quality increment and hierarchical deviation signals are jointly written into the priority experience pool. In the sampling stage, high-information trajectory segments are preferentially extracted to correct text crowding, image-text conflict and visual center deviation in difficult samples. In the inference stage, the optimal policy parameters obtained after training are retained to complete the fast placement without backtracking search. Such a two-stage organization method takes into account the exploration adequacy in the training phase and the execution efficiency in the inference phase, and also enables the model to maintain a relatively stable transfer ability between different poster styles.

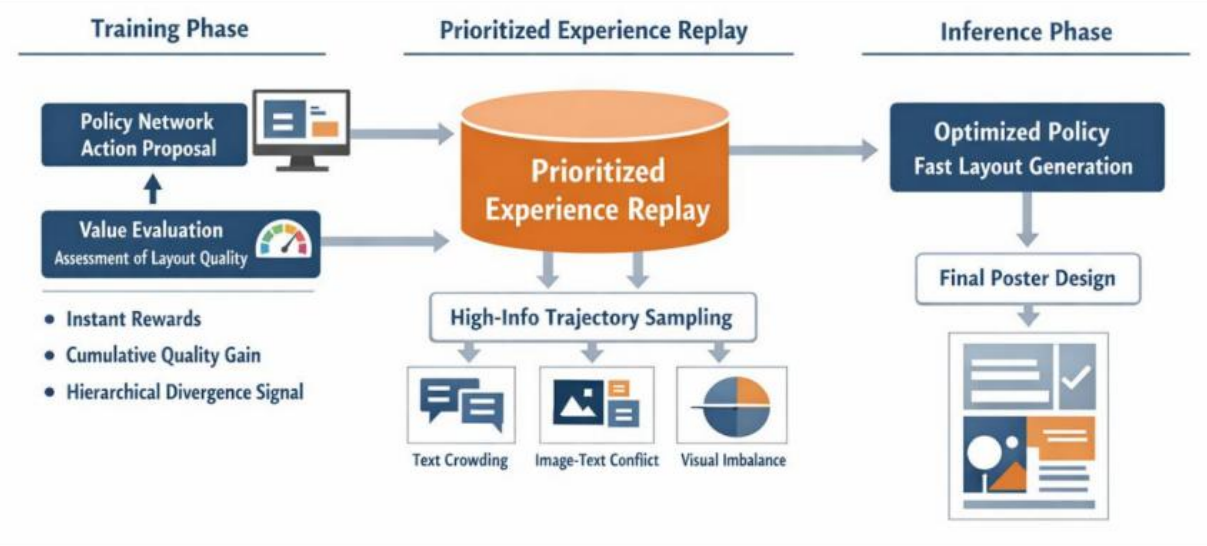

Figure 7: Closed-loop graph of policy update and feedback for the layout layout optimization model

In order to ensure that the action score not only reflects the local geometric changes, but also integrates readability, visual balance and information priority into the value judgment together, this paper adopts the following state-action joint score expression.

$$V_\tau=\xi_1 A_\tau^{leg}+\xi_2 A_\tau^{sym}+\xi_3 A_\tau^{sem}-\xi_4 P_\tau^{col}-\xi_5 P_\tau^{out} \tag{12}$$

Here, V_τ represents the current action value, A_τ^{leg} represents the readability gain, A_τ^{sym} represents the screen balance gain, A_τ^{sem} represents the semantic priority gain,P_τ^{col} and P_τ^{out} represent the overlap penalty and the out-of-bounds penalty respectively, and ξ_1 to ξ_5 are the corresponding weights. The function of equation (12) is to unify the local gain and structural penalty into the same scoring framework, so that the policy update has a clear direction.

Table 2 shows the main components of the layout layout optimization model and their function assignments, and the relevant configurations directly serve the state encoding, action generation and feedback update during training.

Table 2: Layout optimization model core module configuration

Module Name	Input Content	Output Content	Main Function
State Projection Encoder	Geometric features, semantic features, saliency features	Primary state embedding	Unified representation of multi-source layout information
Relational Memory Unit	Primary state embedding, neighborhood features	Relational memory vector	Preserve multi-step layout context
Dual-Head Policy Branch	Current state, candidate action set	Action probabilities and action parameters	Generate decisions for displacement, scaling, and alignment
Value Evaluation Branch	Current state, executed actions	Layout value score	Evaluate the closeness of the current layout to the target arrangement
Prioritized Replay Buffer	Trajectory segments, rewards, errors	Resampled samples	Strengthen learning on hard samples
Target Network Module	Online parameters, soft update coefficient	Smoothed target parameters	Stabilize the training process

In order to make the experience replay more inclined to retain the sample segments with significant structural conflict and high correction value, and enhance the learning ability of the model for local anomalies of complex layout, this paper constructs the following priority sampling weight function.

$$p_j=\frac{(\Delta_j+\epsilon_0)^{\lambda_p}}{\sum_{r=1}^{R}(\Delta_r+\epsilon_0)^{\lambda_p}} \tag{13}$$

Here, p_j represents the sampling probability of the j sample segment, Δ_j represents the temporal difference error of this segment, ϵ_0 represents the smoothing constant, λ_p represents the priority amplification factor, and R represents the total number of samples in the experience pool. The function of equation (13) is to improve the utilization of high-value correction samples, so that difficult actions in complex layouts can be repeatedly learned by the policy network.

In order to incorporate the policy benefit, hierarchical consistency, style stability term and boundary safety constraint into the unified training objective, and make the model continuously maintain the required layout order of commercial posters during the search process, this paper further defines the overall optimization objective as follows.

$$F=E_{j\sim p}\left[\omega_1 l_j+\omega_2\Omega_j+\omega_3\Psi_j+\omega_4\Phi_j\right] \tag{14}$$

Here, F represents the overall training objective, l_j represents the policy payoff term, Ω_j represents the hierarchical consistency term, Ψ_j represents the style stability term, Φ_j represents the boundary safety constraint term, and ω_1 to ω_4 are the corresponding weights. The role of Eq. (14) is to integrate the multiple classes of objectives in the training period into the same optimization framework, so that the model can simultaneously consider structure, style, and enforceability.

In order to further write the mapping between the action selection process and the layout benefit as a learnable probability chain, and ensure that the collaborative search of displacement, scaling and alignment actions can be completed within a unified framework, this paper defines the hierarchical joint distribution of actions as follows.

$$P_\tau^\star=P(a_\tau \mid s_\tau)\cdot P(b_\tau \mid s_\tau,a_\tau)\cdot P(c_\tau \mid s_\tau,a_\tau,b_\tau) \tag{15}$$

Here, $P_\tau^\star$ represents the joint action distribution at time τ, s_τ represents the current layout environment, a_τ represents the displacement action, b_τ represents the scale action, and c_τ represents the alignment action. The function of Eq. (15) is to split the complex actions into chains of conditional probabilities, reduce the difficulty of searching the joint action space, and enhance the interpretability of the policy output.

After the joint modeling is completed, the model can adaptively decide which type of elements to place first, which movement amplitude to use, and when to perform local freezing according to the current canvas state. The title and main image are usually prioritized in the high attention area, the price and button are moved into the sub-center area according to the hierarchical template, and the descriptive text is more responsible for density filling and semantic complement. Since the policy network can re-read the updated layout environment after each action, the layout results do not stay at the level of fixed template copy, but form a dynamic generation process under constraints. At the same time, the module configuration in Table 2 also keeps a clear division of labor of the four types of computing processes: coding, decision, evaluation and playback, which facilitates the implementation of ablation

comparison and complexity analysis in subsequent experimental stages.

4 Experimental data construction

4.1 Process of data acquisition and annotation

The data acquisition process includes the collection of public poster data, commercial promotional image samples and self-built layout design records. The public data mainly comes from the graphic layout data set that can be used for academic research, and the self-built samples come from the layout files organized in e-commerce promotion, brand communication and event poster scenes. All the original samples are uniformly converted into RGB images and structured annotation files, and the image resolution is resampled to a fixed range. The annotation file stores the coordinates, width, height, level and category information of the title, body text, image, logo, price, button and decorative element. In order to ensure that the subsequent reinforcement learning model can read stable input, the boundary of the element is manually checked, and the cross-layer occlusion, text fracture and invalid blank area are corrected twice in the data processing stage. The labeling process adopts a two-stage way of "automatic reservation + manual verification". The system first generates the initial detection box by using the layout analysis program, and then annotators revise the element category, center position and priority labels. Finally, the training samples can be directly sent to the state encoding module. For the case of multiple text blocks in the same poster, the annotation side additionally records the reading order, alignment and primary and secondary relationship for subsequent visual hierarchy modeling. After cleaning, a total of 12480 valid poster samples, 74860 annotation elements and the corresponding layout description files are formed. The data structure can support the computational processes such as state modeling, action learning and reward writeback. In terms of storage mode, each sample retains the original canvas, the normalized canvas, the element index table and the style label file. The style label distinguishes three types of templates: promotion type, brand type and activity type. The label consistency check was crossed by two design coders. When the difference between category judgment and bounding box position exceeded the preset threshold, it was checked by a third party before writing the main library. After processing, the samples are numbered according to a uniform naming rule, and an element-level log is established to track the source of status updates and the path of layout revisions during the training phase.

4.2 Sample screening and evaluation criteria

The sample selection and evaluation criteria focus on the integrity of layout structure, the validity of element annotation and the representativeness of scene. Before the original samples are entered into the main data set, four conditions must be satisfied at the same time: the canvas boundary is complete, the key elements are not less than three categories, the text area is identifiable, and the main body of the image is not severely cropped. For the samples with large area ambiguity, duplicate templates, missing elements or annotation conflicts, the system directly eliminates them in the pre-screening stage. In order to ensure that the training results have a stable comparison basis, the retained samples are re-stratified according to three types of promotion, brand and activity scenes, and the balanced sampling is carried out according to the number of elements, text density and image proportion. Then, the dataset was divided into training set, validation set and test set according to 7:1.5:1.5, and the proportion of categories was the same in the three parts. The evaluation criteria are composed of layout quality score, element alignment accuracy, overlap rate, whitespace balance and average

inference time. The layout quality score is calculated based on the primary and secondary level, visual center of gravity and reading order. Element alignment accuracy is used to measure the consistency of text blocks, image blocks and function labels in horizontal and vertical directions. The overlap rate is used to describe the proportion of invalid occlusion between elements. Whitespace balance is used to measure the distribution difference of blank areas in each quadrant. The average inference time is used to reflect the execution efficiency of the model on a single sample. In order to reduce accidental fluctuations, multiple rounds of repeated statistics are performed on the test results in the experimental stage, and the same hardware and the same batch of samples are used to complete the horizontal comparison. In the screening process, the system also checks the relative position of the title region and the main image region. If both of them fall into the low attention region, the sample will not enter the training set. For pages with too dense text but unclear semantic hierarchy, the annotation side will re-check the priority labels between the title, the body and the button, and confirm that they are correct before writing to the official library. In the statistical stage, the distribution interval of each sample in the number of elements and the ratio of canvas was recorded to ensure that the experimental comparison was based on relatively consistent data.

5 Experimental Results

5.1 Model performance analysis

In order to test the actual effect of the deep reinforcement learning layout model in the poster advertising scene, the experiment is carried out by stratified sampling from the test set according to three types of samples: promotion type, brand type and event type, and statistics are carried out around the layout quality score, element alignment accuracy, overlap rate, white space balance and average inference time. The test results show that the comprehensive layout quality score of the model on 12480 poster samples reaches 91.3, the accuracy of element alignment is 92.6%, the overlap rate is controlled at 1.9%, and the average inference time for a single sample is 0.18 s. The comprehensive scores of promotional, brand and activity samples are 92.1, 90.8 and 91.0, respectively, and the maximum difference between the three types of samples is only 1.3, indicating that the model has a stable structural adaptation ability in different communication scenarios.

In order to further observe the spatial consistency of different element types after the layout is completed, this paper counts the alignment performance of the title, body, main image, button and logo in the three types of samples, and organizes the results as Fig. 8. The average alignment accuracy is 94.2% for the header region, 93.4% for the main image region, 92.8% for the button region, 92.3% for the logo region, and a relatively low 91.1% for the body region. This difference is related to the large deformation amplitude of the text block in multi-column arrangement, line length expansion and local compression. From the hot area distribution, the high accuracy is mainly concentrated on the three key elements of the title, the main image and the button, indicating that the model gives priority to ensuring the spatial stability of the main information and the transformation entrance within the limited action step.

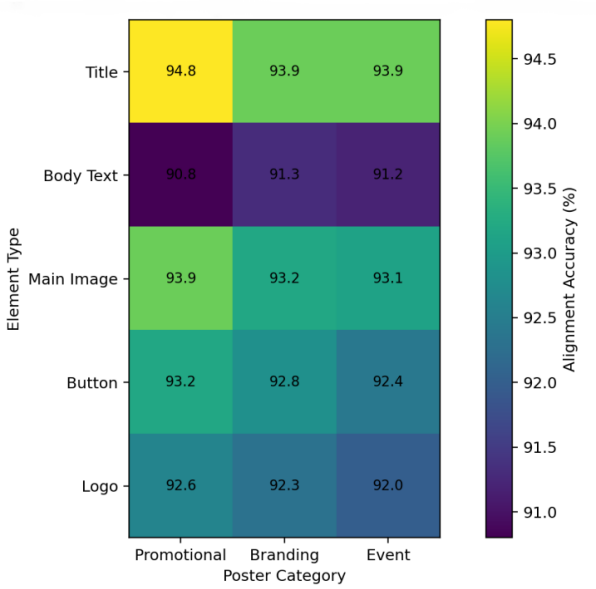

Figure 8: Heatmaps of alignment accuracy for different element types in the three classes of samples

In terms of local conflict control, the inhibitory effect of the model on overlapping area and crowded area is more obvious. In order to present the degree of dispersion of the overlap rate in different style samples, the local conflict statistics of the three types of samples are plotted in Fig. 9. The median overlap was 1.5% for the brand sample, 2.1% for the promotional sample, and 2.0% for the campaign sample. The main body distribution of the three types of samples is concentrated between 1.2% and 2.4%, and the extreme value is up to 4.7%, which mainly appears in the complex layout with multiple text overlapping and a high proportion of main images. This result shows that the model can continue to reduce occlusion conflicts through the action writeback mechanism under the condition of high-density element collocation.

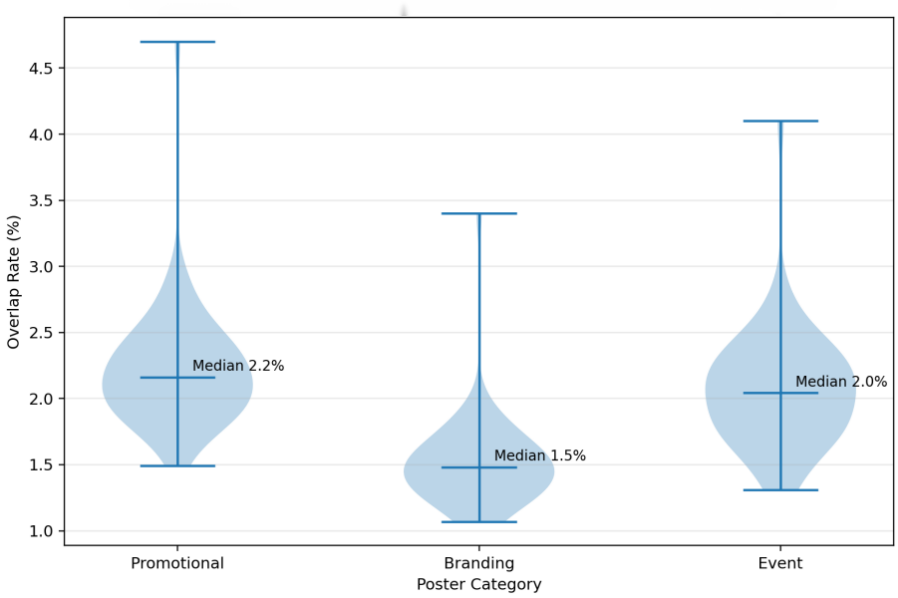

Figure 9: Violin plot of the distribution of the overlap rate of the three classes of style samples

In addition to geometric alignment and local conflict, this paper continues to comprehensively evaluate the model output from four dimensions of readability, balance, priority consistency and reasoning efficiency. The results are shown in Fig. 10, where the readability score is 0.931, the balance score is 0.908, the priority consistency score is 0.917, and the efficiency score is 0.894. The gap of the four indicators was controlled within 0.037, and there was no phenomenon that a single indicator was significantly high and the other dimensions were significantly decreased. This shows that the state coding, hierarchical constraints and reward feedback form a good synergistic relationship after training, and also shows that the model has a stable ability to generate layout in complex poster samples.

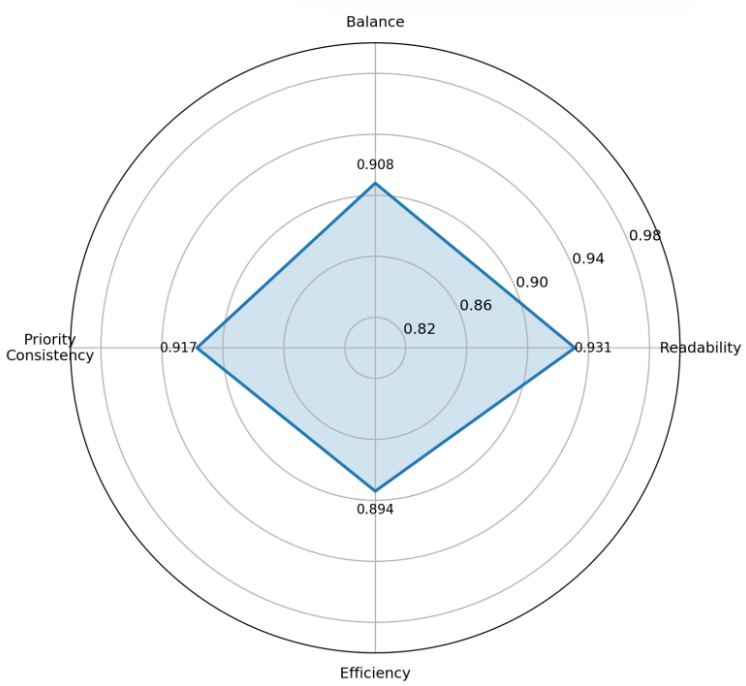

Figure 10: Radar plot of model multidimensional performance metrics

Further observation of the action sequence shows that the model usually takes 11 to 12 steps to complete the arrangement in the simple sample, and takes an average of 15 steps in the high-density promotion sample. Instead of destabilizing the structure, increasing the step size results in more balanced relative spacing between titles, prices, and buttons. The statistical results show that the later stage of the action is mainly used to fine-tune the boundary, safe area and blank ratio, and the early stage is focused on the main image positioning and title anchoring, which indicates that the policy network has formed a phased layout law from main elements to auxiliary elements. At the same time, the number of backoffs in complex samples always remains low.

From the overall results, the current model does not rely on fixed templates to improve local indicators, but forms a continuous computing closed loop between state modeling, hierarchical constraints, action update and feedback writeback. The statistical results of 20 repeated rounds of experiments show that the standard deviation of the comprehensive layout quality score is 0.42, the standard deviation of the alignment accuracy is 0.37, and the standard deviation of the overlap rate is 0.11. For complex samples with eight types of elements, the model still maintains the average blank rate at 18.6%, and makes the visual distance between the main title and the button stable at about 0.214, which provides a reliable basis for subsequent comparison experiments, and also shows that the model can still

maintain a high consistency performance under complex layout conditions.

5.2 Comparative experimental analysis

In order to verify the relative performance of the proposed model under different calculation paths, the rule template method, CNN-Layout, Element-GAN, GNN-Layout and the model in this paper are selected for comparison, and the evaluation is carried out from four dimensions of comprehensive layout quality, alignment accuracy, overlap rate and reasoning time.

Table 3 presents the overall results of the five methods on the unified test set. The results show that the comprehensive layout quality score of the proposed model is 91.3, the alignment accuracy is 92.6%, the overlap rate is 1.9%, and the average inference time is 0.18 s. Compared with the rule template method, the comprehensive score is increased by 12.4 points, the alignment accuracy is increased by 7.9 percentage points, and the overlap rate is decreased by 4.9 percentage points. Although the regular template method is fast, it relies too much on the preset structure, and is prone to rigid arrangement and hierarchical dislocation when facing the change of title length, the fluctuation of image scale and the compression of button area. CNN-Layout improves the local visual feature extraction ability and can alleviate the alignment deviation in simple scenes, but it lacks the continuous constraint on the linkage relationship of multiple elements, and the comprehensive score is still lower than that of the proposed model.

Table 3: Overall performance comparison of different methods

Method	Layout Quality Score	Alignment Accuracy / %	Overlap Ratio / %	Inference Time / s
Rule-Based Template Method	78.9	84.7	6.8	0.05
CNN-Layout	84.2	87.5	4.9	0.12
Element-GAN	86.7	89.8	3.7	0.20
GNN-Layout	88.4	91.1	2.8	0.24
Proposed Model	91.3	92.6	1.9	0.18

Element-GAN has strong generation flexibility and natural layout distribution, but it is strongly dependent on the distribution of training samples. When the text density and the number of decorative elements change greatly, boundary extrusion still occurs in local areas. GNN-Layout improves the collaborative ability with the help of relationship modeling, and the alignment accuracy reaches 91.1%. However, in the process of multiple rounds of position update, without continuous feedback, local errors may still accumulate into overall imbalance. The proposed model performs more balanced on the four indicators, indicating that the reinforcement learning decision chain can continuously revise the layout action according to the current state.

To further observe the contribution of different modules to the quality of the final layout, ablation experiments were continued and the results are shown in Table 4. After removing the visual hierarchy constraint, the comprehensive score drops to 87.8, the consistency of priority is significantly weakened, and the primary and secondary relationship between the title, main image and main text is no longer stable. After removing the action writeback mechanism, the alignment accuracy decreases to 89.6%, and the overlap rate increases to 3.1%, which indicates that if the results of the single step action cannot be sent back in time, it is difficult to repair the local deviation continuously for subsequent adjustment. After removing the relational memory unit, the local conflicts in multiple text samples are significantly increased, especially in the format with dense description information, the compression of adjacent

regions is more likely to occur. The three results show that hierarchical constraints determine the primary and secondary order, action writeback affects local correction, and relational memory unit maintains cross-element cooperation.

Table 4: Results of model ablation experiments

Model Configuration	Layout Quality Score	Alignment Accuracy / %	Overlap Ratio / %
Full Model	91.3	92.6	1.9
Without Visual Hierarchy Constraints	87.8	89.9	2.7
Without Action Write-Back Mechanism	88.5	89.6	3.1
Without Relational Memory Unit	89.1	90.4	2.8

In addition to the overall comparison and ablation, the experiment also statistics the cross-scene generalization ability of the three types of style samples. Table 5 shows that the proposed model scores 92.1, 90.8 and 91.0 on the promotional, brand and activity samples, respectively, with a maximum difference of only 1.3, which is significantly smaller than the other baseline models. The results show that the model does not rely on a single style rule, but learns a stable layout organization. Promotional samples emphasize main headings and button areas, brand samples place more emphasis on white space pacing, and campaign samples tend to contain more explanatory information. The three types of samples have obvious differences in visual center of gravity and information level, while the proposed model can still maintain close scores, indicating that the method has good adaptability under the condition of style switching.

Table 5: Layout quality scores for different style scenarios

Method	Promotional Type	Brand Type	Event Type
Rule-Based Template Method	80.4	77.3	79.0
Element-GAN	87.5	85.2	86.1
GNN-Layout	89.0	87.6	88.2
Proposed Model	92.1	90.8	91.0

Based on the above, it can be seen that the advantages of the proposed model are not only reflected in the improvement of single indicators, but also reflected in the better balance between layout quality, alignment accuracy, overlap control and reasoning efficiency. This indicates that the constructed state modeling, hierarchical constraint and feedback update mechanism can jointly support stable layout output in complex scenes. It also provides a reliable experimental basis for the subsequent automatic deployment of real poster design process, and has further engineering transformation space.

6 Discussion

The existing layout methods have their applicable boundaries in the task of poster advertisement generation. The regular template method relies on preset raster and human experience, and can maintain fast reasoning speed when the number of elements is small. However, when the title, main image, price, and button enter the canvas at the same time, the fixed template is difficult to continue to deal with local congestion and hierarchical conflicts, so the overall layout quality score is low. CNN-Layout can learn local layout rules from samples and has a good fitting ability for common image-text combinations. However, the

convolutional receptive field is still limited in the expression of long-distance element relationships, and its performance in cross-region alignment and center of gravity coordination is not stable. Element-GAN strengthens the sense of integrity in the generation stage, but it is easy to overlap and recover under high-density text conditions. GNN-Layout captures the structural relationship more fully, but the inference cost is high. Compared with these methods, the proposed model organizes state modeling, hierarchical constraints, action decision and reward feedback into a continuous closed loop, so that layout optimization no longer stops at static sampling, but forms a decision process that can be iteratively modified. The experimental results show that the model achieves better performance in comprehensive layout quality, alignment accuracy and overlap control at the same time, which indicates that the visual hierarchy constraint and action writeback mechanism have a direct role in maintaining the primary and secondary order in commercial posters. Further, the price tag and button area in the promotional sample are more sensitive to local adsorption, the logo and white space in the brand sample are more dependent on global balance, and the active sample emphasizes more on the reading channel between the title and the main image. The method in this paper can maintain similar scores in the three scenarios, which indicates that the model has a certain cross-template generalization ability, and also shows that deep reinforcement learning has high engineering application value in the layout generation task. At the same time, the relational memory unit reduces the number of backoffs in complex samples, making the inference link more stable, the output more consistent and easier to reproduce.

7 Conclusions

Focusing on the task of poster advertisement layout, this paper constructs an intelligent optimization model based on deep reinforcement learning. The model takes such elements as title, body text, main image, logo, price and button as objects, and expresses the layout generation as a constrained sequential decision process, which is continuously optimized through state encoding, visual hierarchy constraint, action combination and reward feedback. Experimental results show that the comprehensive layout quality score of the proposed model is 91.3 on 12480 samples, the alignment accuracy is 92.6%, the overlap rate is reduced to 1.9%, and the average reasoning time is 0.18 s, which indicates that the proposed method can ensure the layout quality while maintaining good execution efficiency. Compared with the regular template method, CNN-Layout, Element-GAN and GNN-Layout, the proposed method shows more stable primary and secondary relationship control ability and more balanced space allocation ability in complex scenes. The limitations of this paper are also relatively clear. Although the current training samples cover three types of promotion, brand and event scenes, the data size under the conditions of cross-cultural visual style, abnormal format and ultra-long text is still limited, which leads to the adaptability of the model to extreme format still has room for improvement. Reward functions have incorporated readability, balance, and priority consistency, but color linkage, font sentiment, and fine-grained aesthetic preferences are still not adequately represented. Future research will continue to expand the multi-style dataset, introduce cross-modal semantic alignment and aesthetic preference modeling mechanisms, and further strengthen the support for complex text structures, dynamic graphic content and interactive poster scenes. At the same time, the lightweight deployment strategy will be combined to improve the reasoning efficiency and transfer ability of the model in the actual digital design system. In addition, the human-machine collaborative correction interface will be explored, so that the automatically generated results can be quickly finalized and style fine-tuned with a small amount of manual intervention.

References

[1] Li C, Zhang P, Wang C. Harmonious textual layout generation over natural images via deep aesthetics learning[J]. IEEE Transactions on Multimedia, 2021, 24: 3416-3428.

[2] Chen L, Jing Q, Zhou Y, et al. Element-conditioned GAN for graphic layout generation[J]. Neurocomputing, 2024, 591: 127730.

[3] Chen L, Jing Q, Tsang Y, et al. Iris: a multi-constraint graphic layout generation system[J]. Frontiers of Information Technology & Electronic Engineering, 2024, 25(7): 968-987.

[4] Wu X, Yao S, Zhang Z, et al. Generate custom travel magazine layouts[J]. Intelligent Data Analysis, 2024, 28(3): 825-840.

[5] Cheng S, Sheng D, Yao J, et al. Poster graphic design with your eyes: An approach to automatic textual layout design based on visual perception[J]. Displays, 2023, 79: 102458.

[6] Kakooee R, Dillenburger B. Reimagining space layout design through deep reinforcement learning[J]. Journal of Computational Design and Engineering, 2024, 11(3): 43-55.

[7] Wang L, Liu J, Zeng Y, et al. Automated building layout generation using deep learning and graph algorithms[J]. Automation in Construction, 2023, 154: 105036.

[8] Aalaei M, Saadi M, Rahbar M, et al. Architectural layout generation using a graph-constrained conditional Generative Adversarial Network (GAN)[J]. Automation in Construction, 2023, 155: 105053.

[9] Jiang F, Ma J, Webster C J, et al. Building layout generation using site-embedded GAN model[J]. Automation in construction, 2023, 151: 104888.

[10] Cheng W, Shan Y. Learning layout generation for virtual worlds[J]. Computational Visual Media, 2024, 10(3): 577-592.

[11] Yao Z, Chen Y, Cui J, et al. Conditional room layout generation based on graph neural networks[J]. Computers & Graphics, 2024, 122: 103971.

[12] Zhong Y, Hempel S, Geiger A, et al. Automated building layout generation: Implementation and comparison of STREAMER Early Design Configurator and SDaC Layout Designer[J]. Journal of Building Engineering, 2024, 95: 110163.

[13] Wu W P, Feng Y. Interior space design and automatic layout method based on CNN[J]. Mathematical Problems in Engineering, 2022, 2022(1): 8006069.

[14] Shi Y, Shang M, Qi Z. Intelligent layout generation based on deep generative models: A comprehensive survey[J]. Information Fusion, 2023, 100: 101940.

[15] Banerjee A, Biswas S, Lladós J, et al. SemiDocSeg: harnessing semi-supervised learning for document layout analysis[J]. International Journal on Document Analysis

and Recognition (IJDAR), 2024, 27(3): 317-334.

[16] Gemelli A, Marinai S, Pisaneschi L, et al. Datasets and annotations for layout analysis of scientific articles[J]. International Journal on Document Analysis and Recognition (IJDAR), 2024, 27(4): 683-705.

[17] Peña A, Morales A, Fierrez J, et al. Continuous document layout analysis: Human-in-the-loop AI-based data curation, database, and evaluation in the domain of public affairs[J]. Information Fusion, 2024, 108: 102398.

[18] Vesalainen A, Tolonen M, Ruotsalainen L. Document Layout Error Rate (DLER) metric to evaluate image segmentation methods[J]. Machine Learning with Applications, 2024, 18: 100606.

[19] Wu X, Ma T, Du X, et al. DRFN: A unified framework for complex document layout analysis[J]. Information Processing & Management, 2023, 60(3): 103339.

[20] Wu X, Xiao L, Du X, et al. Cross-domain document layout analysis using document style guide[J]. Expert Systems with Applications, 2024, 245: 123039.