Computer Science and Information Systems 00(0):0000–0000 https://doi.org/10.2298/CSIS260330052Z

Paper Type: Research Paper

End-to-End Prior-Guided Graph Attention Network with Differentiable Structural Priors for Interpretable Mobile Interface Layout Scoring

Long Zeng[1], Zi Wang[2], Wen Zuo[3], and Jun Hoong Chan[4]

[1] School of Information Engineering, Chenzhou Vocational Technical College
Chenzhou 423000, China
zenglong00000@163.com
[2] School of Information Engineering, Chenzhou Vocational Technical College
Chenzhou 423000, China
wangzi1120@163.com (corresponding author)
[3] School of Fine Arts and Artistic Design, Hunan University of Science and Engineering
Yongzhou 425199, China
64012465021@msu.ac.th
[4] School of Computer Science, Peking University
Beijing 100871, China
junhoong95@stu.pku.edu.cn

Abstract. Automated assessment of mobile-interface aesthetics remains difficult because professional judgments depend on spatial relations among components, whereas image-only models mainly summarize global appearance. This study develops an interpretable end-to-end scorer for static mobile screenshots. The proposed prior-guided graph attention network combines a YOLOv10 component detector, a ResNet-50 visual stream, and a graph stream whose attention is modulated by differentiable priors for proximity, alignment, white space, and touch-target accessibility. Comparisons with convolutional, graph, and transformer baselines were conducted on app-level splits of the MUD and CLAY datasets. On MUD, the method obtained a weighted root mean square error of 0.387 and a Spearman rank correlation of 0.856, improving the strongest convolutional baseline by 14.3%. Ablation, detector-sensitivity, cross-dataset, and expert blind-test analyses showed that the structural priors improve accuracy, transferability, and diagnostic agreement while retaining end-to-end latency of 35 milliseconds per screen.

Keywords: Mobile Interface Aesthetics, Graph Attention Network, Structural Priors, Interpretable Regression, Accessibility

1. Introduction

A mobile-interface screenshot is a rendered visual artifact whose perceived quality depends on both pixel appearance and the spatial organization of interface components. Users form judgments of trustworthiness, modernity, and usability within the first moments of exposure, before completing an interaction; alignment, spacing regularity, hierarchy, and white-space partitioning strongly shape that judgment [1,2]. Automated scoring

of static layouts can therefore support rapid design review during compressed release cycles, provided that the score reflects relations among components rather than only global image statistics.

The scope of this study is the static visual layout of a single mobile screen. Transition animation, scrolling behavior, haptic feedback, and micro-interaction timing belong to the broader interaction-quality problem but are not observable in one raster frame. The target is the pre-interaction impression produced by component geometry, grouping, balance, visual treatment, and the accessibility of visible touch targets.

Existing approaches leave an important methodological gap. Rule-based systems measure grid fit, symmetry, overlap, contrast, or margin uniformity, but independent penalties cannot represent trade-offs among layout cues [3,4,5]. Convolutional and transformer models learn perceptual representations from screenshots [6,7,8,9,10,11,12], yet spatial pooling can obscure the pairwise relations that distinguish a disciplined card grid from a visually similar but weakly aligned arrangement. Modern UI resources, including MUD, CLAY, and UICrit, also make leakage-resistant app-level evaluation increasingly important [13,14,15].

Graph neural networks offer an appropriate relational representation because detected components can be modeled as nodes and their geometric relations as edges [16,17,18,19]. Two limitations nevertheless remain in earlier graph-based UI pipelines. First, an external parser can introduce fragmentation and localization errors that cannot be corrected by the scoring loss. Second, latent graph attention does not directly expose the design principles that practitioners use when diagnosing a layout.

The research question is therefore: can an end-to-end scorer that jointly learns component detection and prior-guided graph reasoning improve aesthetic prediction and produce expert-aligned diagnostics relative to image-only and latent-relation baselines? The objective is to test this question on contemporary mobile-interface benchmarks while preserving practical inference cost and transparent attribution.

The proposed architecture, termed PAA-GAT, couples a YOLOv10-S component detector with a dual-stream regressor. A ResNet-50 branch captures global appearance, while a graph-attention branch reasons over detected components. Four differentiable priors - proximity, alignment, white space, and touch-target accessibility - modulate the attention logits. The detector, prior coefficients, graph layers, fusion module, and regression head are optimized under a joint objective.

The main contributions are as follows:

1. End-to-end detection and scoring. Continuous YOLOv10 candidate boxes, class probabilities, and confidence gates are connected to the scoring objective, allowing localization parameters to receive regression gradients while avoiding a frozen parsing bottleneck.
2. Differentiable structural reasoning. Four explicit priors are incorporated into graph attention. The touch-target prior extends aesthetic assessment toward interaction ergonomics by operationalizing the WCAG 2.2 target-size requirement [20].
3. Leakage-resistant evaluation. The method is evaluated on app-level splits of MUD and CLAY, with Rico retained as a legacy comparison, and is examined through ablation, detector sensitivity, hyperparameter sensitivity, cross-dataset transfer, and backbone substitution.

4. Actionable interpretability. A four-dimensional diagnostic vector decomposes each prediction into proximity, alignment, white-space, and touch-target contributions, and its rank agreement with senior designer judgments is assessed in a double-blind study.

On MUD, PAA-GAT obtains a weighted root mean square error of 0.387 and a Spearman rank correlation of 0.856. The error is 14.3% lower than that of the strongest convolutional baseline, while the Gestalt Structure Integrity Ratio increases to 0.671. The expert study reports dimension-wise correlations from 0.76 to 0.84.

The remainder of the paper is organized as follows. Section 2 reviews heuristic, perceptual, graph-based, and hybrid optimization approaches. Section 3 presents the detector-scorer architecture and joint objective. Section 4 describes the datasets, protocols, statistical analyses, and experimental results. Section 5 discusses conclusions, limitations, practical implications, and future work.

2. Related Work

2.1. Heuristic UI Layout Analysis

Early interface-evaluation systems translated design knowledge into deterministic measurements of alignment, grid regularity, contrast, overlap, target spacing, and reading-order continuity [3,4]. These methods remain useful for rapid screening because each violation has a direct interpretation.

Their principal weakness is not the absence of design knowledge but the additive treatment of coupled constraints. A deliberate asymmetry supported by strong column alignment can be penalized like a disordered layout, even though practitioners judge the two compositions differently. Learned visual-principle detectors have shown that interactions among design cues can be more informative than independent threshold violations [5].

Device and layout diversity further challenges fixed thresholds. Responsive dashboards, foldable-screen compositions, and design-system variants can depart from canonical grids without appearing defective. Mobile-UI-Repair further shows that learned, device-aware analysis can identify and localize interface display smells [21]. The present method retains interpretable design relations but represents them as differentiable priors whose influence is calibrated jointly with visual evidence.

2.2. CNN- and Transformer-Based Aesthetic Assessment

Deep learning shifted aesthetic assessment from hand-crafted features to learned representations. Convolutional regressors and transfer-learning systems have been applied to Android screenshots and UI preference prediction [6,7,8]. Calista combines website visual and content-level signals [9], while ensemble studies combine deep outputs with engineered interface features [10]. General image-aesthetic models such as NIMA and MUSIQ provide additional perceptual baselines [11,12].

UI-specific datasets have also expanded the available supervision. UICrit provides expert component-level critiques [13], MUD supplies modern, noise-filtered interfaces for design modeling [14], and CLAY offers denoised layout structures derived from Rico

[15,22]. These resources support stricter app-level partitioning, which reduces visual leakage from screens belonging to the same application.

Image encoders remain strong at color, texture, typography, and global composition, but their pooled features do not explicitly preserve every pairwise geometric relation. Attention can recover spatial sensitivity, yet a low score is still difficult to decompose into alignment, grouping, white-space, or touch-target causes. PAA-GAT addresses this gap by coupling global visual features with an explicit component graph.

2.3. Graph Neural Networks in Visual Reasoning

Graph neural networks encode relational structure through neighborhood aggregation and attention [16,17,18,23,24,25]. Layout-aware graph convolution has demonstrated the value of hierarchical spatial relations for general image-aesthetic assessment [19], while heterogeneous attention, semantic grouping, UI similarity, and screen-recognition studies show that typed or structural relations are useful for interface understanding [26,27,28,29].

The proposed method differs from these models in four respects: it injects explicit differentiable priors into attention logits; it connects a YOLOv10 detector to the regression loss; it fuses topology and global appearance rather than replacing one with the other; and it produces a four-dimensional diagnostic vector intended for practitioner-facing feedback.

2.4. Hybrid Metaheuristic Optimization and Deep Models

Recent computer-vision studies use metaheuristics to search model configurations or tune learning systems. Lightweight YOLOv8 architectures have been examined for circuit-board defect detection [30], and a modified crayfish optimization algorithm has been combined with convolutional networks, AdaBoost, and XGBoost for waste classification [31]. Such approaches can improve model selection when the search space is discrete, non-convex, or expensive to differentiate.

PAA-GAT occupies a different position in the hybrid-method landscape. Its design knowledge is not optimized by a population-based outer loop; instead, prior coefficients are embedded in the attention operator and updated by gradient descent under the task loss. Metaheuristics could be added in future work for architecture search or deployment-aware multi-objective tuning, but they are not required for the present end-to-end formulation.

3. Methodology

Figure 1 summarizes the end-to-end architecture. A mobile screenshot is processed by a YOLOv10-S detector that yields continuous candidate boxes, confidence scores, and class probabilities. The same detections support two parallel pathways. The global vision stream extracts a 128-dimensional descriptor from ResNet-50. The local topology stream constructs a component graph, applies two prior-guided attention layers, and returns a 128-dimensional topology descriptor together with a four-dimensional diagnostic vector. Late fusion maps the concatenated 256-dimensional representation to the final aesthetic score.

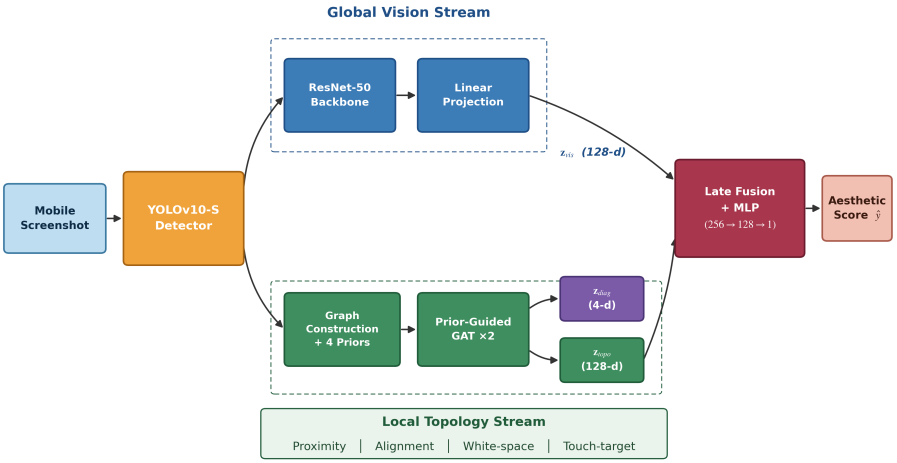


Fig. 1. End-to-end architecture of PAA-GAT. A YOLOv10 detector produces interface components for a global ResNet-50 stream and a prior-guided topology stream; their 128-dimensional descriptors are fused for aesthetic-score regression

3.1. End-to-End UI Component Detection via YOLOv10

The earlier parsing pipeline used a U2-Net salient-object detector [32] followed by mask-to-box conversion, connected-component processing, and non-maximum suppression. That arrangement introduces boundary quantization, separates parsing from score optimization, and adds computation before graph construction.

The revised pipeline uses YOLOv10-S [33]. Its consistent dual-assignment strategy provides an NMS-free one-to-one inference head. Each candidate contains normalized box coordinates (x, y, w, h), an object-confidence value, and logits for eight interface-component categories: text, icon, image, card, input field, toolbar, button, and decorative element.

Detector development uses 6,000 annotated MUD screens: 5,400 for detector training and 600 for detector validation. The detector-development pool is completely disjoint from all 4,200 aesthetic-scoring screens and from their application identifiers. A COCO-pretrained YOLOv10-S checkpoint is first warmed for 20 epochs; this detector-only checkpoint reaches an mAP@0.5 of 0.92 on the held-out 600-screen detector-validation set. After joint detector-scorer optimization, the final validation-selected checkpoint reaches an mAP@0.5 of 0.95 on the same detector-validation set. This 0.95 checkpoint is used for the final MUD results reported in the abstract, Table 2, and Section 5. Text, button, and toolbar each exceed 0.90 average precision, while the heterogeneous decorative category reaches 0.78.

During joint optimization, graph nodes are formed from a fixed-size set of one-to-one candidate boxes before hard confidence thresholding. Box coordinates, object confidence, and class logits remain continuous; confidence acts as a soft gate for node and edge aggregation. Structural priors and attention logits are computed from these values without

stop-gradient operations. The discrete fixed-K routing step and evaluation metrics are not differentiated, so end-to-end denotes joint detector-scorer optimization through the box and logit pathways. The detector learning rate is one hundredth of the scorer learning rate to limit catastrophic forgetting.

3.2. Structural Priors with Touch-Target Accessibility

Four differentiable edge-weight functions encode complementary aspects of layout structure. Proximity, alignment, and white space formalize recurring Gestalt relations, while touch-target accessibility connects visible geometry to interaction ergonomics. All coordinates and dimensions are normalized by screenshot size unless otherwise stated.

Proximity. For nodes i and j, spatial grouping is defined from their centroid vectors by:

$$w_{ij}^{(\mathrm{prox})} = \exp\left(-\frac{\|c_i - c_j\|_2^2}{2\sigma_d^2 + \varepsilon}\right) \tag{1}$$

where the screen-scale distance parameter is the standard deviation of pairwise centroid distances, the norm is Euclidean, and epsilon is fixed at 10^{-6} for numerical stability.

Alignment. Let the anchor set contain the normalized left, right, horizontal-center, top, vertical-center, and bottom coordinates of each component. The strongest anchor match is:

$$w_{ij}^{(\mathrm{align})} = \max_{a \in \mathcal{A}} \exp\left(-\frac{|a_i - a_j|}{\sigma_a}\right) \tag{2}$$

In Equation (2), $\mathcal{A}$ denotes the six anchor types (left, right, horizontal center, top, vertical center, and bottom); a_i and a_j are the corresponding normalized coordinates of nodes i and j; and σ_a is the within-screen standard deviation of all absolute pairwise differences for the selected anchor type. The maximum therefore retains the closest anchor match.

White space. Negative space separates visual groups, supports reading order, and maintains hierarchy [2]. For the minimal rectangle enclosing nodes i and j, the prior is:

$$w_{ij}^{\mathrm{ws}} = \frac{A_{\mathrm{neg}}(B_{ij})}{A_{\mathrm{box}}(B_{ij}) + \varepsilon} \tag{3}$$

In Equation (3), B_{ij} is the minimal axis-aligned rectangle enclosing nodes i and j; $A_{\mathrm{box}}(B_{ij})$ is its total area; $A_{\mathrm{neg}}(B_{ij})$ is the portion of B_{ij} not covered by any detected component box after all boxes are clipped to B_{ij}; and ε is 10^{-6}. Thus, the white-space prior is a normalized empty-area ratio in the interval from zero to one.

Touch-target accessibility. WCAG 2.2 Success Criterion 2.5.8 specifies a minimum target size of 24 by 24 CSS pixels at Level AA, subject to stated exceptions [20]. The constant 24 therefore represents the normative minimum target dimension, not a tuned model parameter. For node i, let its shortest box side be s_i and let τ be 24 divided by $\min(W_s, H_s)$, where W_s and H_s are the screenshot width and height. The differentiable node-compliance score is:

$$\phi_i = \sigma\left(\frac{s_i - \tau}{\sigma_{\mathrm{acc}}}\right) \tag{4}$$

where $\sigma(\cdot)$ is the logistic sigmoid and the sharpness parameter is fixed at 0.005. Non-interactive components receive a compliance value of one. Pairwise accessibility combines target size and separation:

$$w_{ij}^{(\mathrm{acc})} = \phi_i \phi_j \max\left(0, 1 - \frac{\mathrm{ReLU}\big(\delta_{\min} - \mathrm{gap}_{ij}\big)}{\delta_{\min}}\right) \tag{5}$$

Here, gap_{ij} is the minimum edge-to-edge distance between the two component boxes. The constant 8 represents a common base spacing token used to create a smooth separation margin; $\delta_{\min}$ is therefore 8 divided by $\min(W_s, H_s)$. It operationalizes the WCAG separation exception for optimization but is not presented as an additional WCAG threshold. ReLU is the rectified linear unit.

$$\mathrm{Prior}_{ij} = \beta_p w_{ij}^{(\mathrm{prox})} + \beta_a w_{ij}^{(\mathrm{align})} + \beta_w w_{ij}^{(\mathrm{ws})} + \beta_t w_{ij}^{(\mathrm{acc})} \tag{6}$$

The four β coefficients specify the relative contributions of proximity, alignment, white space, and touch-target accessibility. They are parameterized by a softmax so that they remain non-negative and sum to one. The separate scalar λ in Equation (7) controls the global strength of all prior evidence relative to learned feature compatibility.

3.3. Dual-Stream Backbone

The global vision stream processes 224 by 224 renderings with an ImageNet-1K-pretrained ResNet-50 [34]. Global average pooling and a linear projection produce a 128-dimensional visual descriptor. The topology stream forms each node representation from geometry, categorical embeddings, detector confidence, and RoI-aligned visual patches, then projects the concatenated attributes to 128 dimensions. Two prior-guided attention layers produce a 128-dimensional topology descriptor and the four-dimensional diagnostic vector. Late fusion concatenates the visual and topology descriptors so that global appearance and local structure remain separately identifiable before regression.

3.4. Prior-Guided Attention and Regression Head

For each ordered node pair (i,j), the attention operator augments learned feature compatibility with the combined structural prior:

$$e_{ij}^{(\ell,k)} = \mathrm{LeakyReLU}\Big(a^{(\ell,k)\top}\big[\mathbf{W}^{(\ell,k)} h_i^{(\ell)} \,\|\, \mathbf{W}^{(\ell,k)} h_j^{(\ell)}\big] + \lambda\, \mathrm{Prior}_{ij}\Big) \tag{7}$$

In Equation (7), $e_{ij}^{(\ell,k)}$ is the unnormalized attention logit from node i to neighbor j at graph layer ℓ and head k; $h_i^{(\ell)}$ and $h_j^{(\ell)}$ are the input node embeddings at that layer; $\mathbf{W}^{(\ell,k)}$ is the learned projection matrix; $a^{(\ell,k)}$ is the learned attention vector; and the double bar denotes concatenation. LeakyReLU uses a negative slope of 0.2, and λ controls the global strength of the structural prior. The projection matrix W is typeset in bold in the equation and is distinct from the screenshot width W_s. Neighborhood softmax produces normalized attention coefficients. Four heads are used. The 256-dimensional fused descriptor is passed through a 256-128-1 multilayer perceptron with dropout of 0.3. To account for disagreement among raters, each sample receives the weight:

$$w_i = \frac{1}{\sigma_{r,i}^2 + \varepsilon} \tag{8}$$

In Equation (8), $\sigma_{r,i}$ is the standard deviation of the seven raw expert ratings assigned to screen i, $\sigma_{r,i}^2$ is the corresponding inter-rater variance, and ε is 10^{-6}. The weighted score-regression objective is:

$$L_{\text{wreg}} = \frac{\sum_i w_i \left(y_i - \widehat{y}_i\right)^2}{\sum_i w_i} \tag{9}$$

In Equation (9), B is the mini-batch, y_i is the trimmed-mean human rating for screen i, $\widehat{y}_i$ is the predicted aesthetic score, and w_i is the reliability weight from Equation (8).

3.5. End-to-End Objective and Training Protocol

For a mini-batch B, the complete model jointly optimizes detector parameters, graph-attention layers, relative prior coefficients, the fusion module, and the score head under:

$$L_{\text{total}} = L_{\text{wreg}} + \alpha L_{\text{det}} + \gamma \|\boldsymbol{\beta}\|_2^2 + \eta \|\Theta\|_2^2 \tag{10}$$

In Equation (10), the detector term contains YOLOv10 box-regression, classification, and distribution-focal losses; the regression term is defined in Equation (9); $\boldsymbol{\beta}$ is the vector of relative prior coefficients; and Θ contains the remaining trainable parameters. Validation tuning sets α to 1.0, γ to 10^{-4}, and η to 10^{-5}. The first coefficient balances detection and scoring, the second regularizes prior coefficients, and the third applies L2 regularization to the remaining parameters.

Training follows two stages. The detector is warmed for 20 epochs with the detector loss. The complete system is then trained for 180 epochs with the joint objective. Candidate boxes, class probabilities, confidence gates, graph construction, structural-prior calculation, and score regression remain connected except for fixed-K routing. The checkpoint with the lowest validation WRMSE is used for all test and transfer evaluations. The corresponding training and validation trajectories are shown in Figure 2; both losses decrease steadily over the 200-epoch schedule without late-stage validation divergence.

4. Experiments and Results

4.1. Datasets and Preprocessing

The primary scoring corpus is a 4,200-screen subset of MUD [14] obtained after the published filtering procedure and removal of screens with fewer than five detectable components. Table 1 reports the seven functional categories and average graph statistics. As specified in Section 3.1, the separate 6,000-screen detector-development pool and all of its application identifiers are completely disjoint from all 4,200 scoring screens and their application identifiers.

Cross-dataset evaluation uses 3,600 CLAY screens after preprocessing [15], and the Rico archive is retained as a legacy ablation reference [22]. All screens from one application are assigned to a single partition. MUD contains 2,940 training, 630 validation, and

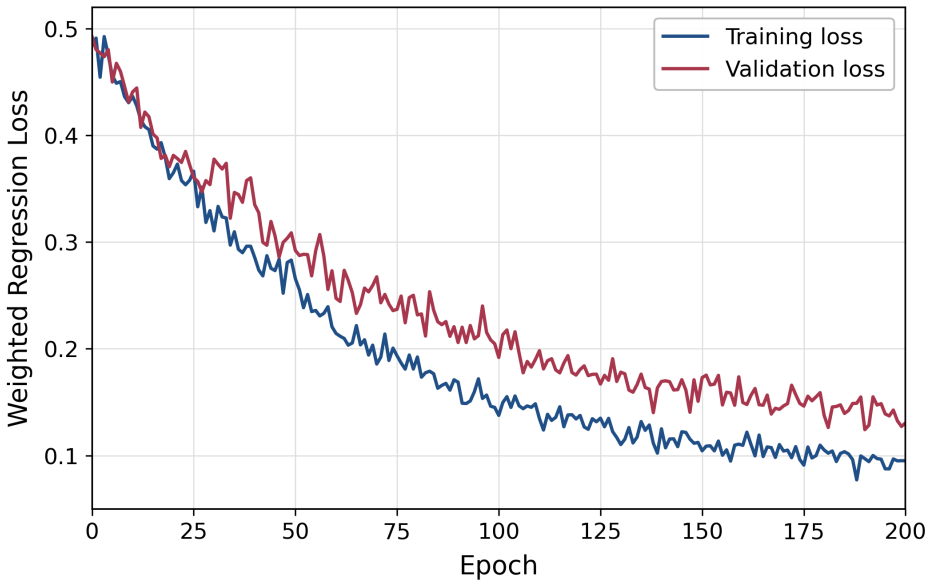


Fig. 2. Training and validation loss over 200 epochs on MUD

Table 1. Distribution of the MUD scoring corpus

Category	Samples	Avg. Nodes	Avg. Align.
Commerce	756	19.2	5.4
Social	648	22.1	6.3
Finance	504	16.1	4.9
Travel	612	20.4	5.8
Education	540	17.5	4.6
Health	516	17.8	5.1
Entertainment	624	21.3	6.0
Total/Avg.	4,200	19.4	5.5

630 test screens; CLAY contains 2,520, 540, and 540 screens, respectively. This app-level protocol prevents near-duplicate screens from the same source application from appearing in both training and evaluation sets.

Seven annotators with at least three years of professional UI/UX experience score each screen on a five-point Likert scale. The target is the trimmed mean after removal of the highest and lowest ratings. Ordinal Krippendorff alpha is 0.76 for MUD and 0.73 for CLAY.

4.2. Metrics, Baselines, and Implementation

Three evaluation metrics are used. Weighted root mean square error measures rating prediction while accounting for inter-rater variance. Spearman rho measures rank consistency. The Gestalt Structure Integrity Ratio measures the proportion of high-attention graph relations that agree with independently derived view-hierarchy relations:

$$\mathrm{WRMSE} = \sqrt{\frac{\sum_i w_i \left(y_i - \widehat{y}_i\right)^2}{\sum_i w_i}} \tag{11}$$

$$\rho = 1 - \frac{6 \sum_i d_i^2}{n\left(n^2 - 1\right)} \tag{12}$$

$$\mathrm{GSIR} = \frac{|E_{\mathrm{attn}}^{\mathrm{top}} \cap E_{\mathrm{struct}}|}{|E_{\mathrm{attn}}^{\mathrm{top}}|} \tag{13}$$

In Equation (11), T is the test set, y_i and $\widehat{y}_i$ are the target and predicted scores, and w_i is defined by Equation (8). In Equation (12), d_i is the difference between the average rank of y_i and the average rank of $\widehat{y}_i$, and n is the number of test screens; the implementation uses average ranks when ties occur. In Equation (13), $E_{\mathrm{attn}}^{\mathrm{top}}$ contains the top 20% of directed graph edges on each screen after attention is averaged across the four heads and two graph layers, E_{struct} is the independently derived view-hierarchy edge set connecting component pairs that share a declared parent container or a parent-child relation, and vertical bars denote set cardinality.

The model is implemented in PyTorch 2.1 and PyTorch Geometric 2.5 with the Ultralytics YOLOv10-S implementation. The PAA-GAT visual stream and the ResNet-50, VGG-16, and EfficientNet-B3 baselines use ImageNet-1K-pretrained weights; YOLOv10-S uses COCO-pretrained weights; MUSIQ uses its public pretrained checkpoint; UIClip uses the public CLIP ViT-B/32 weights [35]; and GCN, GraphSAGE, GAT, and GIN use Xavier initialization [16,23,24,25]. Adam is used with batch size 32 and cosine annealing over 200 epochs. All experiments run on a single NVIDIA RTX 4090.

All scoring models receive 224 by 224 inputs and the same app-level partitions. To preserve layout geometry, flips, rotations, random crops, and perspective transforms are disabled. Training-only photometric augmentation consists of brightness, contrast, and saturation jitter with amplitude 0.10 and probability 0.50, Gaussian blur with probability 0.10, and JPEG-quality perturbation between 85 and 100 with probability 0.20; validation and test images are resized deterministically without augmentation. Detector training uses mosaic with probability 0.50, scale variation of plus or minus 20%, translation of plus or minus 5%, and no horizontal or vertical flipping.

Table 2. Comparison with representative baselines on MUD using an app-level split

Method	WRMSE ↓	ρ ↑	GSIR ↑
ResNet-50 [34]	0.478	0.772	-
VGG-16 [40]	0.503	0.754	-
EfficientNet-B3 [41]	0.452	0.791	-
GCN [16]	0.441	0.805	0.534
GAT [23]	0.431	0.808	0.512
MUSIQ [12]	0.446	0.798	-
UIClip [35]	0.438	0.807	-
PAA-GAT	0.387	0.856	0.671

Learning rate, weight decay, and dropout are selected independently on the validation set from $\{10^{-4}, 3 \times 10^{-4}, 10^{-3}\}$, $\{10^{-5}, 10^{-4}, 10^{-3}\}$, and $\{0.1, 0.3, 0.5\}$, respectively. The selected triples (learning rate, weight decay, dropout) are 3×10^{-4}/10^{-4}/0.3 for the convolutional baselines, except that VGG-16 uses dropout 0.5; 10^{-3}/10^{-4}/0.3 for the graph baselines; 10^{-4}/10^{-4}/0.3 for MUSIQ and UIClip; and 10^{-3}/10^{-4}/0.3 for PAA-GAT. Python, NumPy, PyTorch, and CUDA use random seed 42. Each reported model is the single test evaluation of the checkpoint with the lowest validation WRMSE; the test set is not used for tuning.

Statistical testing addresses the principal baseline comparison and the repeated diagnostic analyses. For PAA-GAT versus EfficientNet-B3, vanilla GAT, and UIClip on the shared 630-screen MUD test set, differences in Spearman correlations are evaluated with Williams' test for two dependent correlations that share the ground-truth rank variable [36,37]. Because the aggregate table does not report the model-to-model rank correlation required by the test, a conservative bound is used: for each comparison, the p-value and standard error are maximized over the complete inter-model-correlation interval that keeps the three-variable correlation matrix positive semidefinite. Two-sided p-values are corrected across the three comparisons by Holm's procedure, following established guidance for multiple classifier comparisons [38,39]; $\Delta\rho$ and conservative 95% confidence intervals are reported as effect sizes. Relative WRMSE reductions are reported descriptively. Two-sided Pearson tests evaluate the six detector checkpoints, and two-sided Spearman tests with 95% confidence intervals evaluate the expert rankings.

4.3. Quantitative Comparison

Table 2 compares PAA-GAT with convolutional, graph, and transformer baselines. VGG-16 is the deep convolutional architecture of Simonyan and Zisserman [40]. ResNet-50 and EfficientNet-B3 represent residual and compound-scaled convolutional encoders [34,41]. The graph baselines use GCN and standard GAT operators [16,23]. MUSIQ is a multi-scale image-quality transformer [12], and UIClip is the CLIP-based UI regression baseline defined in Section 4.2 [35].

PAA-GAT achieves the lowest WRMSE of 0.387 and the highest Spearman rho of 0.856. Relative to EfficientNet-B3, the strongest convolutional baseline, error is reduced by 14.3%. Relative to vanilla GAT, rank correlation improves by 5.9%. The GSIR of

Table 3. Conservative significance analysis for PAA-GAT against the principal baselines on the shared 630-screen MUD test set. Confidence intervals and p-values are worst-case bounds over all feasible inter-model rank correlations

Baseline	WRMSE reduction	$\Delta\rho$	Conservative 95% CI for $\Delta\rho$	Holm-adjusted p upper bound
EfficientNet-B3	14.3%	0.065	[0.035, 0.095]	0.00008
GAT	10.2%	0.048	[0.019, 0.077]	0.0018
UIClip	11.6%	0.049	[0.020, 0.078]	0.0018

0.671 is 31.1% higher than the vanilla-GAT value of 0.512, indicating that explicit priors align attention with meaningful layout relations. Table 3 provides conservative dependent-correlation tests for the three principal baselines. Every conservative confidence interval excludes zero, and every Holm-adjusted p-value upper bound is at most 0.0018. Figure 3 summarizes the point-estimate comparison.

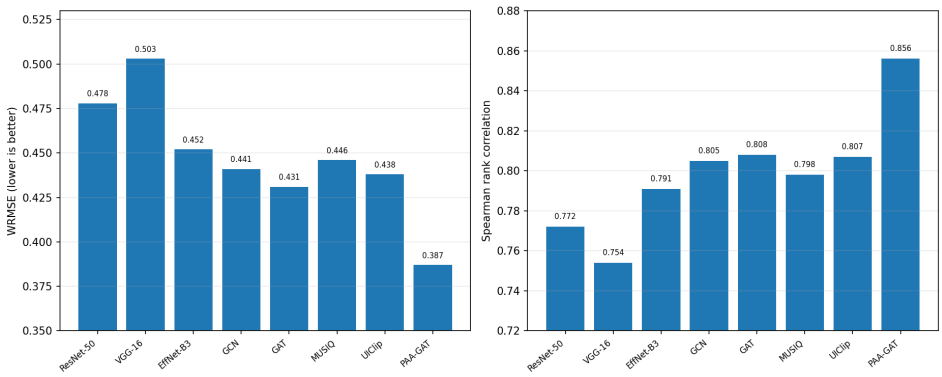


Fig. 3. Baseline comparison for WRMSE (left) and Spearman rank correlation (right)

4.4. Ablation Study

Table 4 removes each structural prior separately and then disables the complete prior set, the visual stream, or the topology stream. This design isolates the contribution of prior knowledge from the contribution of dual-stream fusion.

Removing alignment produces the largest single-prior degradation, increasing WRMSE by 0.034. Proximity, white space, and touch-target removal increase WRMSE by 0.024, 0.018, and 0.011, respectively. The touch-target effect is modest overall but reaches 0.018 on commerce screens, where dense product-card layouts are especially susceptible to target-size and separation violations. Figure 4 presents the normalized ablation profile.

Table 4. Ablation study on MUD. P: proximity; A: alignment; W: white space; T: touch target

Variant	P	A	W	T	WRMSE ↓	ρ ↑	GSIR ↑
PAA-GAT (full)	Yes	Yes	Yes	Yes	0.387	0.856	0.671
Without proximity	No	Yes	Yes	Yes	0.411	0.832	0.624
Without alignment	Yes	No	Yes	Yes	0.421	0.821	0.608
Without white space	Yes	Yes	No	Yes	0.405	0.838	0.641
Without touch target	Yes	Yes	Yes	No	0.398	0.845	0.658
Without all priors	No	No	No	No	0.431	0.808	0.512
Without visual stream	Yes	Yes	Yes	Yes	0.428	0.801	0.663
Without topology stream	-	-	-	-	0.462	0.779	-

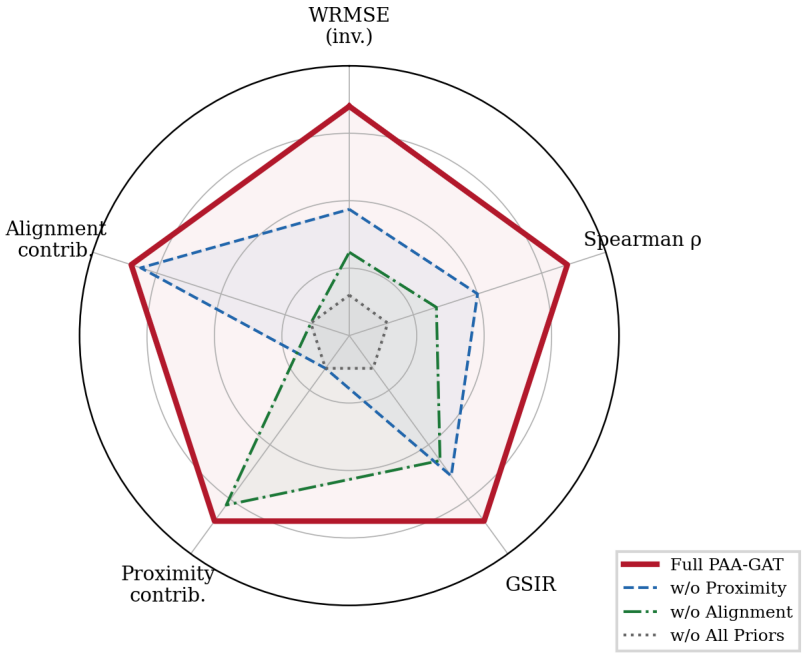


Fig. 4. Normalized comparison of full and ablated variants across five quality axes

Table 5. Effect of YOLOv10 detection accuracy on graph integrity and score precision

mAP@0.5	GSIR ↑	WRMSE ↓	Δ GSIR	Δ WRMSE
0.85	0.612	0.423	—	—
0.87	0.628	0.415	+0.016	−0.008
0.89	0.641	0.408	+0.029	−0.015
0.91	0.653	0.401	+0.041	−0.022
0.93	0.664	0.394	+0.052	−0.029
0.95 (final)	0.671	0.387	+0.059	−0.036

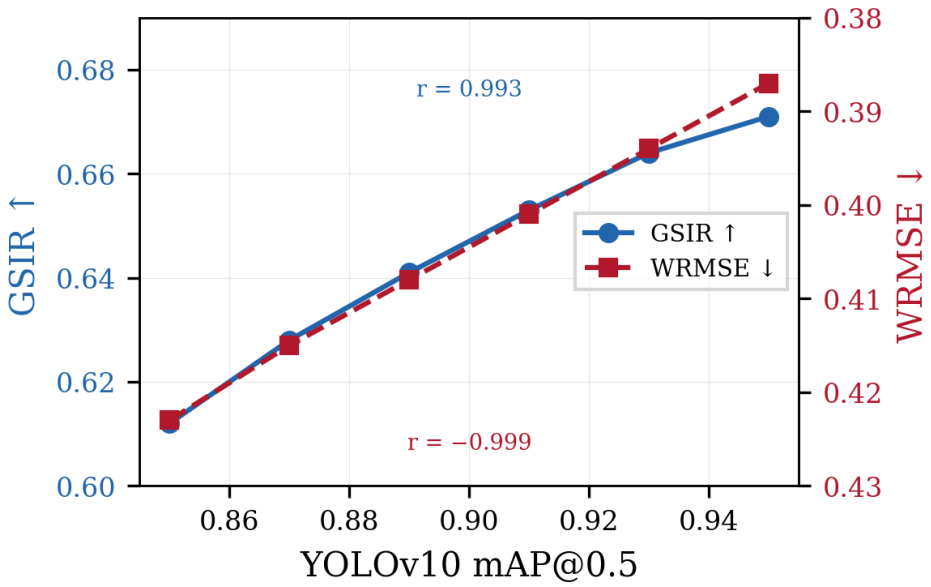


Fig. 5. Downstream sensitivity to YOLOv10 detection accuracy

4.5. YOLOv10 Parser-Sensitivity Analysis

Because component geometry enters the graph directly, detector accuracy should influence structural integrity and score precision. Table 5 evaluates six detector checkpoints whose mAP@0.5 values range from 0.85 to 0.95. The 0.92 detector-only warm-up result reported in Section 3.1 is an intermediate checkpoint; the 0.95 row is the final jointly optimized checkpoint used for the principal results.

Detector mAP is strongly associated with both downstream metrics: Pearson r is 0.993 for GSIR, with a two-sided p-value of 7.33×10^{-5}, and -0.999 for WRMSE, with a two-sided p-value of 1.50×10^{-6}. Each 0.01 increase in mAP@0.5 corresponds on average to a GSIR gain of 0.0059 and a WRMSE reduction of 0.0036. The monotonic trend in Figure 5 indicates that the scorer continues to benefit from improved component localization over the tested range.

Table 6. Agreement between diagnostic components and expert-consensus rankings (50 screens, 10 experts)

Dimension	Spearman ρ	p-value	95% CI
Proximity (z_{prox})	0.79	< 0.001	[0.65, 0.88]
Alignment (z_{align})	0.84	< 0.001	[0.72, 0.91]
White-space (z_{ws})	0.76	< 0.001	[0.61, 0.86]
Touch-target (z_{acc})	0.81	< 0.001	[0.68, 0.89]

4.6. Expert Diagnostic Blind Test

A double-blind study involves ten senior UI/UX designers with a mean of 7.3 years of experience and a range of 5-12 years. Fifty MUD test screens are sampled. Each expert independently ranks the screens on proximity, alignment, white space, and touch-target quality using a forced-ranking procedure and remains blind to model outputs.

Table 6 reports Spearman rank agreement between each component of z_{diag} and the corresponding expert-consensus ranking.

All four correlations exceed 0.75 and have p-values below 0.001. Alignment has the highest agreement at 0.84, consistent with the ablation result that alignment is the most discriminative prior. Touch-target agreement reaches 0.81, indicating that the WCAG-grounded diagnostic captures accessibility concerns independently identified by practitioners.

4.7. Hyperparameter Sensitivity

Figure 6 varies the global prior-strength coefficient λ from 0.0 to 1.0 in increments of 0.1. Performance improves from a WRMSE of 0.431 and a Spearman rho of 0.808 at λ equal to zero to a WRMSE of 0.387 and a Spearman rho of 0.856 at λ equal to 0.4. Larger values progressively overemphasize hand-specified structure relative to learned compatibility. The smooth unimodal curve identifies 0.4 as the validation optimum and also distinguishes λ from the relative β coefficients in Equation (6).

4.8. Cross-Dataset Generalization

Table 7 evaluates in-domain MUD, in-domain CLAY, MUD-to-CLAY zero-shot transfer, and the legacy Rico ablation setting. The corpora differ in visual style, locale, and curation procedure, making the transfer test sensitive to overreliance on texture or source-specific rendering.

As summarized in Figure 7, GSIR remains near or above 0.60 in all settings, while the zero-shot WRMSE increases to 0.439. The result suggests that normalized relational priors transfer more robustly than source-specific appearance features, although the performance gap confirms that corpus shift remains consequential.

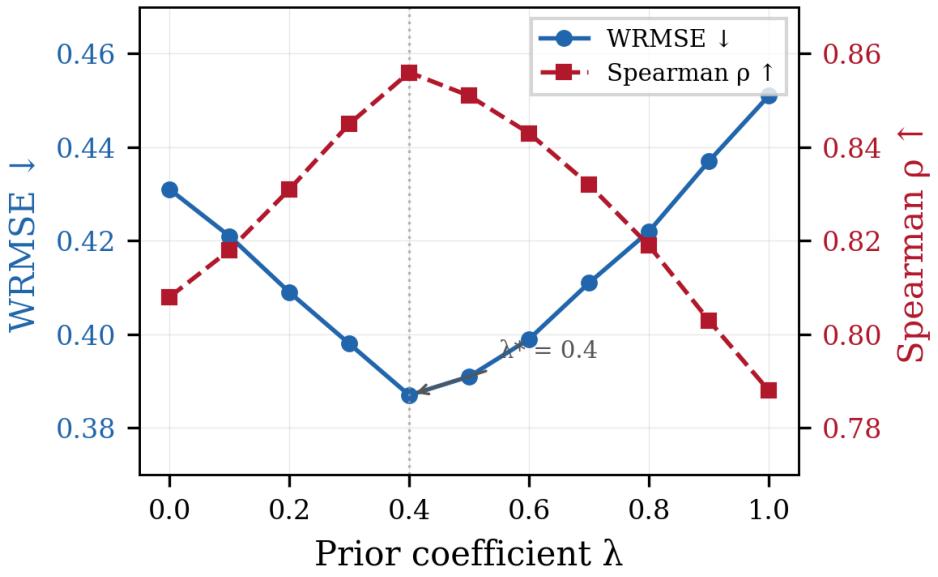


Fig. 6. Sensitivity to the global prior-strength coefficient lambda

Table 7. Cross-dataset generalization

Setting	**WRMSE ↓**	**ρ ↑**	**GSIR ↑**
MUD (app-level)	0.387	0.856	0.671
CLAY (app-level)	0.408	0.837	0.642
MUD → CLAY (zero-shot)	0.439	0.812	0.601
Rico (ablation)	0.431	0.812	0.598

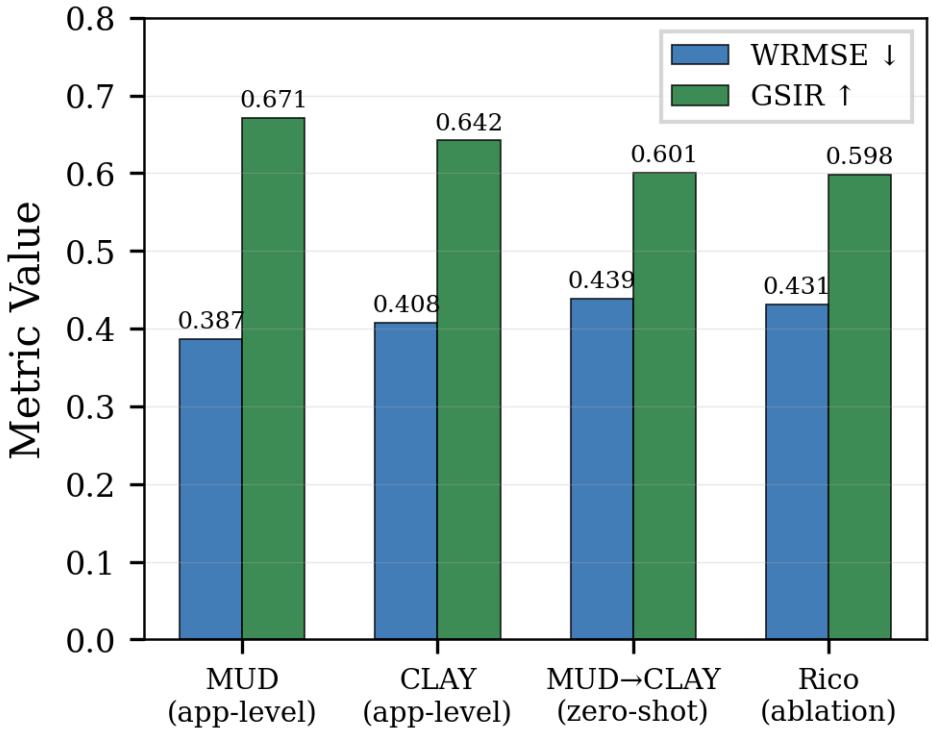


Fig. 7. WRMSE and GSIR across in-domain, transfer, and legacy evaluation settings

Table 8. GNN backbone comparison on MUD

Backbone	WRMSE ↓	ρ ↑	GSIR ↑
GCN [16]	0.441	0.805	0.534
GraphSAGE [24]	0.436	0.812	0.543
GAT [23]	0.431	0.808	0.512
GIN [25]	0.445	0.799	0.526
PAA-GAT	0.387	0.856	0.671

4.9. GNN Backbone Comparison

Table 8 replaces the prior-guided operator with GCN, GraphSAGE, vanilla GAT, and GIN while holding the visual stream and regression head fixed [16,23,24,25]. This comparison separates the proposed attention rule from the general benefit of graph message passing.

Relative to vanilla GAT, the proposed operator reduces WRMSE by 0.044, increases Spearman rho by 0.048, and increases GSIR by 0.159. Figure 8 shows that the advantage is not reproduced by changing only the generic graph backbone.

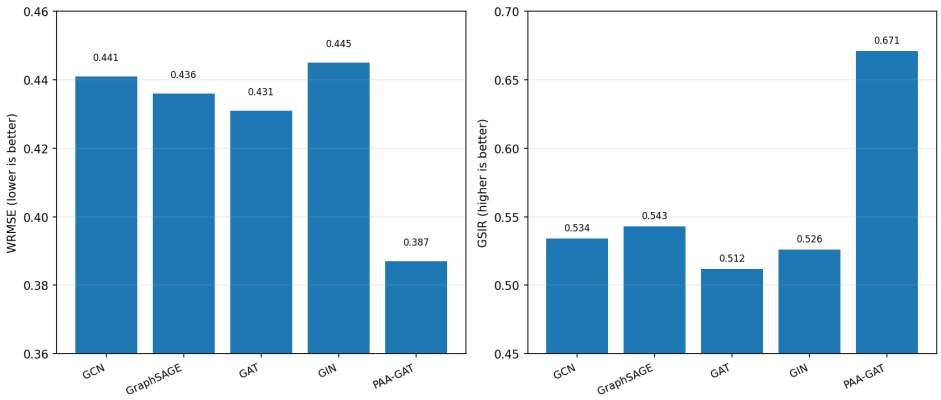


Fig. 8. GNN backbone comparison for WRMSE (left) and GSIR (right)

4.10. Visual Interpretability and Failure Analysis

Figure 9 visualizes prior-modulated attention on a commerce screen. In the highest-scoring quartile of the MUD test set, the mean diagnostic composition is [0.28, 0.39, 0.19, 0.14] for proximity, alignment, white space, and touch target. The lowest-scoring quartile produces [0.35, 0.27, 0.24, 0.14]. High-quality layouts therefore derive more of their structural evidence from consistent anchors, whereas low-quality layouts rely more heavily on local grouping without comparable alignment.

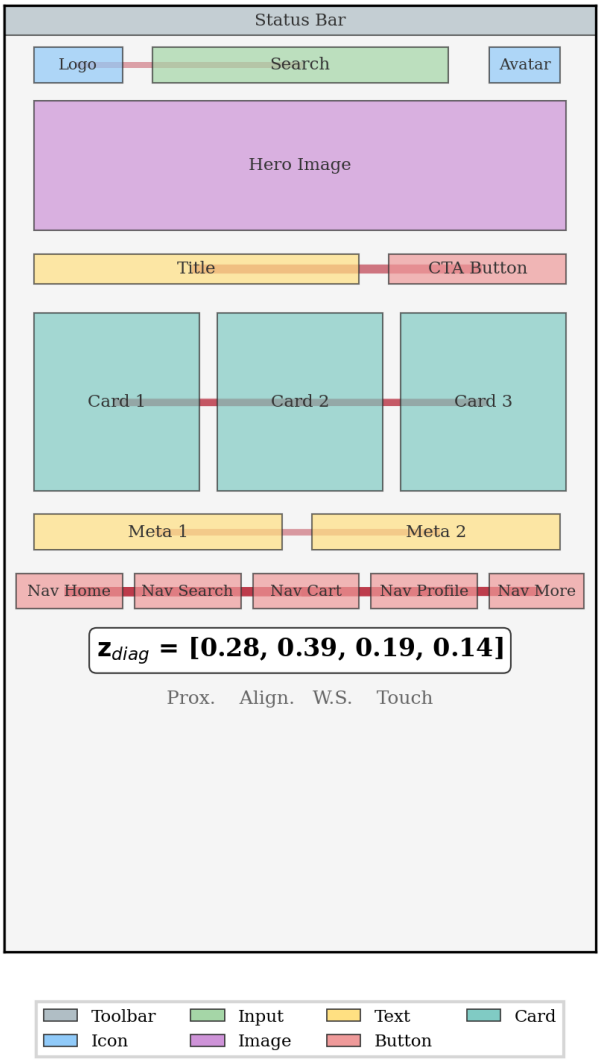


Fig. 9. Prior-modulated attention on a representative commerce screen; edge thickness denotes attention weight and the diagnostic decomposition is shown below the layout

Table 9. End-to-end computational cost

Method	GFLOPs	Params (M)	Latency (ms)	WRMSE ↓
EfficientNet-B3 [41]	1.8	12.2	14	0.452
ResNet-50 [34]	4.1	23.5	22	0.478
VGG-16 [40]	15.5	138.4	31	0.503
GAT [23]	4.3	25.8	28	0.431
MUSIQ [12]	12.4	86.3	45	0.446
UIClip [35]	17.6	149.0	52	0.438
PAA-GAT	6.8	27.1	35	0.387

Two recurrent failure modes are observed. Detector fragmentation can split one navigation bar into many adjacent nodes, causing the proximity prior to overvalue spurious local relations. Deliberately non-standard editorial or promotional compositions can also be penalized because their quality depends on controlled rule breaking that is not represented by the four current priors.

4.11. Computational Cost Analysis

Table 9 compares floating-point operations, parameter count, and latency on the RTX 4090 reference platform. Latency is measured with batch size one after 200 warm-up iterations and averaged over 1,000 timed screens. It includes image preprocessing, YOLOv10 detection, graph construction, both feature streams, fusion, and score regression, but excludes disk input/output.

PAA-GAT requires 6.8 GFLOPs, 27.1 million parameters, and 35 milliseconds per screen end to end. It produces lower error than UIClip with less than 40% of its computation and approximately 18% of its parameter count. The YOLOv10-S detector accounts for 2.4 GFLOPs, which is 14% below the 2.8-GFLOP U^2-Net parser while retaining differentiable box and logit outputs.

5. Conclusions

This study presents an end-to-end mobile-interface scorer that connects pixel-level appearance, component detection, and relational layout reasoning. Replacing an external U^2-Net parser with a jointly optimized YOLOv10 detector allows score gradients to influence component geometry, while four differentiable priors make the graph-attention mechanism diagnostically interpretable.

On MUD, PAA-GAT reaches a WRMSE of 0.387, a Spearman rank correlation of 0.856, and a GSIR of 0.671 with the final jointly optimized detector at mAP@0.5 of 0.95. Conservative Williams tests confirm higher rank correlation than EfficientNet-B3, vanilla GAT, and UIClip after Holm correction, with all adjusted p-value upper bounds at or below 0.0018. Ablation, backbone substitution, cross-dataset evaluation, and parameter sensitivity consistently support the contribution of prior-guided attention. Detector accuracy is significantly associated with both GSIR and WRMSE, and expert agreement with the four diagnostic dimensions ranges from 0.76 to 0.84.

Theoretical limitations arise from the prior set and the static-screen formulation. Proximity, alignment, white space, and touch-target accessibility do not exhaust visual hierarchy; reading order, color contrast, semantic salience, and temporal interaction are omitted. The priors may also penalize high-quality layouts that intentionally depart from conventional structure, and the current fixed-K graph does not model uncertainty over component existence explicitly.

Practical limitations concern data provenance and deployment. App-level splitting reduces effective training diversity, detector errors propagate into graph construction, and the 24-pixel accessibility prior cannot substitute for device-aware testing of dynamic hit regions. The diagnostic vector should be used as decision support rather than as an autonomous design verdict or a legal accessibility-certification tool.

Future work will examine heterogeneous type-specific attention, uncertainty-aware graph construction, multi-page interaction flows, additional differentiable accessibility criteria, and larger expert-review corpora. Metaheuristic search could be used for deployment-aware architecture tuning, while the diagnostic vector could serve as a constrained reward for layout generation. In organizational and public-sector settings, the method may support transparent pre-screening standards for design procurement and quality assurance, provided that thresholds, model versions, and human-review procedures are documented and periodically audited.

References

1. Zhu, T., Yang, Y.: Research on Mobile Learning Platform Interface Design Based on College Students' Visual Attention Characteristics. PLoS ONE 18, e0283778 (2023). https://doi.org/10.1371/journal.pone.0283778
2. Odushegun, L.: Aesthetic Semantics: Affect Rating of Atomic Visual Web Aesthetics for Use in Affective User Experience Design. Int. J. Hum.-Comput. Stud. 171, 102978 (2023). https://doi.org/10.1016/j.ijhcs.2022.102978
3. Kyelem, Y., Zo, M., Kabore, K.K., Ouedraogo, F.: Automatic Graphical User Interface Aesthetic Evaluation Tool Using the UIED Segmentation Algorithm. In: Proceedings of the 3rd International Conference on Human-Machine Interaction, pp. 1-7. ACM, New York (2023). https://doi.org/10.1145/3604383.3604387
4. Yang, Y., Zhuo, Y., Liu, J., Meng, W., Wu, Z.: A Hybrid Quantitative Method for Evaluating HMI Layout Design in Service Robots. Symmetry 17, 2102 (2025). https://doi.org/10.3390/sym17122102
5. Demir, G., Cekmis, A., Yesilkaynak, V.B., Unal, G.: Detecting Visual Design Principles in Art and Architecture through Deep Convolutional Neural Networks. Autom. Constr. 130, 103826 (2021). https://doi.org/10.1016/j.autcon.2021.103826
6. de Souza Lima, A.L., Martins, A.I., Rosa, A.F., Queiros, A., Rocha, N.P., et al.: A Deep Learning Model for the Assessment of the Visual Aesthetics of Mobile User Interfaces. J. Braz. Comput. Soc. 30, 102-115 (2024). https://doi.org/10.5753/JBCS.2024.3255
7. Lima, A.L.S., Martins, O.P.H.R., Gresse von Wangenheim, C., von Wangenheim, A., Borgatto, A.F., Hauck, J.C.R.: Automated Assessment of Visual Aesthetics of Android User Interfaces with Deep Learning. In: Proceedings of IHC 2022, Article 8, pp. 1-11. ACM, New York (2022). https://doi.org/10.1145/3554364.3559113
8. Xing, B., Cao, H., Shi, L., Si, H., Zhao, L.: AI-Driven User Aesthetics Preference Prediction for UI Layouts via Deep Convolutional Neural Networks. Cogn. Comput. Syst. 4, 250-264 (2022). https://doi.org/10.1049/ccs2.12055

9. Delitzas, A., Chatzidimitriou, K.C., Symeonidis, A.L.: Calista: A Deep Learning-Based System for Understanding and Evaluating Website Aesthetics. Int. J. Hum.-Comput. Stud. 175, 103019 (2023). https://doi.org/10.1016/j.ijhcs.2023.103019
10. Keselj, A., Milicevic, M., Zubrinic, K., Car, Z.: The Application of Deep Learning for the Evaluation of User Interfaces. Sensors 22, 9336 (2022). https://doi.org/10.3390/s22239336
11. Talebi, H., Milanfar, P.: NIMA: Neural Image Assessment. IEEE Trans. Image Process. 27, 3998-4011 (2018). https://doi.org/10.1109/TIP.2018.2831899
12. Ke, J., Wang, Q., Wang, Y., Milanfar, P., Yang, F.: MUSIQ: Multi-Scale Image Quality Transformer. In: Proceedings of ICCV 2021, pp. 5148-5157. IEEE (2021)
13. Duan, P., Cheng, C.-Y., Li, G., Hartmann, B., Li, Y.: UICrit: Enhancing Automated Design Evaluation with a UI Critique Dataset. In: Proceedings of UIST 2024, Article 46, pp. 1-17. ACM, New York (2024). https://doi.org/10.1145/3654777.3676381
14. Feng, S., Fu, Y., Zhang, Y., et al.: MUD: Towards a Large-Scale and Noise-Filtered UI Dataset for Modern Style UI Modeling. In: Proceedings of CHI 2024, pp. 1-14. ACM, New York (2024). https://doi.org/10.1145/3613904.3642350
15. Li, G., Baechler, G., Tragut, M., Li, Y.: Learning to Denoise Raw Mobile UI Layouts for Improving Dataset Quality. In: Proceedings of CHI 2022, pp. 1-13. ACM, New York (2022)
16. Kipf, T.N., Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks. In: International Conference on Learning Representations (2017)
17. Zhou, H., Yang, Y., Luo, T., Zhang, J., Li, S.: A Unified Deep Sparse Graph Attention Network for Scene Graph Generation. Pattern Recognit. 123, 108367 (2022). https://doi.org/10.1016/j.patcog.2021.108367
18. Liang, W., Jiang, Y., Liu, Z.: GraphVQA: Language-Guided Graph Neural Networks for Graph-Based Visual Question Answering. In: Proceedings of the 3rd Workshop on Multimodal Artificial Intelligence, pp. 79-86. ACL (2021)
19. She, D., Lai, Y.-K., Yi, G., Xu, K.: Hierarchical Layout-Aware Graph Convolutional Network for Unified Aesthetics Assessment. In: Proceedings of CVPR 2021, pp. 8475-8484. IEEE (2021)
20. World Wide Web Consortium: Web Content Accessibility Guidelines (WCAG) 2.2. W3C Recommendation (2023). https://www.w3.org/TR/WCAG22/
21. Ali, A., Xia, Y., Navid, Q., Khan, Z.A., Khan, J.A., Aldakheel, E.A., Khafaga, D.: Mobile-UI-Repair: A Deep Learning-Based UI Smell Detection Technique for Mobile User Interfaces. PeerJ Comput. Sci. 10, e2028, 1-29 (2024). https://doi.org/10.7717/peerj-cs.2028
22. Deka, B., Huang, Z., Franzen, C., Hibschman, J., Afergan, D., Li, Y., Nichols, J., Kumar, R.: Rico: A Mobile App Dataset for Building Data-Driven Design Applications. In: Proceedings of UIST 2017, pp. 845-854. ACM, New York (2017)
23. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph Attention Networks. In: International Conference on Learning Representations (2018)
24. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive Representation Learning on Large Graphs. In: Advances in Neural Information Processing Systems 30, pp. 1024-1034 (2017)
25. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How Powerful Are Graph Neural Networks? In: International Conference on Learning Representations (2019)
26. Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., Yu, P.S.: Heterogeneous Graph Attention Network. In: Proceedings of WWW 2019, pp. 2022-2032. ACM, New York (2019)
27. Duan, P., Hartmann, B., Nguyen, K., Li, Y., Hearst, M., Morris, M.R.: Towards Semantically-Aware UI Design Tools: Design, Implementation, and Evaluation of Semantic Grouping Guidelines. In: ICML 2023 Workshop on Artificial Intelligence and Human-Computer Interaction (2023). https://research.google/pubs/towards-semantically-aware-ui-design-tools-design-implementation-and-evaluation-of-semantic-grouping-guidelines/

28. Li, J., Mao, J., Zeng, J., Lin, Q., Feng, S., Liang, Z.: UIHash: Detecting Similar Android UIs through Grid-Based Visual Appearance Representation. In: Proceedings of the 33rd USENIX Security Symposium, pp. 1-18. USENIX Association (2024)
29. Li, T.F.Y., Schilit, B.: Screen Recognition: Creating Accessibility Metadata for Mobile Applications. In: Proceedings of CHI 2020, pp. 1-15. ACM, New York (2020)
30. Jovanovic, L., Jokic, A., Milovanovic, M., Antonijevic, M., Zivkovic, M., Bacanin, N.: Exploring Lightweight YOLOv8 Architectures for Circuit Board Defect Detection. In: Proceedings of ICEC 2024, pp. 1-6. IEEE (2024). https://doi.org/10.1109/ICEC59683.2024.10837424
31. Tasic, A., Jovanovic, L., Bacanin, N., Zivkovic, M., Simic, V., Popovic, M., Antonijevic, M.: Towards Sustainable Societies: Convolutional Neural Networks Optimized by Modified Crayfish Optimization Algorithm Aided by AdaBoost and XGBoost for Waste Classification Tasks. Appl. Soft Comput. 175, 113086 (2025). https://doi.org/10.1016/j.asoc.2025.113086
32. Qin, X., Zhang, Z., Huang, C., Dehghan, M., Zaiane, O.R., Jagersand, M.: U2-Net: Going Deeper with Nested U-Structure for Salient Object Detection. Pattern Recognit. 106, 107404 (2020). https://doi.org/10.1016/j.patcog.2020.107404
33. Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., Ding, G.: YOLOv10: Real-Time End-to-End Object Detection. In: Advances in Neural Information Processing Systems 37, pp. 107984-108011 (2024). https://doi.org/10.52202/079017-3429
34. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: Proceedings of CVPR 2016, pp. 770-778. IEEE (2016)
35. Radford, A., Kim, J.W., Hallacy, C., et al.: Learning Transferable Visual Models from Natural Language Supervision. In: Proceedings of ICML 2021, pp. 8748-8763. PMLR (2021)
36. Steiger, J.H.: Tests for Comparing Elements of a Correlation Matrix. Psychol. Bull. 87(2), 245-251 (1980). https://doi.org/10.1037/0033-2909.87.2.245
37. Diedenhofen, B., Musch, J.: cocor: A Comprehensive Solution for the Statistical Comparison of Correlations. PLoS ONE 10(4), e0121945 (2015). https://doi.org/10.1371/journal.pone.0121945
38. Derrac, J., Garcia, S., Molina, D., Herrera, F.: A Practical Tutorial on the Use of Nonparametric Statistical Tests as a Methodology for Comparing Evolutionary and Swarm Intelligence Algorithms. Swarm Evol. Comput. 1, 3-18 (2011). https://doi.org/10.1016/j.swevo.2011.02.002
39. Demsar, J.: Statistical Comparisons of Classifiers over Multiple Data Sets. J. Mach. Learn. Res. 7, 1-30 (2006)
40. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. In: International Conference on Learning Representations (2015)
41. Tan, M., Le, Q.V.: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In: Proceedings of ICML 2019, pp. 6105-6114. PMLR (2019)

Long Zeng received his Ph.D. degree from Sejong University, Seoul, South Korea. His research interests include intelligent design.

Zi Wang was born in Chenzhou, Hunan Province. She received the master's degree in software engineering from Central South University, Changsha. Her research interests include graphic design and artificial intelligence.

Wen Zuo holds a Ph.D. degree and is currently an Associate Professor in the School of Fine Arts and Artistic Design, Hunan University of Science and Engineering, China. Her research interests include intelligent design and visual communication design.

Jun Hoong Chan received his B.Eng. degree in Manufacturing Engineering from the University of Malaya, Malaysia, in 2019, and his M.S. degree in Computer Science and Engineering from Sejong University, South Korea, in 2021. He received his Ph.D. degree in Computer Application Technology from Peking University, China, in 2026. His research interests include computer vision, photometric stereo, reinforcement learning, and 3D vision reconstruction.

Received: July 6, 2026.; Accepted: August 18, 2026.