Relation-enhanced DETR for Component Detection in Graphic Design Reverse Engineering

Xixuan Hao[1*], **Danqing Huang**[2 †], **Jieru Lin**[3*] and **Chin-Yew Lin**[2]
[1]The University of Hong Kong
[2]Microsoft Research
[3]Harbin Institute of Technology
hxxjxw@connect.hku.hk, {dahua, cyl}@microsoft.com, hitjierulin@gmail.com

Abstract

It is a common practice for designers to create digital prototypes from a mock-up/screenshot. Reverse engineering graphic design by detecting its components (e.g., text, icon, button) helps expedite this process. This paper first conducts statistical analysis to emphasize the importance of relations in graphic layouts, which further motivates us to incorporate relation modeling into component detection. Built on the current state-of-the-art DETR (DEtection TRansformer), we introduce a learnable relation matrix to model class correlations. Specifically, the matrix will be added to the DETR decoder to update the query-to-query self-attention. Experiment results on three public datasets show that our approach achieves better performance than several strong baselines. We further visualize the learned relation matrix and observe some reasonable patterns. Moreover, we show an application of component detection where we leverage the detection outputs as augmented training data for layout generation, which achieves promising results.

1 Introduction

Reverse engineering graphic design aims to detect the logical components from a pixel-based design and parse its layout structure. It plays an important role in design understanding and can potentially enable many downstream applications (e.g., digital prototyping from a mock-up artifact or screenshot). In Figure 1, we show some design examples annotated with component classes and bounding boxes, including mobile user interface (UI), poster and slide. As we can see, the layouts are diverse and complex with many free-form and overlapped components, thus bringing more challenges for component detection.

One major characteristic across different types of design is that objects usually follow certain relation patterns. For example in mobile UI, `Icon` is often created inside `ToolBar` as a navigation widget on top of the interface. As another example, `freeform` objects in a slide are usually clustered in a small region to form a meaningful shape. To better quantify this characteristic, we conduct a statistical analysis of relations in graphic layouts with some interesting findings.

Figure 1: Graphic design examples including (a) mobile UI from RICO; (b) posters from Crello; (c) slides from InfoPPT.

Recently MagicLayout [Manandhar *et al.*, 2021] is an initial attempt to incorporate co-occurrence relations into UI component detection. It statistically calculates a co-occurrence matrix between object classes in the corpus and incorporates it as a fixed weight into Faster-RCNN [Ren *et al.*, 2015] to enhance the proposal features. Although MagicLayout has shown the effectiveness of the co-occurrence relation, its matrix is only calculated once as a prior and fixed during the training process, which might not be flexible for model learning. Moreover, the relation is constrained to co-occurrence while other types of relations (e.g., spatial overlap) [Jiang *et al.*, 2018] could also be useful in capturing complex interactions between objects in graphic layouts.

In this paper, we explore a more flexible solution of relation modeling for component detection. We build our method on the current state-of-the-art detection framework DETR [Carion *et al.*, 2020]. DETR is an end-to-end pipeline which uses a sparse set of learnable object queries as input to a Trans-

*Work done during internship at Microsoft Research Asia.
†Corresponding Author.

former encoder-decoder. In the decoder, queries will interact with each other (self-attention) and attend to relevant regions of the image (cross-attention). Specifically, we propose a learnable relation matrix for capturing class-to-class correlations. For each query pair, we retrieve the corresponding weight from the relation matrix using their predicted class indexes from the previous decoder layer. This relation weight will then be added to the self-attention weight to update query-to-query interactions.

We conduct experiments on three publicly-available graphic design datasets, including RICO [Deka *et al.*, 2017] (mobile UI), Crello [Yamaguchi, 2021] (posters) and InfoPPT [Shi *et al.*, 2022] (slides). Experiment results show that such simple relation modeling works surprisingly well and achieves better performance than several strong baselines. Through matrix visualization and case studies, we can observe that the relation matrix has captured certain reasonable correlations between class pairs. Furthermore, we show a potential application of layout generation using the component detection outputs as augmented training data and achieve promising results.

To summarize, the main contributions of this paper are:

- We conduct an in-depth analysis of relations in graphic design layouts to emphasize their importance.
- We propose a simple but effective relation-enhanced self-attention based on the strong object detector DETR, and achieve currently the best performance on three datasets.
- We explore an application of leveraging component detection outputs as augmented training data in layout generation, which shows promising results.

2 Related Work

2.1 Component Detection in Graphic Design

Traditional approaches aggregate edge or contour features with heuristic rules to recover the interface structure [Yeh *et al.*, 2009; Nguyen and Csallner, 2015; Dixon and Fogarty, 2010]. For example, REMAUI [Nguyen and Csallner, 2015] combines the OCR outputs of text boxes and the detected edges of elements to infer the final components on a design using simple merging algorithms. In recent years, as neural detection networks such as Faster RCNN [Ren *et al.*, 2015] and Mask RCNN [He *et al.*, 2017] have shown great performance across different domains, many works directly adopt these detection models for GUI component detection [Moran *et al.*, 2018]. Some methods further incorporate specific features in graphic design. E3Nets [Ma *et al.*, 2021] embeds edge features along with the RGB channels to train a segmentation model, as edges provide useful information of the object skeleton. The most closely related research to this paper is MagicLayout [Manandhar *et al.*, 2021], which exploits common spatial relationships of components in UI layouts. It pre-calculates a co-occurrence matrix between object classes in the corpus and uses this fixed matrix as prior to update the proposal features in Faster RCNN. In this paper, we explore a more flexible solution of relation modeling and will show that our proposed learnable matrix can capture reasonable patterns.

Meanwhile, document layout analysis is a highly relevant task of identifying regions of interest in the scanned image of a document. It mainly focuses on textual documents such as forms [Harley *et al.*, 2015] and scientific articles [Zhong *et al.*, 2019], where text blocks are dominant and layouts are mostly simple rectangle grids. Detailed related works can be referred to [Binmakhashen and Mahmoud, 2019]. Being compared, graphic design layouts have more complex variations with free-form and overlapping objects, which makes its component detection more challenging.

2.2 Object Detection

Previous works on graphic design component detection mainly view it as a detection task. At the present stage, most object detection methods [Girshick, 2015; Ren *et al.*, 2015; He *et al.*, 2015; Lin *et al.*, 2017; Wang *et al.*, 2022a] are known as anchor-based, which rely on numerous hand-crafted target assignments and non-maximum suppression post-processing, and thus are not fully end-to-end trainable and hard for parameter tuning.

DETR and its variants Recently DETR [Carion *et al.*, 2020] has been proposed as a new paradigm for object detection which achieves promising results. It employs a Transformer architecture and replaces hand-crafted components with a set-based global loss. As DETR has suffered from slow and unstable convergence during training, many following works [Zhu *et al.*, 2021; Liu *et al.*, 2022; Yao *et al.*, 2021; Meng *et al.*, 2021; Wang *et al.*, 2022b; Zhang *et al.*, 2022; Gao *et al.*, 2021] have been proposed to eliminate the issues. For example, Deformable DETR [Zhu *et al.*, 2021] designs multi-scale deformable attention to only focus on a small set of relevant regions. DAB-DETR [Liu *et al.*, 2022] explicitly assigns 4D box coordinates to queries as positional prior which obtain faster convergence.

Relation Modeling There are many pioneering works that have already verified the effectiveness of modeling relations in object detection [Jiang *et al.*, 2018; Chen *et al.*, 2019; Xu *et al.*, 2019; Zhao *et al.*, 2021; Bi *et al.*, 2022; Manandhar *et al.*, 2021]. Relations used in previous works can be mainly categorized into three types: (1) *co-occurrence* between object classes. For example, [Chen *et al.*, 2019] uses conditional probabilities to represent co-occurrence; MagicLayout aggregates co-occurrence frequencies of objects in different regions; (2) *spatial relation* computes relative positions between objects. [Zhao *et al.*, 2021] use self-attention to build position-wise spatial relation. SRRV [Bi *et al.*, 2022] simply uses the distance between region proposals and achieves great performance improvement; (3) *learnable relation* introduces more model capacities. For example, SGRN [Xu *et al.*, 2019] defines a sparse graph where the edge weights are tuned during training. In this paper, we follow the direction of learnable relation modeling which is more flexible with less human priors.

3 Methodology

In this section, we first conduct an analysis to emphasize the importance of relations in graphic design layouts. Then

we revisit the DETR architecture and propose our relation-enhanced attention mechanism.

3.1 Relations in Graphic Design Layouts

Relations between objects play quite an important role in graphic design layouts. For example in Figure 2, the combination of `Toolbar` and `Icon` (highlighted in red bold edge) exists in all three interfaces, as they compose a navigation widget which has a high frequency of co-occurring in the upper region.

To better quantify this characteristic, we measure the correlation degree of class pairs at corpus-level using Pointwise Mutual Information (PMI):

$$\mathrm{PMI}(c_x, c_y) = \log \frac{P(c_x, c_y)}{P(c_x)P(c_y)} \tag{1}$$

where $P(c_x, c_y)$ denotes the probability of class c_x and c_y co-occurring in a layout. $\mathrm{PMI} = 0$ indicates statistically independent between two classes, while a larger PMI value indicates a higher positive correlation.

We use the mobile UI dataset RICO for analysis and show its PMI distribution in Figure 3(a) by enumerating all possible class pairs. As for comparison, we also compute the PMI on the natural image dataset COCO [Lin *et al.*, 2014] (Figure 3(b)). In our measurement, there are approximate **93%** class pairs in RICO with PMI values larger than 0, and the distribution is negatively-skewed indicating the majority of class pairs have high positive correlations. While being compared, there are only **62%** frequently co-occurred pairs in COCO and the PMI values are evenly distributed. Based on this statistic, we can observe different characteristics between natural image and graphic design layouts, where the latter contains more highly correlated class pairs.

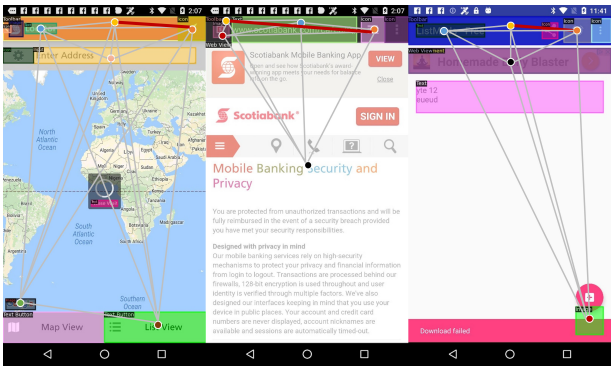
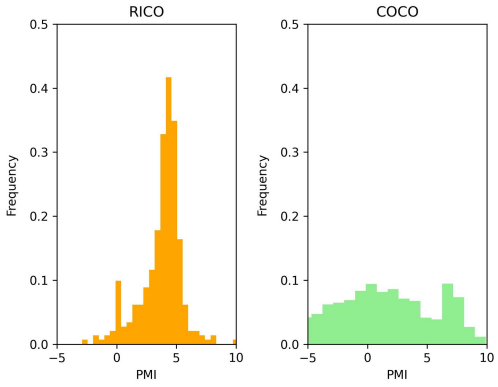


Figure 2: Examples of mobile UIs. `Toolbar` and `Icon` are combined as a navigation widget and have a high co-occurrence probability (highlighted in red bold edge).

3.2 Revisit DETR

Since our method is built on the current state-of-the-art detection model DETR, here we briefly introduce its architecture as shown in Figure 4. It contains a CNN backbone, a Transformer encoder-decoder, and two prediction heads to predict object classes and bounding boxes. For an input image of $(H, W, 3)$,



(a) (b)

Figure 3: PMI distribution of class pairs in (a) RICO and (b) COCO. The distribution in RICO is negatively-skewed, indicating more positive correlated class pairs in RICO.

- The CNN backbone extracts image features and produces a feature map of $(\frac{H}{32}, \frac{W}{32}, C)$. C is the output channel size.
- The encoder takes in the summation of the flattened feature map and the positional embedding, then outputs the encoded image feature of $(\frac{H}{32}, \frac{W}{32}, C)$.
- The decoder inputs a set of learnable object queries (N, C) and updates the queries with self-attention (query-to-query) and cross-attention (query-to-image).
- For each query, two prediction heads predict the class probabilities (N, N_c) and bounding boxes $(N, 4)$ respectively. N_c is the number of classes.

DETR Variants As mentioned in Section 2, there are many following works trying to resolve the issues of slow convergence and unstable training in DETR. For example, Conditional DETR [Meng *et al.*, 2021] projects the object queries of (N, C) to 2D reference points of $(N, 2)$. Similarly, DAB-DETR [Liu *et al.*, 2022] uses anchors of $(N, 4)$ as queries. Our proposed relation-enhanced attention in the next subsection can be generally added to most of the DETR variants. For experiments, we verify our method's effectiveness using the two DETR variants mentioned above.

3.3 Relation-enhanced Attention in DETR

Based on the relation analysis in Section 3.1, we propose a relation-enhanced attention mechanism in the DETR decoder (Figure 5).

Specifically, we initiate a learnable class-to-class matrix $\mathbf{A} \in N_c \times N_c$ for modeling correlations between classes in graphic design layouts. The self-attention in the L-th decoder layer is enhanced with an query-query relation weight $R \in N_q \times N_q$:

$$e_{ij} = \frac{(W_q h_i)(W_k h_j)}{\sqrt{d_k}} + R_{ij} \tag{2}$$

$$a_{ij} = \mathrm{softmax}(e_{ij}) \tag{3}$$

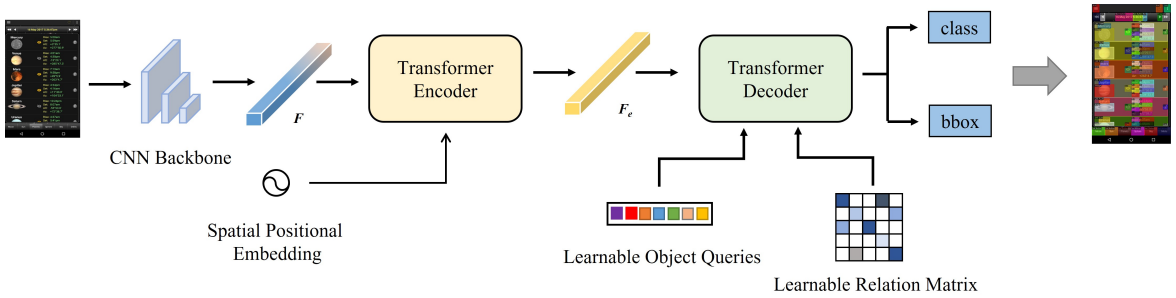


Figure 4: **DETR Pipeline + Learnable Relation Matrix.** It consists of a CNN backbone to extract image features, a Transformer encoder-decoder for query-image interaction, and two heads for class and bounding box prediction. We propose a learnable relation matrix as an additional input to the decoder for modeling class correlations.

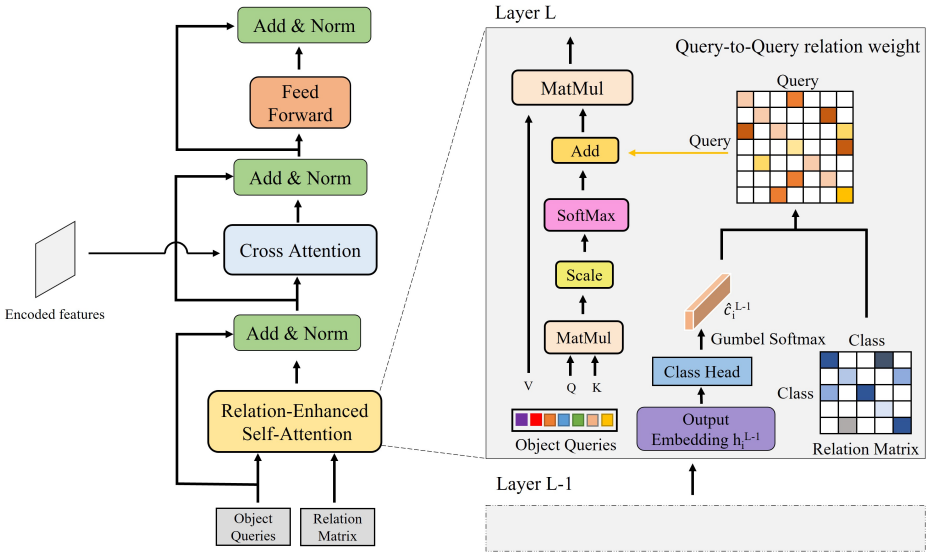


Figure 5: **Transformer decoder with relation-enhanced self-attention.** In each layer, we obtain the query-to-query relation weight R by retrieving corresponding values from the learnable relation matrix $\mathbf{A}$ using the predicted class indexes from the previous layer. We add R to the self-attention weight for updating query-to-query interaction.

where h_i is the hidden state of the i-th object query, W_q, W_k are weights of the learned linear projections for query and key in self-attention, d_k denotes the hidden size of $W_q h_i$, and a_{ij} defines the probability distribution of attention over the queries which are computed from the un-normalized attention scores e_{ij}.

To obtain R_{ij}, we use the class predictions from the previous $(L-1)$-th layer as indexes to retrieve corresponding value from matrix $\mathbf{A}$:

$$R_{ij} = \mathbf{A}[c_i^{L-1}, c_j^{L-1}] \tag{4}$$

$$c_i^{L-1} = \texttt{one-hot}(\text{argmax}\{\text{softmax}(W_c h_i^{L-1})\}) \tag{5}$$

where c_i^{L-1} is the predicted class of the i-th query with maximum probability in the previous layer, and W_c is the weight of the class prediction head.

Since the one-hot assignment operation via `argmax` is non-differentiable, we instead use `Gumbel-Softmax` [Jang *et al.*, 2016] for sampling from class distribution:

$$c_i^{L-1}[t] = \frac{\exp(W_c h_i^{L-1}[t] + \gamma_t)}{\sum_{k=1}^{N_c} \exp(W_c h_i^{L-1}[k] + \gamma_k)} \tag{6}$$

where $\{\gamma\}$ of classes $\{0, \ldots, k, \ldots, N_c\}$ are i.i.d random samples drawn from the `Gumbel(0, 1)` distribution. We apply the straight-through trick in [van den Oord *et al.*, 2017] to compute the class assignment as

$$\hat{c}_i^{L-1} = \texttt{one-hot}(\text{argmax}(c_i^{L-1})) + c_i^{L-1} - \texttt{sg}(c_i^{L-1}) \tag{7}$$

where `sg` is the stop gradient operator. With the straight-through trick, $\hat{c}_i^{L-1}$ has the one-hot value of assignment to a single class, and its gradient is equal to the one of c_i^{L-1}, which makes the relation-enhanced self-attention differentiable and end-to-end trainable.

4 Experiments

4.1 Experiment Settings

Here we introduce the datasets, baselines and evaluation metrics used in our experiments.

Datasets. We consider three publicly-available graphic design datasets: RICO, Crello and InfoPPT[1]. (1) **RICO** [Deka

[1]We do not consider DrawnUI [Ionescu *et al.*, 2020] since it is

Methods	**AP**	**AP**$_{50}$	**AP**$_{75}$	**AP**$_s$	**AP**$_m$	**AP**$_l$	**AR**$_1$	**AR**$_{10}$	**AR**$_{100}$	**AR**$_s$	**AR**$_m$	**AR**$_l$
Faster RCNN [Ren *et al.*, 2015]	50.0	58.7	53.1	6.8	27.7	50.6	42.3	61.4	63.0	9.4	35.9	63.4
Faster RCNN + SGRN [Xu *et al.*, 2019]	50.7	59.2	53.7	7.5	28.4	51.2	42.5	61.6	63.2	10.1	36.1	63.4
Faster RCNN + Magiclayout [Manandhar *et al.*, 2021]	51.0	59.4	54.0	**7.7**	**28.7**	51.5	42.8	61.8	63.4	**10.4**	36.1	63.6
Conditional-DETR [Meng *et al.*, 2021]	51.1	58.3	53.5	2.3	22.3	52.2	43.6	66.3	68.9	5.3	35.9	70.0
Conditional-DETR + SGRN	52.5	59.8	55.0	2.4	21.7	53.5	44.3	67.4	70.1	5.2	35.4	71.1
Conditional-DETR + MagicLayout	52.6	60.2	55.0	2.5	21.8	53.9	44.2	67.4	70.1	5.5	35.7	71.5
Conditional-DETR + relation (ours)	53.1	60.2	55.4	2.6	22.9	54.4	44.0	67.2	69.8	5.6	36.4	71.1
DAB-DETR [Liu *et al.*, 2022]	55.3	62.7	57.7	2.7	25	56.8	46.7	72.7	76.6	7.8	42	78.4
DAB-DETR + SGRN	55.5	63.3	58.3	3.0	24.8	57.1	47.2	73.3	77.2	7.6	41.0	79.1
DAB-DETR + Magiclayout	55.7	63.3	58.4	3.3	25	57.2	47.3	73.8	77.7	8.1	42.6	79.6
DAB-DETR + relation (ours)	**57.6**	**64.9**	**59.9**	3.1	26.1	**59.2**	**47.4**	**74.6**	**78.5**	8.5	**43.2**	**80.2**

Table 1: Performance comparison on RICO dataset.

Methods	**AP**	**AP**$_{50}$	**AP**$_{75}$	**AP**$_s$	**AP**$_m$	**AP**$_l$	**AR**$_1$	**AR**$_{10}$	**AR**$_{100}$	**AR**$_s$	**AR**$_m$	**AR**$_l$
Faster RCNN	63.8	79.7	69.0	21.9	42.2	67.2	50.0	73.8	74.8	29.4	48.4	78.3
Faster RCNN + SGRN	64.3	80.1	69.5	23.1	42.5	67.5	50.2	74.1	75.0	31.7	48.2	78.5
Faster RCNN + Magiclayout	64.2	80.1	69.1	22.0	42.0	67.3	50.1	73.9	74.9	31.6	48.3	78.5
Conditional-DETR	70.7	84.7	73.0	22.1	37.8	75.8	53.5	81.0	84.3	50.2	58.0	88.8
Conditional-DETR + SGRN	70.9	85.0	73.6	28.9	38.7	76.2	53.4	80.9	84.4	52.0	57.1	89.0
Conditional-DETR + MagicLayout	70.7	85.0	73.2	22.9	37.3	76.0	53.4	80.9	84.2	47.0	59.8	88.9
Conditional-DETR + relation (ours)	71.2	85.5	73.4	17.5	38.4	76.8	53.3	81.2	84.5	47.6	59.0	89.2
DAB-DETR	71.0	85.7	73.6	24.3	42.0	76.3	53.3	81.4	85.3	49.2	62.2	89.6
DAB-DETR + SGRN	71.2	85.8	73.4	**29.9**	41.5	76.4	53.4	81.4	85.3	**59.0**	60.6	89.7
DAB-DETR + Magiclayout	71.4	85.8	74.0	24.2	41.8	76.7	53.5	81.7	85.4	52.2	59.6	89.8
DAB-DETR + relation (ours)	**72.5**	**86.3**	**74.8**	25.9	**43.9**	**77.6**	**53.8**	**82.5**	**86.2**	52.8	**62.1**	**90.6**

Table 2: Performance comparison on Crello dataset.

et al., 2017] consists of approximately 70k unique UI screenshots from more than 9.3k Android mobile apps. We use its simplified annotations [Manandhar *et al.*, 2020] of 25 UI component classes with in total of 65K images. (2) **Crello** [Yamaguchi, 2021] contains 23k images with 5 categories, covering a wide range of designs such as social media posts, banner ads and posters collected from *crello.com*. (3) **InfoPPT** [Shi *et al.*, 2022] contains 23k information presentations collected from public websites. We parse the slides (.pptx) to obtain object annotations and then convert image inputs.

Baselines. We consider two detection backbones: Faster RCNN and DETRs (conditional DETR and DAB-DETR). Two baselines are applied to the detection framework: (1) **SGRN** [Xu *et al.*, 2019] is a graph-based feature fusion method by learning a relational graph between region proposals in Faster RCNN; (2) **MagicLayout** [Manandhar *et al.*, 2021] uses a pre-computed class co-occurrence matrix to update proposal features in Faster RCNN. When they are adapted to DETRs, we use their matrices to update the object queries similar to the region proposals in Faster RCNN.

Evaluation Metrics. Following MagicLayout, we use the standard metrics from COCO detection evaluation criteria [Lin *et al.*, 2014]: mean Average Precision (mAP) across different IoU thresholds (IoU= 0.5:0.95, 0.5, 0.75) and scales (small, medium, large). We also report Average Recall (AR) with different numbers of detection (1, 10, 100) and scales.

not currently available.

4.2 Implementation Details

We use open-source implementations of Faster-RCNN [2], Conditional DETR[3] and DAB-DETR[4]. Models are initialized using parameters pretrained on COCO. We run all the models on 8 Tesla V100 GPUs with batch size 8 for 40 epochs and AdamW [Loshchilov and Hutter, 2017] is used for training with weight decay 10^{-4}. We set different learning rates for backbone and other modules to 10^{-5} and 10^{-4} respectively. CosineAnnealing optimizer is used with T_{max} of 40 and decays it by a factor of 0.05 by the end of training. During training, images are resized such that the short side is at least 480 and at most 800 pixels and the long size is at most 1333 pixels.

4.3 Overall Results

We show the overall results in Table 1 (RICO), Table 2 (Crello) and Table 3 (InfoPPT) respectively. As we can see, DETR-based models (Conditional DETR and DAB-DETR) generally outperform Faster RCNN by a large margin. Except that in RICO, DETRs have lower accuracy in terms of AP$_s$, AP$_m$ and AR$_s$. We argue that as RICO has much more classes and larger size variation of objects, Faster RCNN with multi-scale features would handle some specific classes better. Furthermore, our relation-enhanced attention mechanism

[2]https://github.com/open-mmlab/mmdetection

[3]https://github.com/Atten4Vis/ConditionalDETR

[4]https://github.com/IDEA-Research/DAB-DETR

Methods	**AP**	**AP**$_{50}$	**AP**$_{75}$	**AP**$_s$	**AP**$_m$	**AP**$_l$	**AR**$_1$	**AR**$_{10}$	**AR**$_{100}$	**AR**$_s$	**AR**$_m$	**AR**$_l$
Faster RCNN	67.4	82.8	75.2	50.1	64.6	70.4	32.5	71.2	74.2	54.6	71.0	78.5
Faster RCNN + SGRN	67.6	83.1	75.6	52.2	65.7	70.6	32.6	71.5	74.4	55.9	71.6	78.5
Faster RCNN + Magiclayout	67.9	83.5	75.8	52.9	65.7	70.9	32.7	71.4	74.6	55.8	71.3	78.6
Conditional-DETR	67.1	85.5	74.5	58.1	65.8	73.6	32.7	74.2	79.9	68.0	78.2	87.5
Conditional-DETR + SGRN	67.7	86.2	75.1	58.1	66.4	74.6	32.8	74.6	80.2	68.5	79.4	87.7
Conditional-DETR + MagicLayout	68.0	86.6	75.2	60.6	65.6	74.6	32.4	75.1	80.7	70.6	79.6	87.8
Conditional-DETR + relation (ours)	69.1	87.1	77.0	60.2	66.9	75.7	32.8	75.6	81.2	68.6	79.6	88.5
DAB-DETR	68.7	86.8	76.4	57.6	66.7	75.0	32.8	76.0	82.4	70.7	81.1	89.5
DAB-DETR + SGRN	68.9	86.9	76.6	60.9	68.4	74.8	32.2	76.3	82.7	73.8	82.1	89.6
DAB-DETR + Magiclayout	69.4	87.5	77.3	61.8	68.1	75.5	32.9	76.2	82.5	73.4	81.2	89.6
DAB-DETR + relation (ours)	**71.2**	**88.4**	**78.7**	**63.8**	**70.0**	**77.3**	**33.4**	**77.2**	**83.6**	**76.1**	**82.4**	**90.1**

Table 3: Performance comparison on InfoPPT dataset.

has boosted the performance in both DETR variants and obtained better results than SGRN and MagicLayout. For example, under the setting of DAB-DETR, the mAP has been improved from 55.3% to 57.6% on RICO, achieving state-of-the-art performance. Similar gains can be observed in Crello (+1.5%) and InfoPPT (+2.5%).

4.4 Qualitative Results

Here we show some predicted results in Figure 6 on RICO. We keep all the detected objects with confidence scores larger than 0.6. With the use of relation matrix, our model outputs objects with generally higher confidence and more reasonable bounding boxes. For example, our model successfully detects all the `Text` objects with confidence over 0.96 in Figure 6(a). Also, our model can correctly recognize objects that are missed by the baseline DAB-DETR, such as the two white `TextButton` in Figure 6(b). For the above analysis, we use the DAB-DETR setting on RICO.

4.5 Matrix Visualization

To better understand what the relation matrix has learned, we further visualize the relation matrix heatmap in Figure 7(a). We only highlight the top-5 frequent classes in each row for better viewing. The diagonal of the matrix has the largest weight, which indicates that each class pays the most attention to its own. We can also observe high correlations between some class pairs in the learned matrix, such as <`Toolbar`, `Icon`>, <`Input`, `TextButton`>. As an example shown in Figure 7(b), the baseline DAB-DETR failed to detect a `TextButton` object ("SIGN IN" button in blue) while our relation-enhanced model can correctly identify the object with the surrounding context of `Input`. This demonstrates that our relation matrix has captured useful contextual information between class pairs for better detection accuracy.

4.6 Relation Type Ablation

To verify the effectiveness of the learnable relation, we experiment with different types of relations into the self-attention, including (1) class co-occurrence: we use the same pre-computed co-occurrence matrix as in MagicLayout; (2) spatial distance: center-point distance between two object queries using their predicted bounding boxes from previous decoder layer. As shown in Table 4, there is a relatively small improvement using the relation of class co-occurrence and spatial distance. We argue that the learnable matrix can provide more capacities for capturing relevant information and therefore shows more performance gain than the other two types of relations. Due to the space limitation, more ablation studies will be provided in the Supplementary.

Relation Type	AP	AP$_{50}$	AR$_1$	AR$_{10}$
None	55.3	62.7	46.7	72.7
Class Co-occurrence	55.9	63.6	47.2	73.4
Spatial Distance	55.5	63.6	46.7	72.7
Learnable	**57.6**	**64.9**	**47.4**	**74.6**

Table 4: Ablation of different relation types used in the decoder.

4.7 Data Augmentation for Layout Generation

Here we show a potential usage of component detection, which uses the detection outputs as augmented training data for graphic layout generation. This task aims to synthesize a set of diverse and realistic layouts. Assuming that we only have a small amount of labeling data, this experiment verifies if the detection outputs can be used as high-quality training data to improve the layout generation results. In our setting, we sample 500 layouts as the initial set and increasingly add percentages of detection outputs (in total 2k layouts) to train the layout generator. We use the state-of-the-art model LayoutTransformer [Gupta *et al.*, 2021] for generation. Three commonly-used evaluation metrics are reported: FID, Overlap and Alignment. As shown in Table 5, the generation performance gets steadily boosted with increasing amounts of augmented data, which indicates the high quality of our detection outputs. We expect better and more stable results will be obtained with a larger scale of data, which could be possibly collected by parsing layouts from numerous pixel-based designs available online.

Due to the space limitation, we show more analysis and case studies in the Supplementary.

5 Conclusion

In this paper, we study the problem of component detection in reverse engineering graphic design. We conduct a statistical analysis to demonstrate the importance of relations in graphic layouts. Built on the strong detector DETR, we present

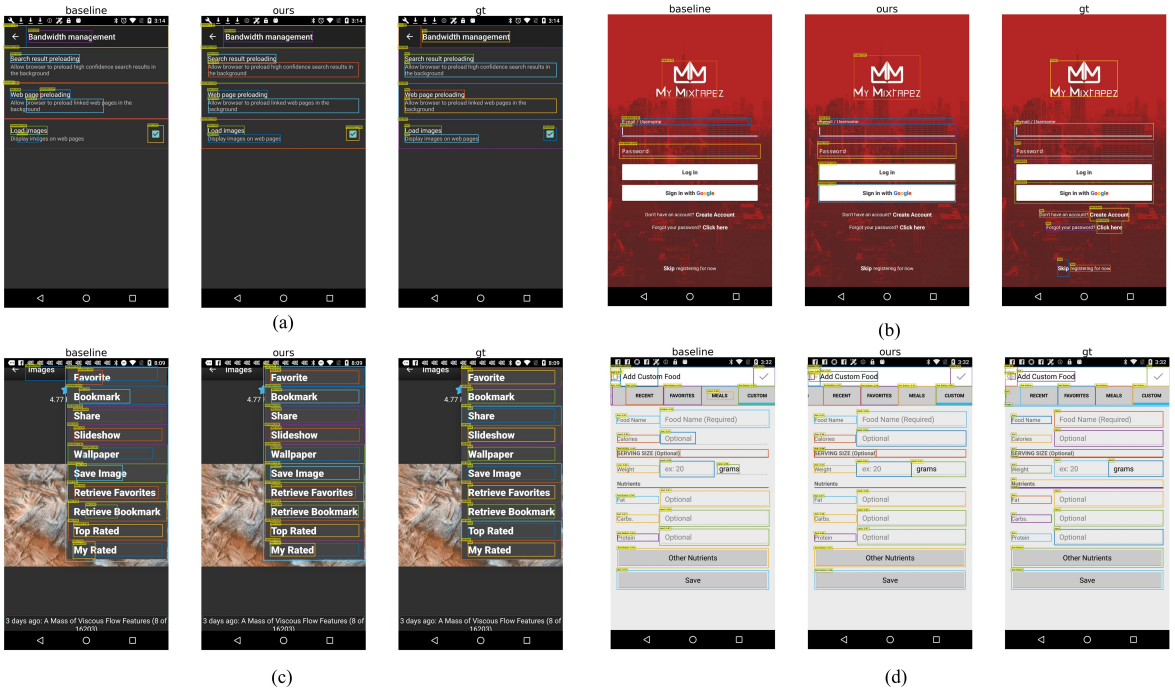


Figure 6: Examples of predictions from DAB-DETR (baseline) and our proposed relation-enhanced DAB-DETR (ours), as well as the ground truth (gt) on RICO. Our method can detect more accurately than the baseline. Zoom-in for better view.

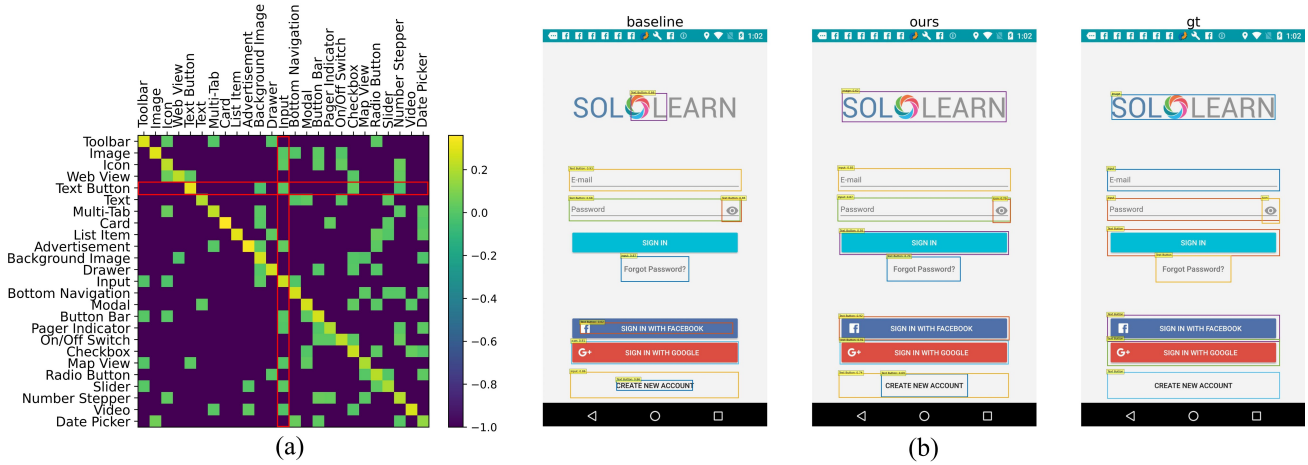


Figure 7: (a) relation matrix visualization. (b) detection examples from DAB-DETR (baseline), our model (ours) and ground truth (gt). Zoom-in for better view.

Training data	FID↓	Overlap	Align
RICO-500	37.16	122.22	**0.16**
+ 10% aug data	34.27	98.26	**0.15**
+ 50% aug data	**32.54**	94.90	**0.16**
+ 100% aug data	39.50	**90.17**	**0.15**
Real data	4.26	48.43	0.20

Table 5: Layout generation results with augmented training data from detection outputs. For the metric Align and Overlap, closer the values to real data (last row), better is the performance. Generated examples will be shown in the Supplementary.

a relation-enhanced self-attention mechanism in DETR decoder for better interactions between object queries. Our method has achieved the best results on three public datasets. Moreover, we leverage the detection outputs as augmented data to improve the layout generation performance. In the future, we plan to explore better solutions to handle the overlapping objects which are not solved well by the current model. Also we will try to use component detection to enable more applications in graphic design intelligence.

References

[Bi *et al.*, 2022] Hengyue Bi, Canhui Xu, Cao Shi, Guozhu Liu, Yuteng Li, Honghong Zhang, and Jing Qu. Srrv: A novel document object detector based on spatial-related relation and vision. *IEEE Transactions on Multimedia*, 2022.

[Binmakhashen and Mahmoud, 2019] Galal M. Binmakhashen and Sabri A. Mahmoud. Document layout analysis: A comprehensive survey. *ACM Comput. Surv.*, 52(6), oct 2019.

[Carion *et al.*, 2020] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[Chen *et al.*, 2019] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5177–5186, 2019.

[Deka *et al.*, 2017] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 845–854, 2017.

[Dixon and Fogarty, 2010] Morgan Dixon and James Fogarty. Prefab: implementing advanced behaviors using pixel-based reverse engineering of interface structure. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1525–1534, 2010.

[Gao *et al.*, 2021] Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of detr with spatially modulated co-attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3621–3630, 2021.

[Girshick, 2015] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[Gupta *et al.*, 2021] Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry S Davis, Vijay Mahadevan, and Abhinav Shrivastava. Layouttransformer: Layout generation and completion with self-attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1004–1014, 2021.

[Harley *et al.*, 2015] Adam W Harley, Alex Ufkes, and Konstantinos G Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2015.

[He *et al.*, 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, 2015.

[He *et al.*, 2017] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017.

[Ionescu *et al.*, 2020] B. Ionescu, H Müller, R Péteri, A. B. Abacha, V. Datla, S. A. Hasan, D. Demner-Fushman, S. Kozlovski, V. Liauchuk, and Y. D. Cid. Overview of the imageclef 2020: Multimedia retrieval in medical, lifelogging, nature, and internet applications. *Springer, Cham*, 2020.

[Jang *et al.*, 2016] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[Jiang *et al.*, 2018] Chenhan Jiang, Hang Xu, Xiaodan Liang, and Liang Lin. Hybrid knowledge routed modules for large-scale object detection. *Advances in Neural Information Processing Systems*, 31, 2018.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[Lin *et al.*, 2017] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

[Liu *et al.*, 2022] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*, 2022.

[Loshchilov and Hutter, 2017] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[Ma *et al.*, 2021] Tianlong Ma, Xingjiao Wu, Xin Li, Xiangcheng Du, Zhao Zhou, Liang Xue, and Cheng Jin. Document layout analysis with aesthetic-guided image augmentation. *arXiv preprint arXiv:2111.13809*, 2021.

[Manandhar *et al.*, 2020] Dipu Manandhar, Dan Ruta, and John Collomosse. Learning structural similarity of user interface layouts using graph networks. In *European Conference on Computer Vision*, pages 730–746. Springer, 2020.

[Manandhar *et al.*, 2021] Dipu Manandhar, Hailin Jin, and John Collomosse. Magic layouts: Structural prior for component detection in user interface designs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15809–15818, 2021.

[Meng *et al.*, 2021] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3651–3660, 2021.

[Moran *et al.*, 2018] Kevin Moran, Carlos Bernal-Cárdenas, Michael Curcio, Richard Bonett, and Denys Poshyvanyk.

Machine learning-based prototyping of graphical user interfaces for mobile apps. *IEEE Transactions on Software Engineering*, 46(2):196–221, 2018.

[Nguyen and Csallner, 2015] Tuan Anh Nguyen and Christoph Csallner. Reverse engineering mobile application user interfaces with remaui (t). In *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 248–259. IEEE, 2015.

[Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[Shi *et al.*, 2022] Danqing Shi, Weiwei Cui, Danqing Huang, Haidong Zhang, and Nan Cao. Reverse-engineering information presentations: Recovering hierarchical grouping from layouts of visual elements. *arXiv preprint arXiv:2201.05194*, 2022.

[van den Oord *et al.*, 2017] Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[Wang *et al.*, 2022a] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2022.

[Wang *et al.*, 2022b] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 2567–2575, 2022.

[Xu *et al.*, 2019] Hang Xu, Chenhan Jiang, Xiaodan Liang, and Zhenguo Li. Spatial-aware graph relation network for large-scale object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9298–9307, 2019.

[Yamaguchi, 2021] Kota Yamaguchi. Canvasvae: Learning to generate vector graphic documents. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5481–5489, 2021.

[Yao *et al.*, 2021] Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. Efficient detr: improving end-to-end object detector with dense prior. *arXiv preprint arXiv:2104.01318*, 2021.

[Yeh *et al.*, 2009] Tom Yeh, Tsung-Hsiang Chang, and Robert C Miller. Sikuli: using gui screenshots for search and automation. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 183–192, 2009.

[Zhang *et al.*, 2022] Gongjie Zhang, Zhipeng Luo, Yingchen Yu, Kaiwen Cui, and Shijian Lu. Accelerating detr convergence via semantic-aligned matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 949–958, 2022.

[Zhao *et al.*, 2021] Jiawei Zhao, Ke Yan, Yifan Zhao, Xiaowei Guo, Feiyue Huang, and Jia Li. Transformer-based dual relation graph for multi-label image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 163–172, 2021.

[Zhong *et al.*, 2019] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. Publaynet: largest dataset ever for document layout analysis. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1015–1022. IEEE, 2019.

[Zhu *et al.*, 2021] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021.