VASCAR: Content-Aware Layout Generation via Visual-Aware Self-Correction: Supplementary Material

Jiahao ZHANG[1,a)] Ryota YOSHIHASHI[2,b)] Shunsuke KITADA[2]
Atsuki OSANAI[2] Yuta NAKASHIMA[1]

Setting	Content		Graphic			
	Occ ↓	Rea ↓	Align ↓	Und ↑	Ove ↓	FID↓
Initial	0.210	0.0218	0.0040	0.97	0.0013	1.05
Initial + LACE	0.209	0.0233	0.0034	0.44	0.0002	1.06
VASCAR (Ours)	0.130	0.0134	0.0017	0.97	0.0012	2.13

Table 1 Comparison of post-hoc constraint optimization with LACE [1] and our VASCAR on the PKU test split. Blue and red numbers show improved and degraded metrics compared to Initial.

Setting	Content		Graphic			
	Occ ↓	Rea ↓	Align ↓	Und ↑	Ove ↓	FID↓
IoU (Ours)	**0.1304**	**0.0134**	**0.0017**	**0.97**	0.0012	**2.13**
Random	0.1740	0.0178	0.0018	**0.97**	**0.0002**	10.72
DreamSim [2]	0.1603	0.0149	0.0018	0.96	0.0010	3.77

Table 2 Comparison of results under different settings for ICL example selection on the PKU test split.

Abstract

Large language models (LLMs) have proven effective for layout generation due to their ability to produce structured description, such as HTML. In this paper, we argue that their limitation in visual understanding leads to insufficient performance in tasks requiring visual content, e.g., content-aware layout generation. Therefore, we explore whether large vision-language models (LVLMs) can be applied to content-aware layout generation and propose the training-free ***V**isual-**A**ware **S**elf-**C**orrection L**A**yout Gene**R**ation* (**VASCAR**), taking inspiration from the iterative revision of designers. VASCAR enables LVLMs (e.g., GPT-4o and Gemini) iteratively refine their outputs with reference to layout rendered layout images. Extensive experiments and user study demonstrate VASCAR's effectiveness and versatility, achieving state-of-the-art (SOTA) layout generation quality.

1. VASCAR vs. Constraint Optimization.

We further question whether correction similar to VASCAR is achievable via constraint-optimization techniques [1], [4] that explicitly minimize the layout metrics, such as Align. We compare VASCAR and the post-processing introduced in LACE [1] applied to our initial layouts (i.e., by the initial prompt). LACE considers Align and Ove, which are differentiable, as the loss and minimize them with gradient decent. Obviously, as seen in table 1, LACE's post-processing failed in improving the non-differentiable metrics. VASCAR can be seen as a gradient-free variant of such post-processing, and it has a clear advantage of reducing non-differential metrics, avoiding severe degradation thanks to LVLM's tendency to generate plausible layouts.

[1] Osaka University
[2] LY Corporation
[a)] {jiahao@is./n-yuta@}ids.osaka-u.ac.jp
[b)] {ryoshiha/s.kitada/atsuki.osanai}@lycorp.co.jp

Temperature	Content		Graphic			
	Occ ↓	Rea ↓	Align ↓	Und ↑	Ove ↓	FID↓
0	0.1956	0.0210	**0.0005**	0.76	0.0032	2.52
0.7	0.1482	0.0162	0.0010	**0.97**	0.0010	**2.00**
1.4 (Ours)	**0.1304**	0.0134	0.0017	**0.97**	0.0012	2.13
2.0	0.1319	**0.0131**	0.0027	0.96	**0.0009**	2.12

Table 3 Comparison of results across different temperature settings of LVLM on the PKU test split.

2. Further Analyses of VASCAR

In this section, we present additional experiments on VASCAR (Gemini), using a fixed setting of $I = 5$ and the PKU test split.

Impact of the number of ICL examples. When $M = 0$ (zero-shot), the models often fail to produce reasonable results [5]. To explore this effect, we evaluate VASCAR using different numbers of ICL examples. The results are summarized in table 4. Specifically, we experiment with $M = 1$, 3, 5, and 10. Our findings indicate that as M increases, performance improves across almost all metrics, with the exception of FID and Ove.

Impact of the number of output candidates. We examine the effect of the number of output candidates ($|Y_q|$) in table 4. Our findings indicate that as $|Y_q|$ increases, VASCAR's performance improves accordingly. Considering the trade-off between performance and efficiency, we set $|Y_q| = 5$ as the default setting for VASCAR.

Impact of λ on the layout scorer. We evaluate the impact of the hyperparameter λ on the layout scorer by the following configurations: (1) *Without Occ*, (2) *Without Rea*, (3) *Without Ove*, and (4) *Without Und*. The results are

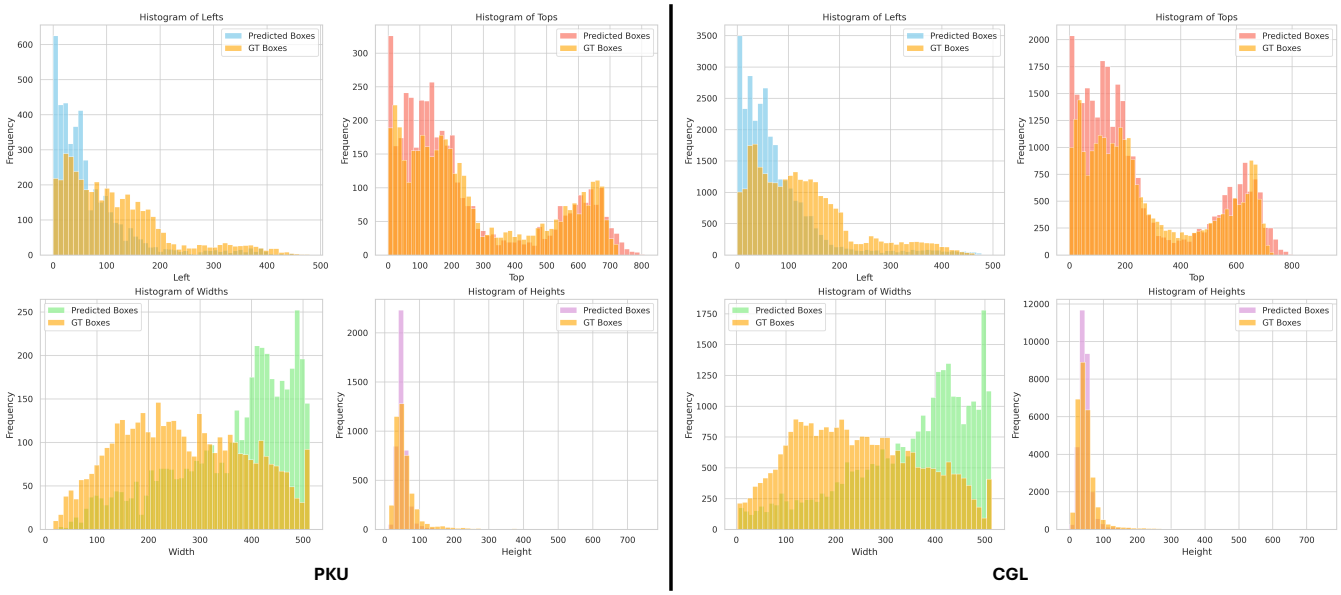


Fig. 1 The distribution of generated layout results (Predicted Boxes) (coordinates: left, top, width, and height) and ground truth (GT Boxes) for the unconstrained layout generation task on the PKU and CGL test splits.

Setting	Content		Graphic			
	Occ ↓	Rea ↓	Align ↓	Und ↑	Ove ↓	FID↓
Number of ICL Examples (M)						
1	0.2014	0.0184	0.0026	0.84	0.0017	**1.45**
3	0.1617	0.0160	0.0020	0.90	0.0016	2.27
5	0.1466	0.0150	0.0018	0.95	**0.0006**	2.25
10 (Ours)	**0.1304**	**0.0134**	**0.0017**	**0.97**	0.0012	2.13
Number of Output Candidates ($\|Y_\mathbf{q}\|$)						
1	0.1659	0.0180	0.0023	0.89	0.0011	2.29
3	0.1393	0.0150	**0.0017**	0.96	0.0010	2.24
5 (Ours)	0.1304	0.0134	**0.0017**	**0.97**	0.0012	2.13
10	**0.1225**	**0.0127**	0.0025	**0.97**	**0.0008**	**2.04**
Ablation Study on λ						
VASCAR (Ours)	0.1304	0.0134	**0.0017**	**0.97**	0.0012	2.13
Initial	0.2080	0.0218	0.0040	**0.97**	0.0013	1.05
w/o Occ	0.1709	**0.0114**	0.0022	**0.97**	**0.0002**	**1.95**
w/o Rea	0.1284	0.0162	0.0024	**0.97**	0.0008	2.03
w/o Ove	0.1301	0.0137	0.0018	**0.97**	0.0030	2.23
w/o Und	**0.1237**	0.0130	0.0023	0.68	0.0007	2.14

Table 4 Comparison of results across various experimental settings on the PKU test split. "Initial" means without self-correction. "w/o" means without. The blue and red values indicate improved and degraded metrics compared to VASCAR.

summarized in table 4. Our findings reveal that omitting a specific λ for the layout scorer leads to a significant degradation in the corresponding metric. While other metrics may show slight improvements in some cases, they remain optimized compared to the "Initial." Therefore, directly optimizing specific metrics is a reasonable approach.

2.1 Impact of ICL Example Retriever

We evaluated different approaches to ICL retrieval. *Random* means selecting ICL examples randomly from $\mathcal{D}$. *DreamSim* is to use DreamSim [2] retriever in align with Horita *et al.* [3]. *IoU* refers to use IoU of saliency map. The results are shown in table 2, indicating that *Random* selection achieved the worst performance, demonstrating that the choice of ICL examples is critical and should not be overlooked. Interestingly, retrieving ICL examples with the human-like *DreamSim* approach also led to inferior performance compared to the *IoU* selection. We argue that, in a training-free framework, selecting ICL examples based on IoU can implicitly enforce spatial information and compensate for the lack of *content constraint* within the textual prompt.

2.2 Impact of Temperature on LVLM

Temperature controls the level of randomness in token selection during generation. Lower temperatures are suited for prompts requiring deterministic or structured responses, while higher temperatures encourage more diverse and creative outputs [6]. To investigate its impact on VASCAR, we experimented with different temperature settings, with results presented in table 3. Our findings show that when the temperature is set to 0, the LVLM tends to generate fixed outputs, lacking diversity and resulting in significant degradation across almost all metrics except for Align. At a temperature of 0.7, the results are also suboptimal. Conversely, increasing the temperature to an upper bound of 2.0 leads to a substantial decline in Occ and Align metrics. These observations highlight that selecting an appropriate temperature is crucial for achieving effective and creative layout generation with VASCAR.

3. Distribution of Generated Layouts

We analyse the layouts generated by VASCAR ($I = 15$) for the unconstrained task on the PKU and CGL test splits. The coordinates of the generated layouts, including left, top, width, and height, are restored to the original image size (513×750), as shown in fig. 1. The results indicate that the generated layouts are predominantly positioned in the upper-left region (0–100 pixels) with a large width (typically 400–500 pixels). From the figures showing the distribution of the top coordinates, there is sparse distribution in

the middle regions of the canvas, suggesting that VASCAR tends to avoid placing layouts in these areas, which are often occupied by the main content.

4. More Visual Examples

We provide additional visual examples of self-correction performed using GPT-4o in fig. 2. Additionally, fig. 3 presents more visual examples from the unannotated test splits of the PKU and CGL datasets. Moreover, fig. 4 highlights visual examples from various constrained layout generation tasks on the PKU and CGL test splits.

5. Prompt Examples

For various constrained layout generation tasks, we adopt different prompts as outlined in [5]. Here, we provide detailed examples of these prompts (excluding images for simplicity) for the following tasks: $\mathbf{C} \rightarrow \mathbf{S} + \mathbf{P}$ (fig. 5), $\mathbf{C} + \mathbf{S} \rightarrow \mathbf{P}$ (fig. 6), **Completion** (fig. 7), **Refinement** (fig. 8), and **Relationship** (fig. 9). Additionally, we present an optimized prompt example in fig. 10.

References

[1] Chen, J., Zhang, R., Zhou, Y. and Chen, C.: Towards Aligned Layout Generation via Diffusion Model with Aesthetic Constraints, *The Twelfth International Conference on Learning Representations* (2024).

[2] Fu, S., Tamir, N., Sundaram, S., Chai, L., Zhang, R., Dekel, T. and Isola, P.: DreamSim: Learning New Dimensions of Human Visual Similarity using Synthetic Data, *Advances in Neural Information Processing Systems*, Vol. 36 (2024).

[3] Horita, D., Inoue, N., Kikuchi, K., Yamaguchi, K. and Aizawa, K.: Retrieval-Augmented Layout Transformer for Content-Aware Layout Generation, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 67–76 (2024).

[4] Kikuchi, K., Simo-Serra, E., Otani, M. and Yamaguchi, K.: Constrained graphic layout generation via latent optimization, *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 88–96 (2021).

[5] Lin, J., Guo, J., Sun, S., Yang, Z., Lou, J.-G. and Zhang, D.: Layoutprompter: Awaken the design ability of large language models, *Advances in Neural Information Processing Systems*, Vol. 36 (2024).

[6] Team, G., Anil, R., Borgeaud, S., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., Millican, K. et al.: Gemini: a family of highly capable multimodal models, *arXiv preprint arXiv:2312.11805* (2023).

Fig. 2 Visual examples of self-correction of VASCAR (GPT-4o, $I = 5$).

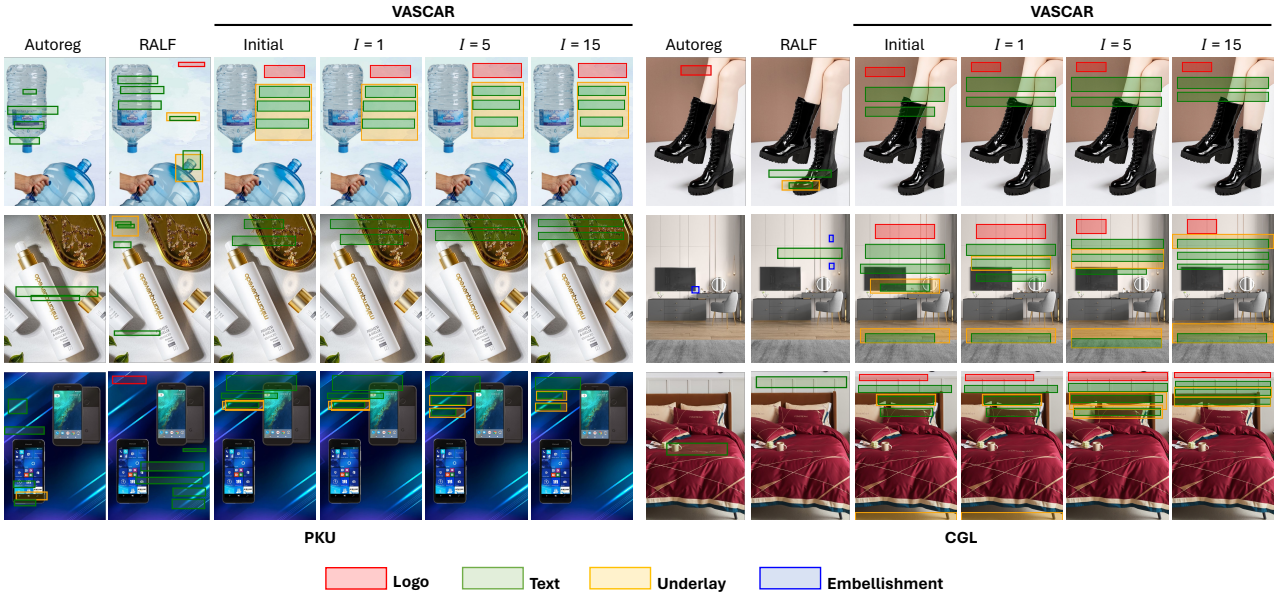


Fig. 3 Visual examples of unannotated test split of PKU and CGL datasets.

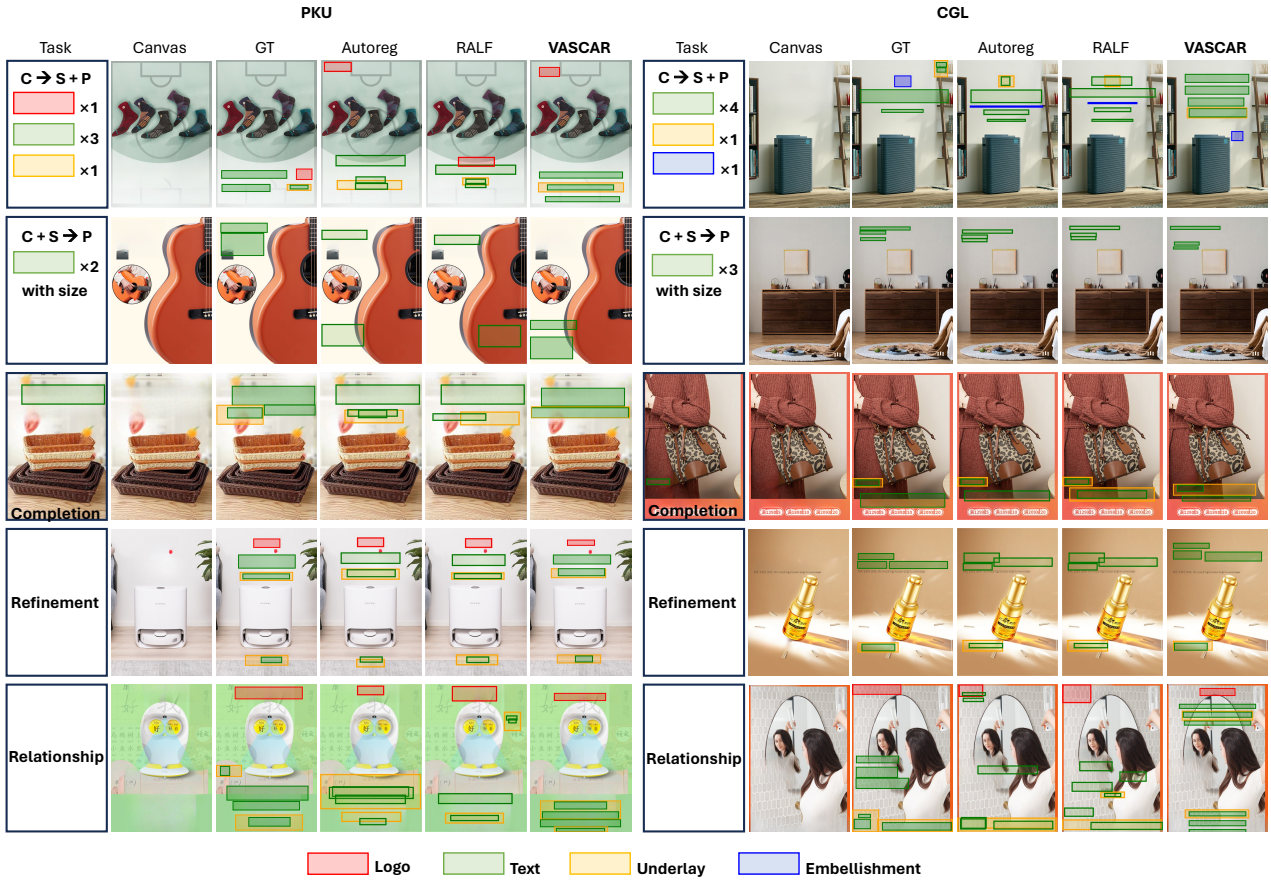


Fig. 4 The visual examples for the constrained layout generation tasks on the PKU and CGL test splits. The arrangements include the canvas, ground truth (GT), Autoreg [3], RALF [3], and VASCAR ($I = 15$).

Please generate a layout based on the given information, and return the result in the exact same HTML format as provided below. Ensure that the generated layout strictly adheres to the specified Element Type Constraint for each element in the query image.
You need to ensure that the generated layout looks realistic, with elements well aligned and avoiding unnecessary overlap.

Task Description: content-aware layout generation
Please place the following elements to avoid salient content (main content in image), ensure underlay must be under text or logo.
Important Note:
- The `Image` examples below contain layout elements such as green text, orange underlay, and red logo.
- The `Query Image` is just the original image without any elements.
- Strictly follow Element Type Constraint to determine the types and order of elements within the layout.

Layout Rules:
1. Green elements represent text and must be positioned with proper alignment.
2. Red elements represent logos and should have a prominent position without overlapping text or underlays.
3. Orange elements represent underlay backgrounds and must be placed underneath only one text or logo.

Layout Domain: poster layout
Canvas Size: canvas width is 102px, canvas height is 150px

Image 1:
Content Constraint: left 19px, top 57px, width 59px, height 63px
Element Type Constraint: logo 0 | text 1 | text 2 | text 3

```
<html>
<body>
<div class="canvas" style="left: 0px; top: 0px; width: 102px; height: 150px"></div>
<div class="logo" style="index: 0; left: 36px; top: 7px; width: 29px; height: 6px"></div>
<div class="text" style="index: 1; left: 38px; top: 23px; width: 26px; height: 9px"></div>
<div class="text" style="index: 2; left: 13px; top: 34px; width: 74px; height: 7px"></div>
<div class="text" style="index: 3; left: 16px; top: 41px; width: 63px; height: 7px"></div>
</body>
</html>
```

Image 2:
Content Constraint: left 24px, top 59px, width 52px, height 68px
Element Type Constraint: logo 0 | text 1 | text 2 | text 3 | text 4 | underlay 5 | underlay 6

```
<html>
<body>
<div class="canvas" style="left: 0px; top: 0px; width: 102px; height: 150px"></div>
<div class="logo" style="index: 0; left: 3px; top: 2px; width: 30px; height: 7px"></div>
<div class="text" style="index: 1; left: 25px; top: 15px; width: 53px; height: 5px"></div>
<div class="text" style="index: 2; left: 13px; top: 24px; width: 76px; height: 11px"></div>
<div class="text" style="index: 3; left: 36px; top: 42px; width: 24px; height: 4px"></div>
<div class="text" style="index: 4; left: 74px; top: 62px; width: 11px; height: 13px"></div>
<div class="underlay" style="index: 5; left: 31px; top: 40px; width: 40px; height: 9px"></div>
<div class="underlay" style="index: 6; left: 69px; top: 57px; width: 20px; height: 22px"></div>
</body>
</html>
```

……

Query Image:
Content Constraint: left 23px, top 57px, width 53px, height 65px
Element Type Constraint: logo 0 | text 1 | text 2 | text 3 | underlay 4 | underlay 5

Fig. 5 A prompt example of $\mathbf{C} \rightarrow \mathbf{S} + \mathbf{P}$ task.

Please generate a layout based on the given information, and return the result in the exact same HTML format as provided below. Ensure that the generated layout strictly adheres to the specified Element Type Constraint for each element in the query image.
You need to ensure that the generated layout looks realistic, with elements well aligned and avoiding unnecessary overlap.

Task Description: content-aware layout generation conditioned on given element types and sizes
Please place the following elements to avoid salient content (main content in image), ensure underlay must be under text or logo.
Important Note:
- The `Image` examples below contain layout elements such as green text, orange underlay, and red logo.
- The `Query Image` is just the original image without any elements.
- Strictly follow Element Type Constraint to determine the types and order of elements within the layout.

Layout Rules:
1. Green elements represent text and must be positioned with proper alignment.
2. Red elements represent logos and should have a prominent position without overlapping text or underlays.
3. Orange elements represent underlay backgrounds and must be placed underneath only one text or logo.

Layout Domain: poster layout
Canvas Size: canvas width is 102px, canvas height is 150px

Image 1:
Content Constraint: left 20px, top 57px, width 55px, height 71px
Element Type and Size Constraint: logo 0 29 6 | text 1 84 11 | text 2 51 6 | text 3 18 6 | text 4 18 6 | underlay 5 84 7 | underlay 6 29 29

```
<html>
<body>
<div class="canvas" style="left: 0px; top: 0px; width: 102px; height: 150px"></div>
<div class="logo" style="index: 0; left: 36px; top: 9px; width: 29px; height: 6px"></div>
<div class="text" style="index: 1; left: 8px; top: 23px; width: 84px; height: 11px"></div>
<div class="text" style="index: 2; left: 24px; top: 37px; width: 51px; height: 6px"></div>
<div class="text" style="index: 3; left: 16px; top: 59px; width: 18px; height: 6px"></div>
<div class="text" style="index: 4; left: 15px; top: 66px; width: 18px; height: 6px"></div>
<div class="underlay" style="index: 5; left: 7px; top: 36px; width: 84px; height: 7px"></div>
<div class="underlay" style="index: 6; left: 10px; top: 52px; width: 29px; height: 29px"></div>
</body>
</html>
```

Image 2:
Content Constraint: left 23px, top 62px, width 57px, height 60px | left 26px, top 65px, width 18px, height 20px
Element Type and Size Constraint: logo 0 62 12 | text 1 86 14 | text 2 38 11 | underlay 3 62 16

```
<html>
<body>
<div class="canvas" style="left: 0px; top: 0px; width: 102px; height: 150px"></div>
<div class="logo" style="index: 0; left: 18px; top: 3px; width: 62px; height: 12px"></div>
<div class="text" style="index: 1; left: 7px; top: 19px; width: 86px; height: 14px"></div>
<div class="text" style="index: 2; left: 31px; top: 35px; width: 38px; height: 11px"></div>
<div class="underlay" style="index: 3; left: 19px; top: 33px; width: 62px; height: 16px"></div>
</body>
</html>
```

……

Query Image:
Content Constraint: left 23px, top 57px, width 53px, height 65px
Element Type and Size Constraint: logo 0 27 9 | text 1 58 14 | text 2 47 5 | text 3 21 6 | underlay 4 53 9 | underlay 5 43 10

Fig. 6 A prompt example of **C + S→P** task.

Please generate a layout based on the given information, and return the result in the exact same HTML format as provided below. Ensure that the generated layout strictly adheres to the specified Element Type Constraint for each element in the query image.
You need to ensure that the generated layout looks realistic, with elements well aligned and avoiding unnecessary overlap.

Task Description: content-aware layout completion

Please place the following elements to avoid salient content (main content in image), ensure underlay must be under text or logo.
Important Note:
- The `Image` examples below contain layout elements such as green text, orange underlay, and red logo.
- The `Query Image` is just the original image without any elements.
- Strictly follow Element Type Constraint to determine the types and order of elements within the layout.

Layout Rules:
1. Green elements represent text and must be positioned with proper alignment.
2. Red elements represent logos and should have a prominent position without overlapping text or underlays.
3. Orange elements represent underlay backgrounds and must be placed underneath only one text or logo.

Layout Domain: poster layout
Canvas Size: canvas width is 102px, canvas height is 150px

Image 1:
Content Constraint: left 26px, top 59px, width 47px, height 61px
Partial Layout: logo 0 11 7 80 10

```
<html>
<body>
<div class="canvas" style="left: 0px; top: 0px; width: 102px; height: 150px"></div>
<div class="logo" style="index: 0; left: 11px; top: 7px; width: 80px; height: 10px"></div>
<div class="text" style="index: 1; left: 7px; top: 23px; width: 86px; height: 18px"></div>
<div class="text" style="index: 2; left: 18px; top: 47px; width: 65px; height: 9px"></div>
<div class="text" style="index: 3; left: 35px; top: 130px; width: 29px; height: 6px"></div>
<div class="underlay" style="index: 4; left: 13px; top: 45px; width: 75px; height: 12px"></div>
<div class="underlay" style="index: 5; left: 27px; top: 126px; width: 47px; height: 13px"></div>
</body>
</html>
```

Image 2:
Content Constraint: left 20px, top 63px, width 56px, height 60px
Partial Layout: logo 0 26 18 48 9

```
<html>
<body>
<div class="canvas" style="left: 0px; top: 0px; width: 102px; height: 150px"></div>
<div class="logo" style="index: 0; left: 26px; top: 18px; width: 48px; height: 9px"></div>
<div class="text" style="index: 1; left: 10px; top: 30px; width: 81px; height: 12px"></div>
<div class="text" style="index: 2; left: 24px; top: 45px; width: 52px; height: 5px"></div>
<div class="text" style="index: 3; left: 77px; top: 64px; width: 13px; height: 13px"></div>
<div class="underlay" style="index: 4; left: 19px; top: 43px; width: 62px; height: 9px"></div>
<div class="underlay" style="index: 5; left: 71px; top: 58px; width: 24px; height: 24px"></div>
</body>
</html>
```

……

Query Image:
Content Constraint: left 23px, top 57px, width 53px, height 65px
Partial Layout: logo 0 37 10 27 9

Fig. 7 A prompt example of **Completion** task.

Please generate a layout based on the given information, and return the result in the exact same HTML format as provided below. Ensure that the generated layout strictly adheres to the specified Element Type Constraint for each element in the query image.
You need to ensure that the generated layout looks realistic, with elements well aligned and avoiding unnecessary overlap.

Task Description: content-aware layout refinement

Please place the following elements to avoid salient content (main content in image), ensure underlay must be under text or logo.
Important Note:
- The `Image` examples below contain layout elements such as green text, orange underlay, and red logo.
- The `Query Image` is just the original image without any elements.
- Strictly follow Element Type Constraint to determine the types and order of elements within the layout.

Layout Rules:
1. Green elements represent text and must be positioned with proper alignment.
2. Red elements represent logos and should have a prominent position without overlapping text or underlays.
3. Orange elements represent underlay backgrounds and must be placed underneath only one text or logo.

Layout Domain: poster layout
Canvas Size: canvas width is 102px, canvas height is 150px

Image 1:
Content Constraint: left 20px, top 63px, width 56px, height 60px
Noise Layout: logo 0 27 22 46 7 | text 1 9 30 81 13 | text 2 22 46 52 5 | text 3 78 62 13 15 | underlay 4 20 42 61 9 | underlay 5 73 58 24 27

```
<html>
<body>
<div class="canvas" style="left: 0px; top: 0px; width: 102px; height: 150px"></div>
<div class="logo" style="index: 0; left: 26px; top: 18px; width: 48px; height: 9px"></div>
<div class="text" style="index: 1; left: 10px; top: 30px; width: 81px; height: 12px"></div>
<div class="text" style="index: 2; left: 24px; top: 45px; width: 52px; height: 5px"></div>
<div class="text" style="index: 3; left: 77px; top: 64px; width: 13px; height: 13px"></div>
<div class="underlay" style="index: 4; left: 19px; top: 43px; width: 62px; height: 9px"></div>
<div class="underlay" style="index: 5; left: 71px; top: 58px; width: 24px; height: 24px"></div>
</body>
</html>
```

Image 2:
Content Constraint: left 20px, top 57px, width 55px, height 71px
Noise Layout: logo 0 38 10 30 3 | text 1 8 22 84 10 | text 2 24 38 51 7 | text 3 15 58 18 5 | text 4 17 68 18 6 | underlay 5 9 36 83 7 | underlay 6 10 52 29 28

```
<html>
<body>
<div class="canvas" style="left: 0px; top: 0px; width: 102px; height: 150px"></div>
<div class="logo" style="index: 0; left: 36px; top: 9px; width: 29px; height: 6px"></div>
<div class="text" style="index: 1; left: 8px; top: 23px; width: 84px; height: 11px"></div>
<div class="text" style="index: 2; left: 24px; top: 37px; width: 51px; height: 6px"></div>
<div class="text" style="index: 3; left: 16px; top: 59px; width: 18px; height: 6px"></div>
<div class="text" style="index: 4; left: 15px; top: 66px; width: 18px; height: 6px"></div>
<div class="underlay" style="index: 5; left: 7px; top: 36px; width: 84px; height: 7px"></div>
<div class="underlay" style="index: 6; left: 10px; top: 52px; width: 29px; height: 29px"></div>
</body>
</html>
```

……

Query Image:
Content Constraint: left 23px, top 57px, width 53px, height 65px
Noise Layout: logo 0 37 6 26 10 | text 1 23 27 58 14 | text 2 25 49 48 7 | text 3 46 130 20 6 | underlay 4 24 45 55 9 | underlay 5 28 127 43 12

Fig. 8 A prompt example of **Refinement** task.

Please generate a layout based on the given information, and return the result in the exact same HTML format as provided below. Ensure that the generated layout strictly adheres to the specified Element Type Constraint for each element in the query image.
You need to ensure that the generated layout looks realistic, with elements well aligned and avoiding unnecessary overlap.

Task Description: content-aware layout conditioned on given element relationships
'A left B' means that the center coordinate of A is to the left of the center coordinate of B.
'A right B' means that the center coordinate of A is to the right of the center coordinate of B.
'A top B' means that the center coordinate of A is above the center coordinate of B.
'A bottom B' means that the center coordinate of A is below the center coordinate of B.
'A center B' means that the center coordinate of A and the center coordinate of B are very close.
'A smaller B' means that the area of A is smaller than the area of B.
'A larger B' means that the area of A is larger than the area of B.
'A equal B' means that the area of A and the area of B are very close.
Here, center coordinate = (left + width / 2, top + height / 2), area = width * height

Please place the following elements to avoid salient content (main content in image), ensure underlay must be under text or logo.
Important Note:
- The `Image` examples below contain layout elements such as green text, orange underlay, and red logo.
- The `Query Image` is just the original image without any elements.
- Strictly follow Element Type Constraint to determine the types and order of elements within the layout.

Layout Rules:
1. Green elements represent text and must be positioned with proper alignment.
2. Red elements represent logos and should have a prominent position without overlapping text or underlays.
3. Orange elements represent underlay backgrounds and must be placed underneath only one text or logo.

Layout Domain: poster layout
Canvas Size: canvas width is 102px, canvas height is 150px

Image 1:
Content Constraint: left 26px, top 59px, width 47px, height 61px
Element Type Constraint: logo 0 | text 1 | text 2 | text 3 | underlay 4 | underlay 5
Element Relationship Constraint: underlay 5 smaller logo 0 | text 3 bottom text 2

```
<html>
<body>
<div class="canvas" style="left: 0px; top: 0px; width: 102px; height: 150px"></div>\n<div
class="logo" style="index: 0; left: 11px; top: 7px; width: 80px; height: 10px"></div>\n<div
class="text" style="index: 1; left: 7px; top: 23px; width: 86px; height: 18px"></div>\n<div
class="text" style="index: 2; left: 18px; top: 47px; width: 65px; height: 9px"></div>\n<div
class="text" style="index: 3; left: 35px; top: 130px; width: 29px; height: 6px"></div>\n<div
class="underlay" style="index: 4; left: 13px; top: 45px; width: 75px; height: 12px"></div>\n<div
class="underlay" style="index: 5; left: 27px; top: 126px; width: 47px; height: 13px"></div>
</body>
</html>
```

Image 2:
Content Constraint: left 25px, top 60px, width 51px, height 64px
Element Type Constraint: text 0 | text 1 | text 2 | text 3 | text 4 | underlay 5 | underlay 6 | underlay 7
Element Relationship Constraint: text 1 top canvas | underlay 6 center canvas | text 4 smaller text 3 | underlay 6 center text 3 | underlay 7 larger text 4 | underlay 7 center text 4 | underlay 7 right underlay 5

```
<html>
<body>
<div class="canvas" style="left: 0px; top: 0px; width: 102px; height: 150px"></div>
<div class="text" style="index: 0; left: 2px; top: 24px; width: 96px; height: 15px"></div>
<div class="text" style="index: 1; left: 11px; top: 41px; width: 77px; height: 5px"></div>
<div class="text" style="index: 2; left: 6px; top: 49px; width: 25px; height: 7px"></div>
<div class="text" style="index: 3; left: 39px; top: 49px; width: 25px; height: 7px"></div>
<div class="text" style="index: 4; left: 71px; top: 50px; width: 24px; height: 6px"></div>
<div class="underlay" style="index: 5; left: 4px; top: 48px; width: 29px; height: 8px"></div>
<div class="underlay" style="index: 6; left: 36px; top: 48px; width: 29px; height: 8px"></div>
<div class="underlay" style="index: 7; left: 68px; top: 48px; width: 29px; height: 8px"></div>
</body>
</html>
```

……

Query Image:
Content Constraint: left 23px, top 57px, width 53px, height 65px
Element Type Constraint: logo 0 | text 1 | text 2 | text 3 | underlay 4 | underlay 5
Element Relationship Constraint: text 3 bottom canvas | underlay 5 bottom canvas | text 1 larger logo 0 | text 3 smaller logo 0

Fig. 9 A prompt example of **Relationship** task.

(Initial Prompt)
Here are the generated results of Query Image: Red is Logo; Green is Text; Orange is Underlay.

Output Image 1:
<html>
<body>

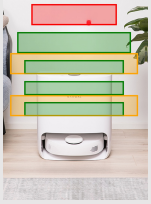
</body>
</html>


Output Image 1

The score is:
0.8929
Please refine the layout following the instruction below:
- Ensure that the underlay is properly covered by text or logos, enhancing visual appeal.
- Reduce the overlap between content constraint areas and layout elements to avoid occluding important content.
- Enhance the readability of text elements by adjusting font size, spacing, or positioning.

Output Image 2:
<html>
<body>

</body>
</html>


Output Image 2

The score is:
0.8433
Please refine the layout following the instruction below:
- Ensure that the underlay is properly covered by text or logos, enhancing visual appeal.
- Reduce the overlap between content constraint areas and layout elements to avoid occluding important content.

……

Please propose new layout; Score is high as much as possible.
(Generation by LVLM…)

Fig. 10 A prompt example of optimized prompt.