

# Seeing is Improving: Visual Feedback for Iterative Text Layout Refinement

## Supplementary Material

### 1. Dataset Details

We collected approximately 200K samples, including free and paid data from the internet. Each sample contains a background image, target text, well-formatted SVG code, and the corresponding rendered image. Each sample has an average of 9.8 text boxes, with an average text length of 84.7. Fig. 1 is a data example. Based on these data, our training and testing data were constructed, including the dataset for the Cold-Start SFT phase, the queries for the reinforcement learning phase, the training and testing datasets for the reward model, and finally, the dataset used for evaluation.

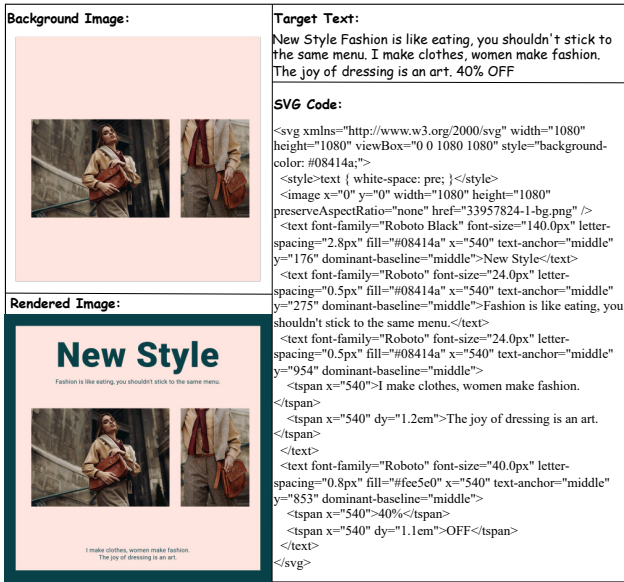


Figure 1. Data example, consisting of a background image, target text, SVG layout code, and the corresponding rendered image.

#### 1.1. Cold-Start Dataset Details

We use the method described in the main text and set the maximum number of modifications to 3 during the data generation process. Since each data trajectory includes an initial generation and a final "satisfied" output, the total number of turns is the number of modifications plus two. This process yielded a total of 8K trajectories, comprising 2,359 two-turn samples (0 modifications), 1,266 three-turn samples (1 modification), 2,030 four-turn samples (2 modifications), and 2,537 five-turn samples (3 modifications). Total data volume is 8K.

During cold-start data construction, the prompts used to guide Doubao-Seed-1.6 are prompt 1 and prompt 2, where

prompt 1 is utilized for Initial Reasoning synthesis and prompt 2 for Multi-round reflection synthesis.

#### 1.2. Layout Reward Model Dataset

Herein, a qualitative comparison of data across the four quality levels (Level-I, Level-II, Level-III, Level-IV) is presented. These four levels exhibit distinct differences in typesetting quality: Level-I demonstrates the best typesetting quality, while Level-IV, by contrast, shows the worst.

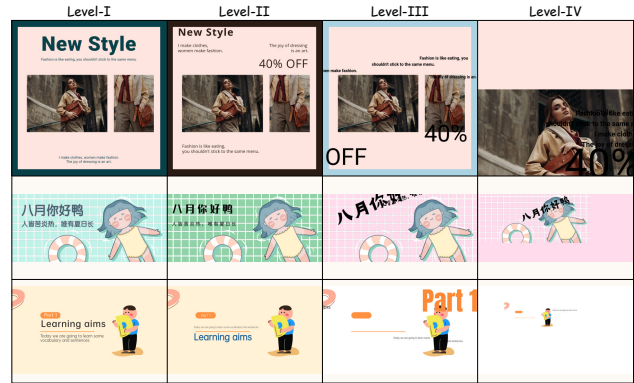


Figure 2. Qualitative comparison of four levels of data in the training data of the reward model

### 2. Training Details

#### 2.1. VFLM

All experiments are conducted on a cluster of 16 NVIDIA H200 GPUs. For the Cold-Start SFT stage, we trained the model with 8k data for 2 epochs with a learning rate of  $1 \times 10^{-5}$  and a batch size of 64. During this stage, the image `max_pixels` was set to  $1024 \times 28 \times 28$ .

For the RL stage, we set the maximum number of tool calls to 4. The weights for  $R_{ocr}$  and  $R_{svg}$  (denoted as  $\alpha$ ) were set to 0.25, while the weight for  $R_{format}$  was set to 0.1. We prepare up to 32K samples for training, with early stopping based on reward metrics during RL training. We employed a strict on-policy training strategy with the following configuration: batch size of 64, 8 rollouts per sample, sampling temperature of 1.0, KL divergence coefficient of  $1e-3$ , and learning rate of  $1e-6$ .

#### 2.2. Layout Reward Model

We trained the reward model on a preference dataset constructed from 200K layout samples. Four quality levels (Level-I, Level-II, Level-III, Level-IV) were generated for each query, yielding 1.2M preference pairs. We randomly

select 25K pairs as the test set, using the remainder for training. During training, we use a batch size of 512 and train for 2100 steps.

### 2.3. Ablation Study

In the ablation experiments, Single-Round RL, RL from Pretrained, and Direct Output adopt the same hyperparameter configuration as VFLM. For Direct Output SFT, 40k samples are used for SFT, with a batch size of 128, a learning rate of 1e-5, and training conducted for 2 epochs.

## 3. Evaluation Details

### 3.1. Evaluation metrics

We use an OCR engine<sup>1</sup> to recognize text in design images and evaluate the accuracy of rendered text using character-level f-measure. In the RL reward function,  $R_{ocr}$  and  $R_{svg}$  are evaluated using accuracy. Specifically, a character in the OCR recognition result is defined as a True Positive (TP) if it appears in the annotation; otherwise, it is classified as a False Positive (FP). A False Negative (FN) indicates that a character is only present in the annotation but absent from the OCR recognition result. Accordingly, character-level precision, recall, f-measure and accuracy can be formulated as follows:

$$\begin{aligned}
 Char\_P &= \frac{TP}{TP + FP}, \\
 Char\_R &= \frac{TP}{TP + FN}, \\
 Char\_F &= \frac{2 \times Char\_P \times Char\_R}{Char\_P + Char\_R}, \\
 Char\_Acc &= \frac{TP}{TP + FP + FN}.
 \end{aligned} \tag{1}$$

For the GPT4o evaluation, we assess the effect along four dimensions: Text Accuracy, Text-Background Harmony, Text Presentation Quality, and Meaning Expression Adaptability. The evaluation prompt is shown in prompt 5. Fig. 3 is a detailed evaluation sample.

### 3.2. Model Prompt

#### 3.2.1. VFLM System prompt

The VFLM system prompt is shown in prompt 3, includes tool definitions and task descriptions.

#### 3.2.2. MLLM System prompt

The system prompt of MLLM models, as shown in prompt 4, only omits the definition of tools and the statement of multi-round responses compared with that of VFLM.

<sup>1</sup><https://github.com/PaddlePaddle/PaddleOCR>

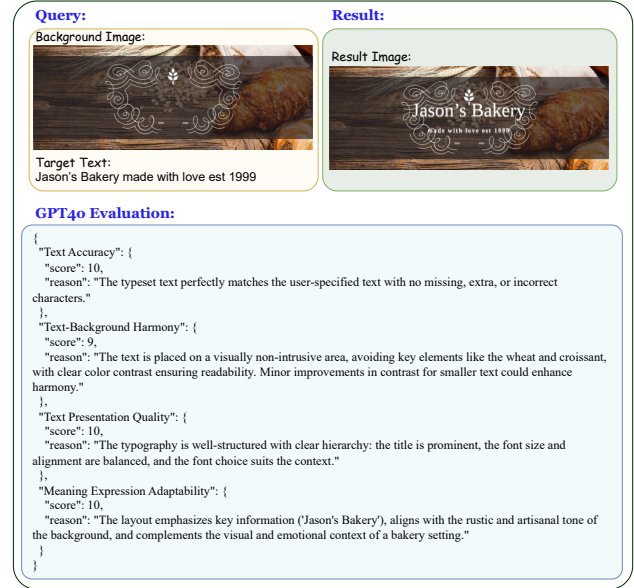


Figure 3. A detailed GPT4o evaluation output.

## 4. More Experiments

### 4.1. Number of rounds of reflection

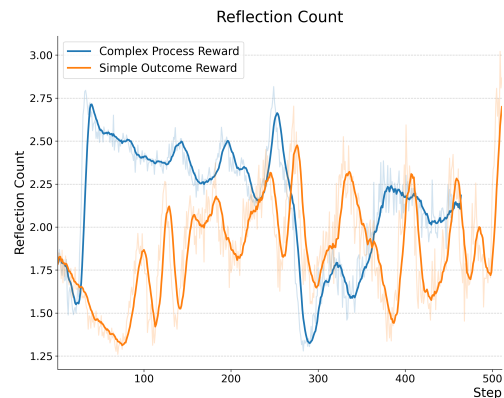


Figure 4. Training processes: reflection times of VFLM, constrained between 1 and 4.

Fig. 4 illustrates the dynamic evolution of reflection counts during training. While the complex process reward employs strategic termination control (Eq. (9)-Eq. (10)) to enforce stability, VFLM under the simple outcome reward exhibits a distinct, insightful trajectory. Specifically, in the initial phase (the first 100 steps), we observe a decline in reflection turns. Combining with Fig. 4, we can analyze this is attributed to the model's initial instability in output formatting, where iterative attempts often degraded quality compared to the initial generation, prompting the model to curtail its reasoning depth. However, beyond 100 steps, as the output format stabilizes, the model discovers that iter-

ative optimization yields superior rewards. Consequently, the reflection count begins to fluctuate and rise, reflecting the model’s autonomous realization that deeper reflection correlates with better layout quality, rather than relying on rigid, pre-defined constraints.

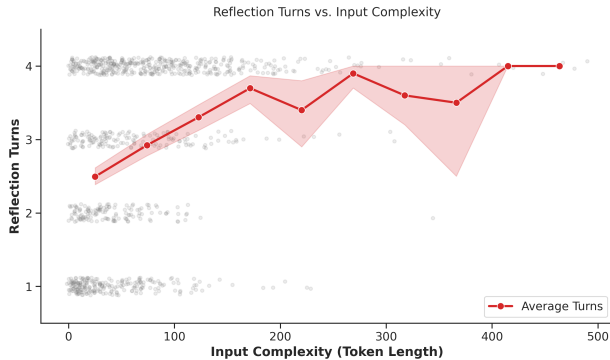


Figure 5. **Reflection Turns vs. Input Complexity.** The positive trend (red line) demonstrates that VFLM autonomously increases reflection depth for more complex tasks.

To further analyze the number of reflection rounds VFLM performs, we examined the correlation between input complexity (measured by the token length of the Target Text) and the number of reasoning rounds. As shown in Figure Fig. 5, which presents the model outputs of 1000 samples from VFLM on the TextLayout test set, the gray scatter points depict the raw distribution of inference steps, which are inherently discrete integers. To visualize the underlying trend amidst this variance, the red solid line tracks the average reflection turns across complexity intervals. A clear positive trend is observable: for concise inputs (< 100 tokens), the model efficiently converges with fewer refinement steps (averaging  $\sim 2.5$  turns). In contrast, as input complexity increases to over 400 tokens, the model adaptively increases its reasoning depth, approaching the maximum of 4 turns. This confirms that VFLM actively perceives layout difficulty and allocates computational resources accordingly.

## 4.2. Human Study

To further enhance evaluation reliability, we supplement our analysis with a blind human study involving 16 participants, who evaluated outputs from Claude3.7, GPT4o-Image, OpenCOLE, and VFLM on three dimensions across 320 randomly selected queries from three datasets. A total of 960 votes were collected. As shown in the Figure 6, VFLM performed comparably to GPT4o-Image on text coordination and achieved the best results in text accuracy and overall aesthetics.

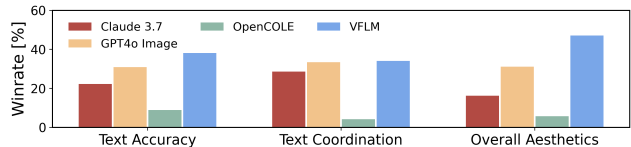


Figure 6. Results of human study.

## 4.3. Compute Cost and Latency

As shown in Figure 7, we report the average inference time of VFLM, OpenCOLE, and IGD on the TextLayout test set using 4x4090 GPUs. Although VFLM has the highest latency, it yields substantial performance gains.

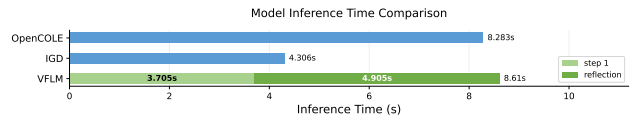


Figure 7. Comparison of inference time and latency with OpenCOLE and IGD.

## 4.4. More Baseline

Table 1. Metrics and RM score on OpenCOLE\* and MLLM feedback baseline.

Model	Char-F $\uparrow$	$R_{ali}$ $\downarrow$	$R_{ove}$ $\downarrow$	$R_{com}$ $\downarrow$	RM Score $\uparrow$
OpenCOLE*	0.7671	0.7114	0.0143	21.8279	0.3620
VFLM	<b>0.9071</b>	0.0035	<b>0.0059</b>	<b>15.4583</b>	<b>0.5415</b>
Qwen2.5-VL-7B-feedback	0.6830 $\blacktriangle$	<b>0.0025<math>\blacktriangle</math></b>	0.0188 $\blacktriangle$	28.8221 $\blacktriangledown$	0.0910 $\blacktriangledown$
GPT-4o-feedback	0.8494 $\blacktriangle$	0.0039 $\blacktriangle$	0.0028 $\blacktriangle$	20.8852 $\blacktriangledown$	0.3662 $\blacktriangle$

The table above shows the results of OpenCOLE trained on the same training set as VFLM. Although retraining brings substantial improvements over the original OpenCOLE, it still underperforms VFLM. We further integrate visual feedback into GPT-4o and Qwen2.5-VL-7B baselines. As shown in the table, without task-specific training, introducing visual feedback to MLLMs yields only marginal gains, with some metrics even showing slight declines.

## 4.5. Ablation on Other Datasets

Table 2. Ablation experiments on Graphic quality metrics and OCR metrics on Crello test set.

Model	OCR	Graphic			RM Score $\uparrow$
	Char-F $\uparrow$	$R_{ali}$ $\downarrow$	$R_{ove}$ $\downarrow$	$R_{com}$ $\downarrow$	
VFLM-step1	0.8774	0.0046	0.0061	19.5917	0.4392
VFLM	<b>0.9256</b>	0.0025	<b>0.0022</b>	<b>14.8063</b>	<b>0.5548</b>
Cold-Start-step1	0.7904	0.0118	0.0410	24.3143	0.2721
Cold-Start	0.7928	0.0121	0.0402	24.0888	0.2774
Single-Round RL	0.8829	0.0010	0.0094	24.4845	0.4007
RL from Pretrained	0.8792	<b>0.0004</b>	<b>0.0012</b>	30.3397	0.3482
Direct Output	<u>0.9092</u>	0.0009	<u>0.0022</u>	19.3983	0.4596
Direct Output SFT	0.8960	0.0032	0.0192	18.3754	<u>0.4680</u>

Table 3. Ablation experiments on Graphic quality metrics and OCR metrics on DESIGNERINTENTION test set.

Model	OCR	Graphic			RM Score $\uparrow$
	Char-F $\uparrow$	$R_{ali}$ $\downarrow$	$R_{ove}$ $\downarrow$	$R_{com}$ $\downarrow$	
VFLM-step1	0.9415	0.0033	0.0023	14.1230	0.5285
VFLM	<b>0.9663</b>	0.0024	0.0008	<b>12.1167</b>	<b>0.5688</b>
Cold-Start-step1	0.8860	0.0130	0.0264	17.5229	0.3784
Cold-Start	0.8881	0.0127	0.0257	16.9280	0.3850
Single-Round RL	0.9382	0.0006	0.0033	16.7466	0.4768
RL from Pretrained	0.9430	<b>0.0005</b>	<b>0.0001</b>	18.0272	0.4692
Direct Output	0.9590	<b>0.0005</b>	0.0010	14.6338	0.5167
Direct Output SFT	0.9434	0.0017	0.0082	13.1702	0.5398

Tab. 2 and Tab. 3 demonstrate the metrics of all models in the ablation experiments on the Crello and DESIGNER-INTENTION test sets. Consistent with the conclusions in the TextLayout dataset, VFLM also achieves significant advantages over other baselines on these two datasets, fully demonstrating the generalization ability of our visual feedback method.

#### 4.6. Ablation on Reward Functions

In our RL training process, the score for layout performance ( $R_{score}$ ) from the reward model and two rewards based on text accuracy:

$$R_{score} = R_{layout} + \alpha \cdot (R_{ocr} + R_{svg}). \quad (2)$$

We perform ablation to investigate whether using only  $R_{score}$  from the reward model is effective. To save validation time, tests are conducted on the models of the two ablation experiments: Single-Round RL and Direct Output.

Table 4. Ablation experiments on the reward function on the Text-Layout test set.

Model	OCR	Graphic			RM Score $\uparrow$
	Char-F $\uparrow$	$R_{ali}$ $\downarrow$	$R_{ove}$ $\downarrow$	$R_{com}$ $\downarrow$	
Single-Round RL	<b>0.8792</b>	0.0024	<b>0.0053</b>	<b>18.9428</b>	0.4063
Single-Round RL-w/o OCR	0.8696	0.0024	0.0054	20.0621	<b>0.4153</b>
Direct Output	<b>0.9237</b>	0.0027	<b>0.0021</b>	17.0654	0.4964
Direct Output-w/vo OCR	0.9146	<b>0.0026</b>	0.0059	<b>16.6961</b>	<b>0.5039</b>

The results are presented in Tab. 4. It can be observed from the table that the two models exhibit little difference in RL training performance with and without the  $R_{ocr}$  and  $R_{svg}$  rewards, with many metrics being relatively close. Since  $R_{ocr}$  and  $R_{svg}$  have low computational costs and the task essentially involves text typesetting, these two reward functions are retained in the final training of VFLM.

## 5. Case Study

Fig. 8 presents a detailed case generated by VFLM. The SVG initially produced by VFLM exhibits several obvious issues in terms of rendering effect; for instance, the main title "Merry Christmas" is too large and overlaps with the silhouette of the castle. After receiving visual feedback from

the rendered image, it can be observed from the model's reasoning content (highlighted in the figure) that the model identified these problems and implemented appropriate improvements. Following the second round of visual feedback, the model deemed the current layout satisfactory and output it to user as the final result. Fig. 9 is another case.

Fig. 10 show the more output cases of VFLM. Fig. 11 and Fig. 12 demonstrate the comparison between VFLM and all other models.



Target Text:  
 "Christ is Born",  
 "Merry Christmas",  
 "Glorify Him"

<think>First, determine the hierarchy of the text. The main title "Merry Christmas" should be the most prominent, so I choose a large font size (108px), use the "Source Han Serif SC SemiBold" font, set the color to #eec0d8, and center it at the position (602, 255).  
 Next, for the text "Christ is Born Glorify Him", as it is a subtitle, I use the "Source Han Sans SC Heavy" font, size 21.7px, letter-spacing 3.3px, and the color #eec0d8. I center it at (602, 331).  
 Throughout the process, I ensure that the text is centered by setting the x-coordinate to 602 (the horizontal center of the 1200px width background) and adjust the y-coordinate to position the text appropriately in the vertical space.  
 Let me export the image first to see how the layout looks.</think>  
 <tool\_call>  
 TOOL: svg\_to\_image\_tool  
 PARAMS:  
 svg\_code:  
 ```svg  
 <svg xmlns="http://www.w3.org/2000/svg" width="1200" height="628" style="background-color: #8865aa;">  
 <style>text { white-space: pre; }</style>  
 <image x="0" y="0" width="1200" height="628" preserveAspectRatio="none" href="background-image.png" />  
 <text font-family="Source Han Serif SC SemiBold" font-size="108.0px" fill="#eec0d8" x="602" text-anchor="middle" y="255" dominant-baseline="middle">Merry Christmas</text>  
 <text font-family="Source Han Sans SC Heavy" font-size="21.7px" letter-spacing="3.3px" fill="#eec0d8" x="602" text-anchor="middle" y="331" dominant-baseline="middle">Christ is Born Glorify Him</text>  
 </svg>  
 </tool\_call>



<think>I will check the SVG rendering effect of my version. First, **the main title "Merry Christmas" is too large and overlaps with the silhouette of the castle, making the overall layout feel cluttered. The text color also blends too much with the background, reducing readability.** On the positive side, the text content is complete with no missing parts. To improve, **I need to restructure the text hierarchy and positioning.** I'll reduce the font size of "Merry Christmas" to make it more proportional to the background, adjust its position to the upper middle area to avoid overlapping with the castle silhouette, **change the color to a warmer tone** like #fcedd9 to enhance contrast with the background, and split the text into two lines for better visual balance. Next, I will improve my SVG code.</think>  
 <tool\_call>  
 TOOL: svg\_to\_image\_tool  
 PARAMS:  
 svg\_code:  
 ```svg  
 <svg xmlns="http://www.w3.org/2000/svg" width="1200" height="628" style="background-color: #926c98;">  
 <style>text { white-space: pre; }</style>  
 <image x="0" y="0" width="1200" height="628" preserveAspectRatio="none" href="background-image.png" />  
 <text font-family="Source Han Sans SC Heavy" font-size="51.0px" fill="#fcedd9" x="600" text-anchor="middle" y="157" dominant-baseline="middle">Merry Christmas</text>  
 <text font-family="Source Han Sans SC Heavy" font-size="21.0px" letter-spacing="3.1px" fill="#fcedd9" x="600" text-anchor="middle" y="219" dominant-baseline="middle">Christ is Born Glorify Him</text>  
 </svg>  
 </tool\_call>



<think>I will check the SVG rendering effect of this version. The main title "Merry Christmas" is now properly positioned in the upper middle area, with an appropriate font size and color that contrasts well with the background, making it the focal point of the layout. The subtitle "Christ is Born Glorify Him" is centered below the main title, creating a clear visual hierarchy. There is no text overlapping with the silhouette elements of the background, and all text content is complete and clearly readable. The overall layout feels balanced and harmonious, effectively conveying the Christmas message. **I believe the current SVG layout effect is very good and can be output to the user.**</think>  
 <answer>  
 ```svg  
 <svg xmlns="http://www.w3.org/2000/svg" width="1200" height="628" style="background-color: #926c98;">  
 <style>text { white-space: pre; }</style>  
 <image x="0" y="0" width="1200" height="628" preserveAspectRatio="none" href="background-image.png" />  
 <text font-family="Source Han Sans SC Heavy" font-size="51.0px" fill="#fcedd9" x="600" text-anchor="middle" y="157" dominant-baseline="middle">Merry Christmas</text>  
 <text font-family="Source Han Sans SC Heavy" font-size="21.0px" letter-spacing="3.1px" fill="#fcedd9" x="600" text-anchor="middle" y="219" dominant-baseline="middle">Christ is Born Glorify Him</text>  
 </svg>  
 </answer>



Figure 8. A detailed VFLM output case.

Target Text:  
超值组合  
招牌牛肉套餐

<think>First, I need to determine the placement of two texts: "超值组合" and "招牌牛肉套餐". Looking at the background, the red area on the right is a suitable space, so I center both texts there. For "招牌牛肉套餐", as the key information, I choose a bold font (Source Han Sans SC Heavy) with a larger size (50px) to emphasize it. For "超值组合", I use a regular font (Source Han Sans SC Regular) with a slightly smaller size (29px) to create a hierarchy.

To make the text stand out against the red background, I use a double - outline effect. I define each text in <defs> first. Then, I create a mask: the mask combines a white stroke and a black fill of the text. After that, I use <use> to apply the fill color (#ff3b7) for "超值组合" and #ee1c1e for "招牌牛肉套餐" and the masked stroke (#ae1311). The stroke width is set to 4px, which is wide enough for a clear outline without being too thick.

Both texts are center - aligned using text - anchor="middle". "招牌牛肉套餐" is positioned higher (y = 61) as the main headline, and "超值组合" is below it (y = 116) as a supporting line. Using white - space: pre in the style ensures proper text formatting. Reusing <use> elements for fill, stroke, and mask keeps the code efficient while achieving the desired visual effects.

Let me export the image first to see how the layout looks.</think>

```

<tool_call>
TOOL: svg_to_image_tool
PARAMS:
svg_code:
'''svg
<svg xmlns="http://www.w3.org/2000/svg" width="686" height="176" viewBox="0 0 686 176" style="background-color: #fff;">
<style>text { white-space: pre; }</style>
<image x="0" y="0" width="686" height="176" preserveAspectRatio="none" href="background-image.png" />
<defs>
<text id="text-1" font-family="Source Han Sans SC Regular" font-size="29.0px" x="454" text-anchor="middle" y="116" dominant-baseline="middle">超值组合</text>
<mask id="text-1-outer-1">
<use href="#text-1" stroke="white" stroke-width="4" />
<use href="#text-1" fill="black" />
</mask>
</defs>
<defs>
<text id="text-2" font-family="Source Han Sans SC Heavy" font-size="50.0px" x="454" text-anchor="middle" y="61" dominant-baseline="middle">招牌牛肉套餐</text>
<mask id="text-2-outer-1">
<use href="#text-2" stroke="white" stroke-width="4" />
<use href="#text-2" fill="black" />
</mask>
</defs>
</defs>
<use href="#text-1" fill="#ff3b7" />
<use href="#text-1" fill="none" />
<use href="#text-1" mask="url(#text-1-outer-1)" stroke="#ae1311" stroke-width="4" />
<use href="#text-2" fill="#ee1c1e" />
<use href="#text-2" fill="none" />
<use href="#text-2" mask="url(#text-2-outer-1)" stroke="#ae1311" stroke-width="4" />
</svg>
'''
</tool_call>

```

<think>I will check the SVG rendering effect of my version. First, I notice that both key texts "超值组合" and "招牌牛肉套餐" are present and not overlapping, which is a good foundation. However, the visual hierarchy seems inverted. The prominent red background area on the right should highlight the main promotional message, but currently "招牌牛肉套餐" is placed there in large red text - this causes poor readability against the red background and weakens the focus. Meanwhile, "超值组合" is confined to the small white rectangular box below, reducing its impact as a key selling point.

To improve, I need to swap the text content between the two positions. "超值组合" should be the headline, placed prominently on the red background with a larger size and white color for contrast. "招牌牛肉套餐" should then occupy the white box, using a smaller size and dark red color to complement the box. I'll adjust the first text element's content to "超值组合", set font-size to 72px, fill to "#fff", and center it horizontally on the red area (x around 328, y around 61). The second text element will be updated to "招牌牛肉套餐", with font-size 32px, fill "#b50101", and centered in the white box (x around 350, y around 120). This will fix readability and establish correct visual hierarchy. Next, I will improve my SVG code.</think>

```

<tool_call>
TOOL: svg_to_image_tool
PARAMS:
svg_code:
'''svg
<svg xmlns="http://www.w3.org/2000/svg" width="686" height="176" viewBox="0 0 686 176" style="background-color: #fff;">
<style>text { white-space: pre; }</style>
<image x="0" y="0" width="686" height="176" preserveAspectRatio="none" href="background-image.png" />
<text font-family="PangZhenGui-PMZD" font-size="72.0px" letter-spacing="-5.6px" fill="#fff" x="328" y="61" dominant-baseline="middle">超值组合</text>
<text font-family="PangZhenGui-PMZD" font-size="32.0px" letter-spacing="-1.3px" fill="#b50101" x="350" y="120" dominant-baseline="middle">招牌牛肉套餐</text>
</svg>
'''
</tool_call>

```

<think>I will check the SVG rendering effect of this version. The layout now presents a clear visual hierarchy: "超值组合" stands out as the main headline in large white text against the red background, immediately capturing attention, which effectively emphasizes the promotional focus. "招牌牛肉套餐" is neatly positioned within the white rectangular area with red text, creating strong contrast against the white background for excellent readability. Both texts are centered appropriately, with no overlapping, blocking, or exceeding the background's aesthetic, making the promotional message clear and visually appealing. The current SVG layout effect is very good and can be output to the user.</think>

```

<answer>'''svg
<svg xmlns="http://www.w3.org/2000/svg" width="686" height="176" viewBox="0 0 686 176" style="background-color: #fff;">
<style>text { white-space: pre; }</style>
<image x="0" y="0" width="686" height="176" preserveAspectRatio="none" href="background-image.png" />
<text font-family="PangZhenGui-PMZD" font-size="72.0px" letter-spacing="-5.8px" fill="#fff" x="328" y="61" dominant-baseline="middle">超值组合</text>
<text font-family="PangZhenGui-PMZD" font-size="32.0px" letter-spacing="-1.3px" fill="#b50101" x="350" y="120" dominant-baseline="middle">招牌牛肉套餐</text>
</svg>
'''
</answer>

```

Figure 9. Another detailed VFLM output case.

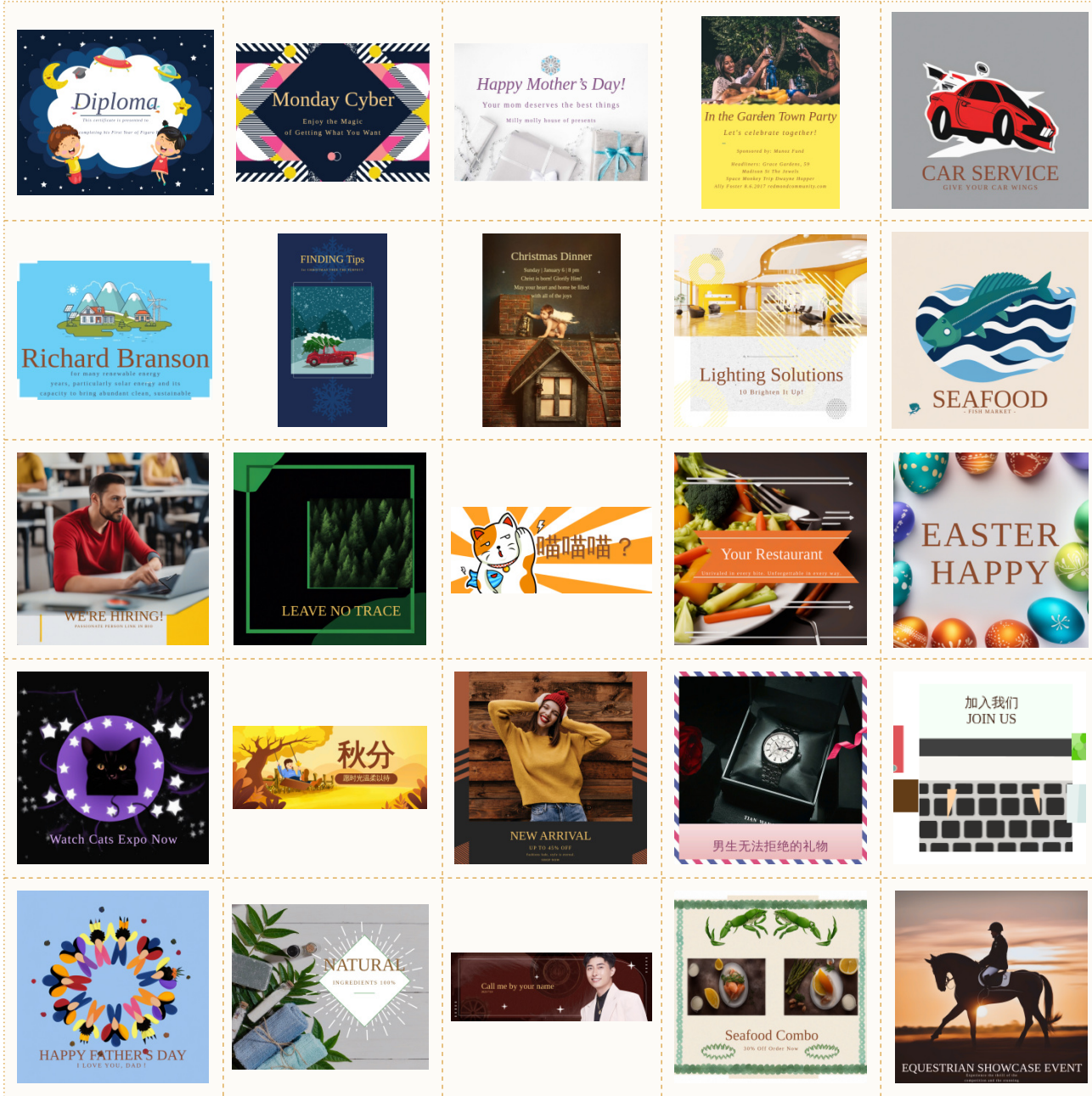


Figure 10. More VFLM output cases.

Target Text

www.fridayfantasy.com  
For more information visit  
our website  
FANTASY FILM FESTIVAL  
FRIDAY NIGHT MAGIC

BURGER  
Club  
OPEN  
24/7

FISH MARKET ORDER  
NOW SELLING FRESH  
SEAFOOD ON ICE GET  
THE BEST PRICE FROM  
US

认识兴趣是力求认识世界，兴趣是最好的老师，学生有心曲的学习，即可在学习活动中体验效率，又可体验成功过，提高效率，所以应培养并激发学生学习的兴趣，才能保证课堂的效率。关键词：兴趣、学习、数学、效率 导论 INTRODUCTION

·热爱互联网行业，对未来充满期待·好奇心强，喜欢探索研究新事物·自驱力强，时时刻刻鞭策自己，敢想敢做·有较强的目标导向意识及抗压能力，沟通能力强 我们希望

Background image

GPT4o

Claude3.7

Doubao-Seed-1.6

Qwen2.5-VL-72B

GPT4o-Image



Figure 11. In comparison with all existing methods(Part1).

|                  |                                                                                                                                                                           |                                  |                                                                                      |                                                                                                                                                                   |                                                                                     |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------|--------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| Target Text      | <p>www.fridayfantasy.com<br/>For more information visit our website<br/>FANTASY FILM FESTIVAL<br/>FRIDAY NIGHT MAGIC</p>                                                  | <p>BURGER Club<br/>OPEN 24/7</p> | <p>FISH MARKET ORDER NOW SELLING FRESH SEAFOOD ON ICE GET THE BEST PRICE FROM US</p> | <p>认识兴趣是力求认识世界，兴趣是最好的老师，学生有心曲的学习，即可在学习活动中体验效率，又可体验成功过，提高效率，所以应培养并激发学生学习的兴趣，才能保证课堂的效率。关键词：兴趣、学习、数学、效率 导论 INTRODUCTION</p>                                          | <p>热爱互联网行业，对未来充满期待，好奇心强，喜欢探索研究新事物，自驱力强，时时刻刻鞭策自己，敢想敢做，有较强的目标导向意识及抗压能力，沟通能力强 我们希望</p> |
| Background image |                                                                                                                                                                           |                                  |                                                                                      |                                                                                                                                                                   |                                                                                     |
| Qwen-Image-Edit  | <p>www.fridayfantasy.com<br/>www.fridayfantasy.com<br/>for more information visit our website<br/>FANTASY FILM FESTIVAL<br/>FRIDAY NIGHT MAGIC<br/>FRIDAY NIGHT MAGIC</p> | <p>BURGER Club<br/>OPEN 24/7</p> | <p>FISH MARKET ORDER NOW SELLING FRESH SEAFOOD ON ICE GET THE BEST PRICE FROM US</p> | <p>认识兴趣是力求认识世界，兴趣是最好的老师，学生有心曲的学习，飞舟兴学，幸在以习惯难以逾越的逆境功绩成学在培养优越，懂得理更理向卓越的弄，驱动奇培养穿激发效应保那地堂效率；关键词：学习 求站：座学、体何由体斗，才不侯切非动派基三 语：SITTY EUIDUSTORICE</p> <p>INTRODUCTION</p> | <p>热爱互联网行业，对未来充满期待，好奇心强，喜欢探索研究新事物，自驱力强，时时刻刻鞭策自己，敢想敢做，有较强的目标导向意识及抗压能力，沟通能力强 我们希望</p> |
| FLUX Kontext     |                                                                                                                                                                           |                                  |                                                                                      |                                                                                                                                                                   |                                                                                     |
| IGD              |                                                                                                                                                                           |                                  |                                                                                      |                                                                                                                                                                   |                                                                                     |
| OpenCOLE         |                                                                                                                                                                           |                                  |                                                                                      |                                                                                                                                                                   |                                                                                     |
| VFLM             |                                                                                                                                                                           |                                  |                                                                                      |                                                                                                                                                                   |                                                                                     |

Figure 12. In comparison with all existing methods(Part2).

## Initial Reasoning Process Prompt

### Role setting:

You are an experienced Layout and SVG engineer.

### Task:

Here is a result of using SVG code to typeset specific text on an input background image. I will provide you with the designed SVG code and the rendered image of this code, which has a very beautiful layout effect.

Now, assuming you are the designer of typesetting this SVG, what is your thought process when typesetting this SVG?

Could you please use the voice of a designer to briefly describe your thought process when designing this SVG based on the SVG code and rendering results? How did you design this SVG?

Ensure that your design ideas are consistent and closely related to the design results of this SVG. Do not fabricate content that is not included in the SVG, as the SVG only typesets the given text based on the given background image. Therefore, the typesetting ideas should focus on the implementation of the text, including text position, font, size, beautification effect, etc. Don't pay attention to the text already in the background image. Do not pay attention to unnecessary other graphic icons and other elements, and do not analyze how good the rendered image is from the perspective of typesetting, because your thinking is based on not seeing the rendered image.

```
background-image.png: <image>
image size : {image_width} * {image_height}
texts: {target_text}
```

### svg code:

```
```svg
{svg_code}
```
```

The image result rendered by this svg: <image>

### Output requirements:

Please directly output the thinking process of the person in the designer's tone, without any other descriptive content. Answer in English.

## Multi-Round Reflection Data Synthesis Prompt

USER:

You are an experienced typesetter and SVG engineer, skilled at elegantly typesetting specified text on user provided background images. You know how to apply unique aesthetic principles to design professional and attractive layout, using SVG code to create beautiful layouts. Please design the final layout plan based on the background image and text content provided by the user. In SVG code, use the image tag to reference the background image: href="background-image.png", while other elements only need to be designed with content related to the text. Please design an SVG code layout scheme based on the background image and text content provided by the following users. You first think about it, and then output the final SVG code. The format is<think>...</think>\n<answer>...</answer>

```
background-image.png: <image>
image size: {image_width}*{image_height}
Please arrange the following text on the background image:
{target_text}
```

ASSISTANT:

{Reasoning Process and SVG generated by Qwen2.5-VL-7B}

USER:

Now, I'll show you the effect of this version of SVG layout, and you need to improve this SVG layout effect.

I'll also give you a standard SVG layout result, and you need to improve your SVG layout according to this standard SVG layout result.

- In your output, you need to speak in the tone of a designer, stating that you've reviewed the SVG result of your initial layout, then reflected on it and made corrections. Note that you've designed an initial version of the SVG, and now I've provided you with the rendered image. Your output should focus on examining the image, ensuring it's a reflection and correction of your initial SVG layout result. The direction of correction is the correct effect I gave you, but don't expose in the output that you're improving based on the standard effect. Pretend you've thought it out on your own.

- The output should include your thinking process for SVG layout, how to improve your SVG layout result step by step. You need to point out which parts of your initial layout were good and which were bad and needed modification. For each modification point, be specific about how to modify the SVG code. Don't just qualitatively say which aspects you'll modify. Pay attention to the tone, which should be like that of a designer, and the content of the output should conform to the designer's way of thinking.

- During the modification process, key considerations should be text position, whether there is any text missing, text overlapping, text being blocked, and whether the text exceeds the background image range, etc. These considerations need to be included in the output.

- Your output modification process may involve multiple steps. If your initial layout is not very different from the standard one, you can make only one modification; if there is a large gap, multiple steps of modification are required. You need to simulate the designer's thinking process and gradually improve the SVG layout. Each time you modify, choose the part with the worst effect to improve. Explain the specific SVG code improvements in the thinking process. After modifying one version, only make changes to the SVG part that needs to be modified in this step, and don't change the other parts for now. Output the complete SVG code; then proceed to the next modification until you think the SVG layout effect is very good. Don't make too many modifications. Ensure that each modification is better than the previous one, with a maximum of 3 modifications. The SVG code after the last modification needs to be output, and its effect should be the same as that of the standard code I gave you.

- You need to answer one modification each time, and then I'll show you the rendered effect of the SVG you modified, and you'll make the next modification.

- Based on the rendered image effect I give you after each of your modifications, decide whether the next modification is needed. Each modification should have a significant improvement, not just a minor one. For example, when the order of different text tags doesn't affect the SVG rendering effect, there's no need for additional modification. Since I require you to make as few steps of modification as possible, each modification should have a significant improvement.

Your output is the thinking process of a designer improving the SVG layout after reviewing the first version they designed. I've given you the standard SVG code, and you should modify the SVG code in this direction. However, note that your output is based on not having seen this standard SVG, as if the designer is reflecting after designing the initial draft and modifying it to the final standard SVG version through multiple steps.

After the last modification, you need to output the final inspection, indicating that after checking the image, you think the current SVG layout effect is very good and can be replied to the user.

Your initial SVG layout effect is shown in the figure below:

<image>

This is standard and beautiful SVG code. The code and its rendered effect diagram are as follows.

```
```svg
{svg_code}
```
```

<image>

Output requirements:

Please directly output the thinking process in the tone of a designer, without any other descriptive content. Be careful not to reveal that you have seen the standard SVG effect. Transform it into your own thinking. The output should conform to the designer's thinking process, that is, how you think about improving the layout by yourself, not by comparing with the standard effect. Do not output the word "standard".

If improvement is needed, the first sentence in each step of the thinking process should be: "I will check the SVG rendering effect of my version...", and the last sentence should be: "Next, I will improve my SVG code."

These two beginning and ending sentences are necessary and cannot be omitted, but you can modify the language to maintain the same meaning and make the output diverse.

Your output needs to specifically point out which effects in your first version are good and do not need improvement, which effects are poor and need improvement, and how to specifically modify the SVG code. If you think the SVG layout effect of your first version is very close to or even better than the standard SVG rendering effect I provided, you can describe your satisfaction with this SVG layout and that you think it has achieved a very good effect and does not need further improvement.

Answer in English.

Output requirements:

- You need to output in the form of multi - round conversations. According to the number of modifications you decide, the output format for each modification is as follows:

# Step {current modification number} of modification:

## Thinking process: Here, think about how to make the modification.

## SVG code: Modify the complete SVG code.

- After the final modification, the rendering result of your SVG code should be exactly the same as that of the reference SVG code I provided.

- After the last modification is output, I will provide you with the rendered image again. Then you need to output a final reflection, indicating that you will check the SVG rendering effect of this version and think that the current SVG layout effect is very good and does not need to be improved further, and it can be output to the user. The output format of the final reflection is:

# Final rethink: ...

USER:

Your current SVG layout effect is shown in the figure below:

<image>

ASSISTANT:

...

## VFLM System Prompt

System:

You are a helpful assistant.

# Tools

You may call one or more functions to assist with the user query.

You are provided with function signatures within <tools></tools> XML tags:

```
<tools>
{
  "type": "function ",
  "function": {
    "name": "svg_to_image_tool",
    "description": "Convert SVG code to an image.",
    "parameters": {
      "type": "object",
      "properties": {
        "svg_code": {
          "type": "string",
          "description": "The SVG code to convert to an image."
        }
      },
      "required": [
        "svg_code"
      ]
    }
  }
}
</tools>
```

# How to call this tool

Wrap the SVG code with specific markers (`` and ``) within <tool\_call></tool\_call> XML tags.

**\*\*Example\*\*:**

```
<tool_call>
TOOL: svg_to_image_tool
PARAMS:
svg_code:
``svg
...
</tool_call>
```

You are an experienced visual layout designer and SVG engineer, skilled at elegantly typesetting specified text on background images provided by users.

You know how to apply unique aesthetic principles to design professional and appealing layouts, using SVG code to create beautiful layouts. Please design a final layout plan based on the background image and text content provided by the user.

In the SVG code, use the image tag to reference the background image:

href=\"background-image.png\", and other elements only need to design content related to the text.

Please design an SVG code layout plan based on the following background image and text content provided by the user.

You should first view the background image, think about how to typeset the text on the background image, design a version of SVG code, correctly reference the background image in the SVG code, then call the `svg_to_image` tool, and you will get the picture of your SVG. Then, based on the picture, judge whether the typesetting of your picture meets the expectations, whether the background image is correctly referenced, and whether the text is beautiful. If the typesetting effect is not good enough, modify the SVG code, and repeatedly reflect after tool calls until the typesetting effect is better. Finally, output the final SVG code.

Format: `<think>...</think>\n<tool_call>...</tool_call>`(if tools needed)  
`<answer>...</answer>`

User:

`background-image.png: <image>`  
`image size: {image_width}*{image_height}`  
`texts: {text}`

### MLLM System Prompt

System:

You are an experienced layout designer and SVG engineer, proficient in elegantly laying out specified text on a background image provided by the user. You have a deep understanding of how to use unique aesthetic principles to design a professional and attractive layout. Use SVG code to create a beautiful layout. Please design the final layout plan according to the background image and text content provided by the user. In the SVG code, use the image tag to reference the background image: `href="background-image.png"`, and only design the elements related to the text. Please design the SVG code layout plan according to the background image and text content provided below.

User:

`background-image.png: <image>`  
`image size: {image_width}*{image_height}`  
`texts: {text}`

## GPT4o Evaluation Prompt

You are an autonomous AI Assistant specializing in evaluating the typesetting effects of a typesetting model. This model's core task is to typeset user-specified text on a background image; your goal is to provide objective, targeted, and constructive scoring and feedback based on text-typesetting-specific principles and practical application needs. Your evaluation covers four independent dimensions: text content accuracy, text-background visual harmony, text presentation quality, and meaning expression adaptability. You will be provided with the background image, the user's original specified text, and the typeset result (background image + typeset text). Your task is to score the typesetting effect objectively based on the following 4 criteria and provide concise reasoning for each score.

### Scoring rules:

- For each of the 4 criteria, score objectively and rigorously on an independent scale of 1-10. For a single criterion, a score of 10 means flawless performance (no issues, fully meeting expectations); a score of 7 indicates minor flaws (no impact on core performance); a score of 4 reflects significant shortcomings (affecting core performance); a score of 1-2 signifies severe issues (rendering the function of this criterion ineffective).
- Keep reasoning concise (1-2 sentences per criterion), focusing on specific performance. If the output is too long, it will be truncated.
- Only respond in JSON format with 4 top-level keys corresponding to the 4 Grading criteria. Each key's value is an object containing "score" (integer 1-10) and "reason" (string). No other information.

### Grading criteria:

1. Text Accuracy (1-10): Evaluate consistency with the user's original text (no missing/extra/wrong characters, no spelling/grammatical errors in Chinese/English). Score 10: 100% accurate; Score 1: massive errors or unrecognizable characters.

2. Text-Background Harmony (1-10): Evaluate visual coordination: (1) text avoids blocking the background's main subject (key figures, core graphics); (2) text color/transparency ensures clear contrast with the background (no blurring). Score 10: no blocking, perfect contrast; Score 1: complete blocking or unreadable due to poor contrast.

3. Text Presentation Quality (1-10): Evaluate text's own properties: (1) structural rationality (clear title/body hierarchy, compliance with reading habits, balanced spacing); (2) physical readability (appropriate font selection, suitable size, neat alignment). Score 10: clear structure, highly readable; Score 1: chaotic structure and physically unreadable.

4. Meaning Expression Adaptability (1-10): Evaluate meaning transmission: (1) key information is highlighted (via weight/color/size); (2) layout matches text's emotional tone (e.g., serious text uses rigorous typography); (3) text position aligns with the background's semantic context (e.g., "ocean protection" text near ocean elements). Score 10: amplifies meaning, matches tone, aligns with background semantics; Score 1: contradicts meaning/tone or conflicts with background semantics.

### The background image:

<image>

User's original specified text: {text\_content}

### The typeset result:

<image>