SEGA: A Stepwise Evolution Paradigm for Content-Aware Layout Generation with Design Prior

Supplementary Material

1. Overview

In this appendix, we provide additional details which were omitted in the main manuscript due to space constraints. First, we introduce more specifics for our presented GenPoster-100K dataset, including the data diversity and data structure. Then, we give more technical details of our proposed method. Finally, more experimental results are reported to perform a comprehensive evaluation of our method.

2. GenPoster-100K Dataset

Here, we introduce more details of our proposed GenPoster-100K dataset. Due to the limited coverage of existing datasets, which only include e-commerce scenarios and have relatively single layout patterns, methods based on large models can easily fit the layout patterns and achieve performance close to saturation, as shown in Table 1. Therefore, a dataset with more diverse scenes and more complex layouts is needed to expand the boundary of the models in designing layouts. The key features of the dataset are in the following.

2.1. Diversity

The dataset is composed of 105456 poster instances. The dataset includes a diverse range of poster categories such as commercial, event, and product posters to ensure that the dataset covers a broad spectrum of real-world applications. Some examples of our proposed dataset are displayed in Figure 1. he diversity of the dataset is reflected in the poster's resolution, element quantity, and element attributes as well. The poster resolution ranges from 640x480 to 4000x3000, with element quantities spanning from a few to hundreds, as shown in Figure 2. This ensures that the dataset covers a broad spectrum of real-world layouts. In terms of element attributes, we provide fine-grained properties such as text content, font, font size, case, letter spacing, line spacing, rotation angle, alignment, text box, opacity, and color. These rich attributes offer a solid foundation for models to understand poster design and layouts. In terms of text content, the dataset includes different kinds of English text, such as titles, subtitles, body text, telephone numbers, addresses, websites, etc, as shown in Figure 4. Long body text is often represented by non-meaningful text, allowing for a more nuanced understanding and arrangement of textual content.

2.2. Hierarchical Structure

In the dataset, the elements' hierarchical structure of each poster is preserved, as shown in Figure 3. Our dataset provides clean and artifact-free background images and rendered images of each text element, allowing for the flexible composition of design elements. One can decide which element should be put on or taken off from the canvas, which makes the design of various learning tasks possible. In poster design, a coherent layout can only be furnished upon comprehending the intertextual relationships and their respective roles within the context of the background imagery. The dataset also depicts the structure of the texts. For example, '`Back`', '`To`', and '`School`' as separate layers, which adds a layer of complexity to the design task by requiring the models to understand the inter-text relations.

2.3. Posters with Textual Content

A significant advantage of our dataset is that many of the PSD files are homogeneous, that is to say, multiple PSD files are from the same original PSD file. These sub-PSDs possess the same text but different background designs. The model needs to understand the same text, which allows for training models to be robust against variations in text arrangement. This feature helps to prevent overfitting by ensuring that the model learns to adapt to different contexts. Furthermore, these similar poster instances can be utilized to examine the robustness of retrieval-based methods as a potential usage.

3. More Details of Our Method

3.1. Dataset Prune

In section 3.3.1 of the main manuscript, we mention that we prune the layout GT by removing samples that are not aligned with our design principles. Specifically, we used unsupervised metrics to evaluate the quality of GT and excluded the lower-scoring 70% samples that violated design principles, totaling 4623.

3.2. Perturb Algorithm

In section 3.3.2 of the main manuscript, we mention that we perturb the layout GT to increase the diversity of RF module training data. We follow the symbol usage in the main manuscript and the detailed perturbing process is described in algorithm 1.

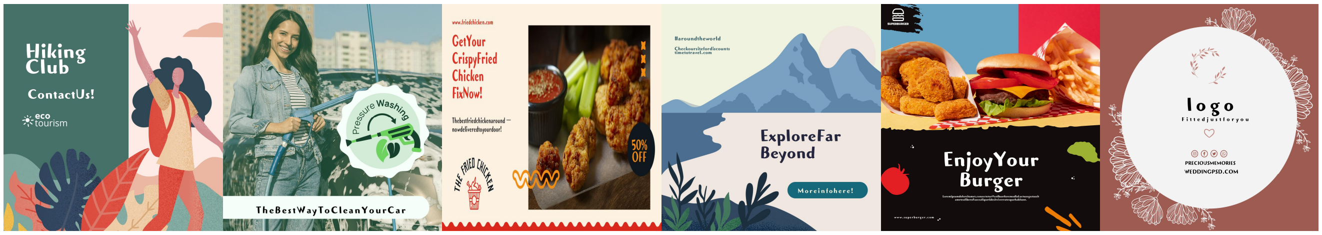


Figure 1. **Some exemplars from GenPoster-100K dataset.**

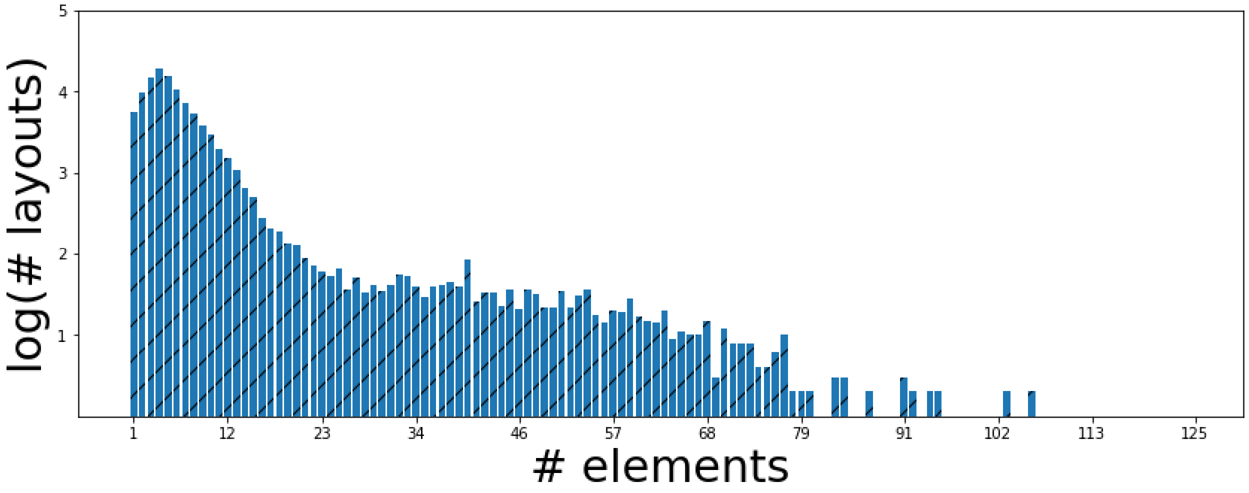
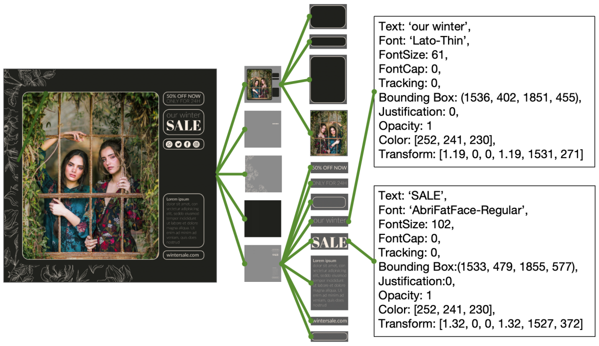


Figure 2. **Statistics on layout variety in GenPoster-100K dataset.**



Text: 'our winter',
Font: 'Lato-Thin',
FontSize: 61,
FontCap: 0,
Tracking: 0,
Bounding Box: (1536, 402, 1851, 455),
Justification: 0,
Opacity: 1
Color: [252, 241, 230],
Transform: [1.19, 0, 0, 1.19, 1531, 271]

Text: 'SALE',
Font: 'AbriFatFace-Regular',
FontSize: 102,
FontCap: 0,
Tracking: 0,
Bounding Box:(1533, 479, 1855, 577),
Justification:0,
Opacity: 1
Color: [252, 241, 230],
Transform: [1.32, 0, 0, 1.32, 1527, 372]

Figure 3. **The hierarchical structure of a poster instance.**

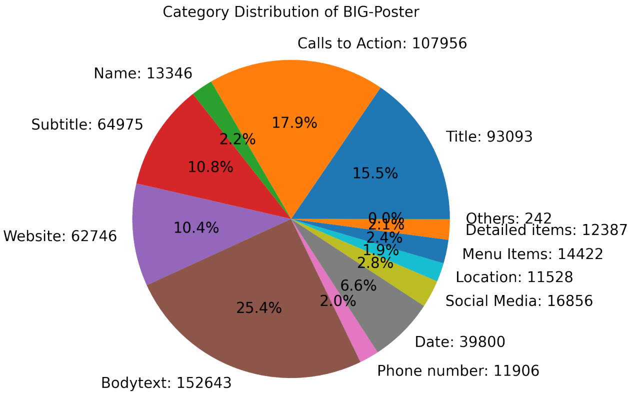


Figure 4. **Category distribution of GenPoster-100K's elements.**

4. Experiments

4.1. Implementation details

4.1.1. Training Setup

All our experiments are carried out with PyTorch with AdamW optimizer and 8 NVIDIA A100-40GB GPUs. We follow most of the training recipe of instruction tuning in LLaVA-1.5 [10] to set the hyper-parameters for experiments. The batch-size is set to 128. To preserve the learned knowledge of the pre-trained multimodal LLM, we leverage Low-rank adaptation (LoRA) [4] to perform efficient fine-tuning and completely freeze the vision backbone. To improve the training efficiency of LoRA, we empirically set the rank value to 128 and alpha value to 256. Moreover, the rope scaling [11] is adopted to enable the model to handle longer prompts for including the layout to be refined, and the scaling factor hyper-parameter is set to 2.

Algorithm 1 Pseudocode for perturbing layout

```
Input: Layout GT $L^{GT}$ , Threshold $\epsilon$
Output: Perturbed layout $L^{Perturb}$
$total\_num \leftarrow Len(L^{GT})$
$L^{Perturb} \leftarrow L^{GT}$
while $True$ do
    $idx \leftarrow Randomchoose(total\_num)$
    $e_{idx} \leftarrow L^{GT}[idx]$
    $e^*_{idx} \leftarrow Perturb(e_{idx})$
    $L^{Perturb}[idx] \leftarrow e^*_{idx}$
    $p\_cur \leftarrow Uniform(0, 1)$
    if $p\_cur < \epsilon$ then
        break
    end if
end while
return $L^{Perturb}$ =0
```

4.1.2. Testing Setup

To ensure the reliability of the results, we generate layouts on three independent trials and report the average of the metrics. With the temperature set to 0.2, we use random sampling for all the models.

4.1.3. Instruction Tuning Template

SEGA adopts two sequential stages: Coarse-level Estimation and Fine-level Refinement. Due to the different focuses of the two stages, we use different instruction tuning templates and show them in Figure 5. For the elements in the layout, we uniformly express them in the format of $\{$*category*, *bbox*$\}$ to facilitate data connection between the two stages.

4.2. Dataset

In this part, we introduce our utilized three datasets: PKU [3], CGL [16], and Crello [13]. PKU [3] and CGL [16] for content-aware layout generation, primarily composed of posters from e-commerce scenarios, such as cosmetics and clothing. Specifically, the PKU dataset includes three element categories: logo, text, and underlay, and contains 9,974 annotated posters, constituted by layouts and corresponding content images. While CGL additionally contains embellishment elements and comprises 60,548 annotated posters. For these two datasets, we follow RALF [2] to create dataset splits with a train/val/test ratio of roughly 8:1:1 in the annotated subset. Since the two above-mentioned datasets are mainly restricted by e-commerce themes, to testify the layout designing ability of the model in the more challenging and generalized scene, we employ the Crello [13] dataset and conduct experiments on it to further evaluate our method. It contains 23,182 posters collected from the web, preserving all the separated elements for each poster without layout element category annotation. To annotate the category for each layout element, we first use Yi-34B LLM [15] to classify all of the input text of textual elements into 13 predefined categories (i.e. 'Title', 'Subtitle', 'Bodytext', 'Date', 'Name', 'Website', 'Phone number', 'Detailed items', 'Calls to Action', 'Menu Items', 'Social Media', 'Location', 'Others'). Practically, we randomly choose 1000 layouts to check the predefined categories, finding it achieves 96% annotation accuracy. To further ensure the quality of category annotation, we correct the potential mistakes over the whole dataset by manually verifying. Then, we combine all non-text elements to create a background content image. Doing so allows us to organize the Crello dataset into a format similar to PKU and CGL, making it suitable for the content-aware layout generation task.

It is noteworthy that in the Crello dataset, some underlay elements are coupled with the background image. We get a preliminary draft of the underlay GT by detecting closed curves that cover text elements and obtain the final underlay GT through manual inspection. Then, we render all non-text layers onto the background image, construct a canvas, and plan all text elements on it, which means the model needs to detect the underlay in the canvas and place the element in it. This approach yields two distinct benefits. Firstly, unlike the PKU and CGL datasets, which inpaints the original poster to form a canvas, our generated canvas does not contain any artifacts. Secondly, since the underlay is no longer a discrete element requiring planning but is already integrated into the canvas, it necessitates that the model undertakes an analysis of the arrangement of elements on the canvas when planning text elements. This intricate analysis aligns with the expectations of the content-aware layout generation task. Lastly, we also correct the potential failed annotation via manual adjustment over the dataset.

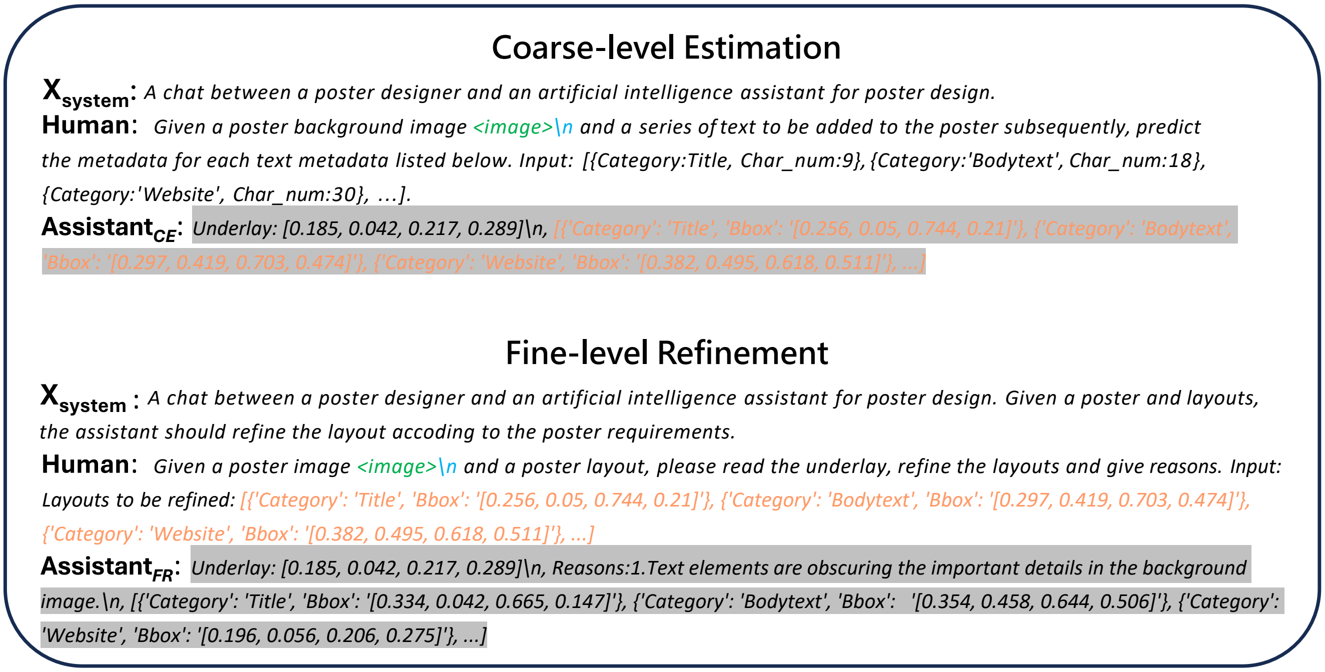
4.3. Evaluation Metrics

Following previous works [3, 8, 12], we use three sets of metrics to evaluate our method: *Graphic metrics*, *Content metrics*, and *Aesthetic scores*. *Graphic metrics* assess the coherence among predicted elements, including Alignment (*Ali* $\downarrow$), Overlay (*Ove* $\downarrow$), Underlay effectiveness loose (*Und_l* $\uparrow$), and Underlay effectiveness strict (*Und_s* $\uparrow$).

Alignment (*Ali* $\downarrow$) is the extent of spatial non-alignment between elements.

Overlay (*Ove* $\downarrow$) represents the average Intersection over the union of all element pairs, excluding underlay elements.

Underlay effectiveness (*Und* $\uparrow$) calculates the proportion of valid underlay elements to the total underlay



Coarse-level Estimation

$\mathbf{X}_{\mathbf{system}}$: *A chat between a poster designer and an artificial intelligence assistant for poster design.*

Human: *Given a poster background image <image>\n and a series of text to be added to the poster subsequently, predict the metadata for each text metadata listed below. Input: [{Category:Title, Char_num:9}, {Category:'Bodytext', Char_num:18}, {Category:'Website', Char_num:30}, ...].*

Assistant$_{CE}$: *Underlay: [0.185, 0.042, 0.217, 0.289]\n, [{'Category': 'Title', 'Bbox': '[0.256, 0.05, 0.744, 0.21]'}, {'Category': 'Bodytext', 'Bbox': '[0.297, 0.419, 0.703, 0.474]'}, {'Category': 'Website', 'Bbox': '[0.382, 0.495, 0.618, 0.511]'}, ...]*

Fine-level Refinement

$\mathbf{X}_{\mathbf{system}}$: *A chat between a poster designer and an artificial intelligence assistant for poster design. Given a poster and layouts, the assistant should refine the layout accoding to the poster requirements.*

Human: *Given a poster image <image>\n and a poster layout, please read the underlay, refine the layouts and give reasons. Input: Layouts to be refined: [{'Category': 'Title', 'Bbox': '[0.256, 0.05, 0.744, 0.21]'}, {'Category': 'Bodytext', 'Bbox': '[0.297, 0.419, 0.703, 0.474]'}, {'Category': 'Website', 'Bbox': '[0.382, 0.495, 0.618, 0.511]'}, ...]*

Assistant$_{FR}$: *Underlay: [0.185, 0.042, 0.217, 0.289]\n, Reasons:1.Text elements are obscuring the important details in the background image.\n, [{'Category': 'Title', 'Bbox': '[0.334, 0.042, 0.665, 0.147]'}, {'Category': 'Bodytext', 'Bbox': '[0.354, 0.458, 0.644, 0.506]'}, {'Category': 'Website', 'Bbox': '[0.196, 0.056, 0.206, 0.275]'}, ...]*

Figure 5. **An example of our instruction tuning templates.** As shown in the orange part of the figure, the output of the model in the Coarse-level Estimation stage serves as the input of the model in the Fine-level Refinement stage. The input image is marked as <image>. The parts marked by gray background are used to compute the loss in the auto-regressive model.

elements. An underlay is regarded as valid and scores 1 if it entirely covers a non-underlay element; otherwise, it scores 0 in strict standard, subscribed $Und\_s \uparrow$. In contrast, $Und\_l \uparrow$ calculates the area ratio of the non-underlay element covered by the underlay. It is worth noting that since the underlay in Crello dataset has already been rendered on the canvas. All text elements and the GT of the underlay elements are used to calculate this metric.

Content metrics evaluate the harmony between the predicted elements and the background image, including Readability score ($Rea \downarrow$) and Occlusion (Occ ↓).

Readability score ($Rea \downarrow$) evaluates the non-flatness of text elements by calculating gradients in the image space along both vertical and horizontal axes within these elements.

Occlusion ($Occ \downarrow$) computes the average saliency value in the overlapping region between the saliency map S and the layout elements.

Due to the overlap between our design priors and unsupervised evaluation metrics, to fairly evaluate the effectiveness of our method, we also referred to COLE [7] to leverage GPT for quality evaluation from a more comprehensive perspective. Specifically, by utilizing GPT-4V [1], we conducted a comprehensive evaluation named *aesthetic scores* that measure the overall aesthetic quality and harmony of the elements in the graphic composition from four independent perspectives and reported the final average score.

$S_{DL} \uparrow$ means the graphic design should present a clean, balanced, and consistent layout. The organization of elements should enhance the message, with clear paths for the eye to follow.

$S_{GI} \uparrow$ reflects that any graphics or images used should enhance the design rather than distract from it. They should be high quality, relevant, and harmonious with other elements.

$S_{IO} \uparrow$ evaluates the innovation level of the design.

$S_{TV} \uparrow$ represents text readability. A lower score is assigned if the readability of the text is poor due to the color of the text being similar to the background color or overlapping of the text.

4.4. Methods in Experiments

The main comparison methods in our experiments are introduced here:

CGL-GAN [16] is a non-autoregressive encoder–decoder model employing a Transformer architecture.

FlexDM [6] can take all the rendered elements as inputs into a transformer. We assemble all graphical elements into a background, then feed it along with all the rendered text to FlexDM for fair comparison.

RALF [2] is a method for layout generation that uses a transformer as its backbone and leverages advanced retrieval enhancement techniques, which is currently the top-performing method among non-large model techniques in terms of performance.

PosterLlama [12] is a layout generation network that reforms layout elements into HTML code and leverages the rich design knowledge embedded within language models.

SEGA w/o FR refers to the model only adopts the coarse-level estimation module fine-tuned on the target dataset using the open-source multi-modal large-scale model, LLaVA-1.5 [9]. It is used as the comparative baseline for our method.

SEGA w/o FR (Ens-k) uses the same model with SEGA w/o FR module, but infer k times and adopt the best results as the final output. It is worth noting that we use the method of averaging the results of unsupervised indicators to select the best inference result.

SEGA is our full model, employs a stepwise evolution Paradigm for Content-Aware Layout Generation. It first uses the Coarse-level Estimation module (CE module) to roughly estimate the intermediate layouts and utilizes the Fine-level Refinement module (FR module) to iteratively refine the intermediate layouts.

4.5. More Quantitative Results

Content-aware layout generation has two main subtasks: constrained layout generation and unconstrained layout generation. Their difference lies in whether generation constraints exist, such as having a title and two text elements. In the main text, we focus on the more challenging constrained layout generation and only post the unconstrained experiment in the Crello dataset. Here, we conduct more unconstrained layout generation experiments in the PKU and CGL datasets. As shown in Table 1, our method SEGA achieves the best results in most indicators, except the occlusion and unreadability where the LLM-based methods are not good due to the limited vision encoder. We still achieve the sub-optimal results in the comparison methods.

Moreover, we add quantitative results for different inference rounds that are not presented in the main manuscript. As shown in Table 3, when the number of iteration rounds increases, the quality of the layout gradually improves and reaches a maximum.

4.6. Impact of Different Configurations

In this section, we introduce more experimental results that are not contained in the main manuscript, including the role of each design principles, the impact of the Data pruning, the impact of different initialization of FR modules, different ways of layout rendering in the canvases and the detail of the perturbed layouts for FR module training.

4.6.1. Design Principles

To more effectively illustrate the significance of the design principles we have summarized, we eliminate all design principles in SEGA, thereby obtaining config 1 as presented in Table 2. Subsequently, we incrementally incorporate the design principles into the model. It is evident that upon the integration of the corresponding design principles into our method, the associated performance is enhanced, thereby validating the successful utilization of each layout design principle by SEGA.

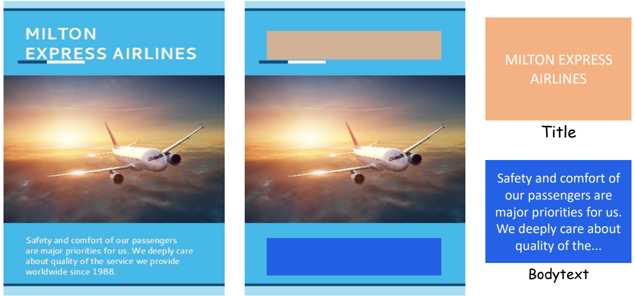

MILTON
EXPRESS AIRLINES
Safety and comfort of our passengers are major priorities for us. We deeply care about quality of the service we provide worldwide since 1988.
MILTON EXPRESS AIRLINES
Title
Safety and comfort of our passengers are major priorities for us. We deeply care about quality of the...
Bodytext

Figure 6. **Example for different rendering strategies.**

4.6.2. Data Prune

From the results of the ablation experiment, it can be seen that our data-pruning strategy has greatly improved the performance of the model, but we must emphasize that this only holds for the training of the refine model. As shown in Table 4, the performance of the baseline model trained on the same pruned dataset decreased. We believe this is because the refinement stage has higher requirements for data quality compared to planning, and planning places more emphasis on diversity.

4.6.3. Initialization of FR module

We evaluate the impact of different initialization manner of the FR module, including those from LLaVA-1.5 and the CE module. As illustrated in Table 5, the FR module, when initialized from the CE module, yields superior performance, due to its richer layout design knowledge.

4.6.4. Element Rendering Strategy

Inspired by the previous work about visual prompt [14], we try to figure out the best render method to put the element layers into the canvas. We try two different strategies for layer rendering: 1) the layout with category-wise color blocks, and 2) the original layer image, illustrated in Figure 6. As shown in Table 6, rendering layers with corresponding color blocks is better than rendering the corresponding layer image of the poster. Thereby, we adopt rendering color blocks as the rendering method of SEGA.

4.7. Impact of Pre-training on the GenPoster-100K Dataset

In order to investigate the effectiveness of GenPoster-100K as a pre-training dataset, we also conduct transfer training experiments on the PKU and CGL dataset, using the pre-trained model weights on GenPoster-100K as initialization

Method	PKU						CGL					
	Graphic				Content		Graphic				Content	
	Ali ↓	*Ove* ↓	*Und_l* ↑	*Und_s* ↑	*Read* ↓	*Occ* ↓	Ali ↓	*Ove* ↓	*Und_l* ↑	*Und_s* ↑	*Read* ↓	*Occ* ↓
Non-LLM Based												
CGL-GAN [17] (IJCAI, 2022)	-	0.0380	-	0.4800	0.0158	0.1320	-	0.0470	-	0.6500	0.0213	0.1400
LayoutDM [5] (CVPR, 2023)	-	0.1720	-	0.4600	0.0201	0.1520	-	0.0260	-	0.7900	0.0192	0.1270
RALF [2] (CVPR, 2024)	0.0028	0.0083	0.9808	0.9201	**0.0128**	0.1195	0.0023	0.0041	0.9912	0.9756	**0.0179**	**0.1246**
LLM Based												
PosterLlama [12] (ECCV, 2024)	0.0015	0.0030	0.9998	0.9910	0.0188	0.2087	**0.0005**	0.0007	**0.9990**	0.9909	0.0302	0.2471
SEGA *w/o* FR module	0.0021	0.0004	0.9908	0.9863	0.0132	0.1162	0.0019	**0.0002**	0.9988	0.9875	0.0292	0.2406
SEGA	**0.0014**	**0.0003**	**1.0000**	**0.9986**	0.0131	**0.1160**	0.0017	0.0004	0.9936	**0.9917**	0.0285	0.2367

Table 1. **Comparisons results on the PKU and CGL dataset under unconstrained generation setting.** For better display, we **bold** the best values and underline the second values. Our SEGA achieves the best results except for the content metric that the LLM-based method is not good at, and also achieves better performance than PotserLlama in terms of those two metrics.

Config	Design Principles				Graphic				Content	
	Alignment	Occulusion	Underlay	Overlay	Ali ↓	*Ove* ↓	*Und_l* ↑	*Und_s* ↑	*Read* ↓	*Occ* ↓
1					0.0098	0.0092	0.9567	0.9157	0.0252	0.3947
2	✓				**0.0086**	0.0095	0.9537	0.9200	**0.0250**	0.3942
3	✓	✓			0.0087	0.0093	0.9523	0.9094	0.0259	0.3910
4	✓	✓	✓		0.0089	0.0089	**0.9593**	0.9247	0.0257	0.3917
5	✓	✓	✓	✓	0.0095	**0.0025**	0.9541	**0.9270**	0.0260	**0.3907**

Table 2. **The impact of different design principles.** We remove all design principles integrated into the SEGA framework and inject design principles one by one to analyze their impact on model performance.

Method	Graphic				Content	
	Ali ↓	*Ove* ↓	*Und_l* ↑	*Und_s* ↑	*Read* ↓	*Occ* ↓
SEGA *w/o* FR	0.0102	0.0093	0.8485	0.7315	0.0271	0.3948
SEGA $_{T=1}$	0.0096	0.0025	0.9553	0.9235	0.0265	0.3919
SEGA $_{T=2}$	**0.0095**	0.0024	0.9541	0.9270	0.0260	0.3907
SEGA $_{T=3}$	0.0096	**0.0023**	0.9603	0.9358	**0.0257**	0.3913
SEGA $_{T=4}$	0.0094	0.0025	**0.9630**	**0.9375**	**0.0257**	**0.3902**

Table 3. **The impact of refinement iteration rounds under constrained generation setting.**

Setting	Graphic				Content	
	Ali ↓	*Ove* ↓	*Und_l* ↑	*Und_s* ↑	*Read* ↓	*Occ* ↓
Origin Data	0.0102	**0.0093**	**0.8485**	**0.7315**	**0.0271**	**0.3948**
Pruned Data	**0.0076**	0.0107	0.7071	0.5207	0.0414	0.4329

Table 4. **The Impact of Data Prune on SEGA w/o FR module.** Data prune is not as effective on SEGA w/o FR module as it is on the refine model.

Setting	Graphic				Content	
	Ali ↓	*Ove* ↓	*Und_l* ↑	*Und_s* ↑	*Read* ↓	*Occ* ↓
Initialized from LLaVA	0.0097	0.0031	**0.9610**	0.9188	0.0285	0.3921
Initialized from CE	**0.0095**	**0.0025**	0.9541	**0.9270**	**0.0260**	**0.3907**

Table 5. **The impact of different initialization strategy of the FR module.**

Setting	Graphic				Content	
	Ali ↓	*Ove* ↓	*Und_l* ↑	*Und_s* ↑	*Read* ↓	*Occ* ↓
Render Layer Image	**0.0092**	0.0043	0.9307	0.9246	**0.0255**	0.3943
Render Color Block	0.0095	**0.0025**	**0.9541**	**0.9270**	0.0260	**0.3907**

Table 6. **The impact of different layout rendering modes to be refined onto the background image.**

and then conducting SEGA training. To further validate its effectiveness, we compare it with the SEGA 13B. As shown in Table 7, even with our best approach, pre-training can further improve the performance of the model.

4.8. More Qualitative Results

In the main text, due to limited space, we only present some layout results. Here, we display more comparative visualization results, as shown in Figure 7, and Figure 8.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 4

[2] Daichi Horita, Naoto Inoue, Kotaro Kikuchi, Kota Yamaguchi, and Kiyoharu Aizawa. Retrieval-augmented layout transformer for content-aware layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 67–76, 2024. 3, 4, 6

[3] Hsiao Yuan Hsu, Xiangteng He, Yuxin Peng, Hao Kong, and Qing Zhang. Posterlayout: A new benchmark and approach

Method	PKU						CGL					
	Graphic				Content		Graphic				Content	
	Ali ↓	*Ove* ↓	*Und_l* ↑	*Und_s* ↑	*Read* ↓	*Occ* ↓	Ali ↓	*Ove* ↓	*Und_l* ↑	*Und_s* ↑	*Read* ↓	*Occ* ↓
SEGA	0.0034	0.0029	0.9902	0.9823	0.0138	0.1215	0.0019	0.0009	**0.9947**	0.9892	0.0291	0.2401
SEGA *	0.0034	**0.0022**	**0.9904**	**0.9857**	0.0138	**0.1210**	0.0019	0.0009	0.9944	**0.9902**	0.0291	**0.2397**
GT	0.0036	0.0009	0.9950	0.9444	0.0119	0.1185	0.0023	0.0003	0.9937	0.9402	0.0296	0.2390

Table 7. **The impact of the GenPoster-100K pre-training on the PUK and CGL Dataset.** For better demonstration, we **bold** the best values except for the tied first place.

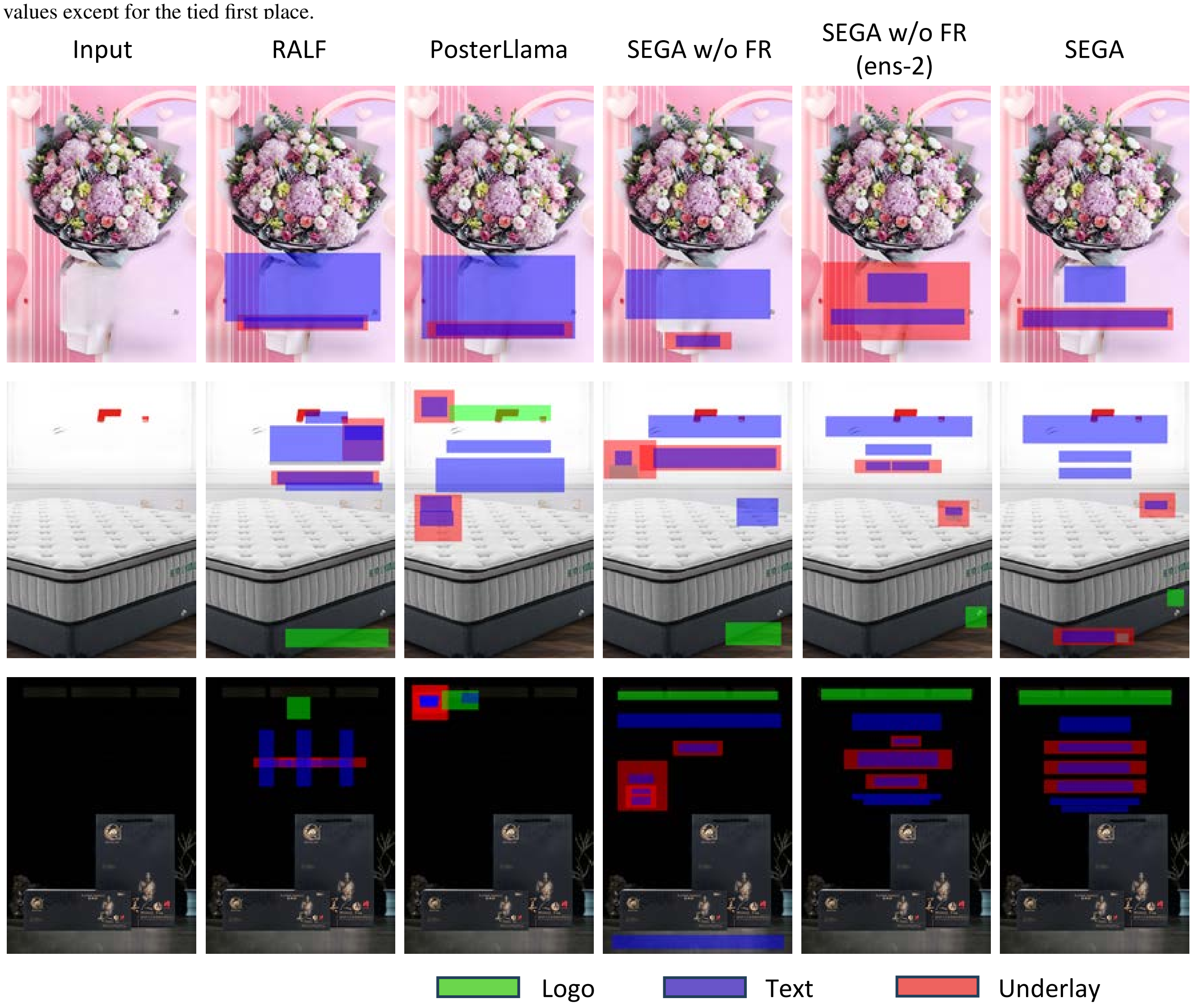


Figure 7. **Qualitative results in the CGL dataset.**

for content-aware visual-textual presentation layout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6018–6026, 2023. 3

[4] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 2

[5] Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Layoutdm: Discrete diffusion model for controllable layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10167–10176, 2023. 6

[6] Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Towards flexible multi-modal document models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages

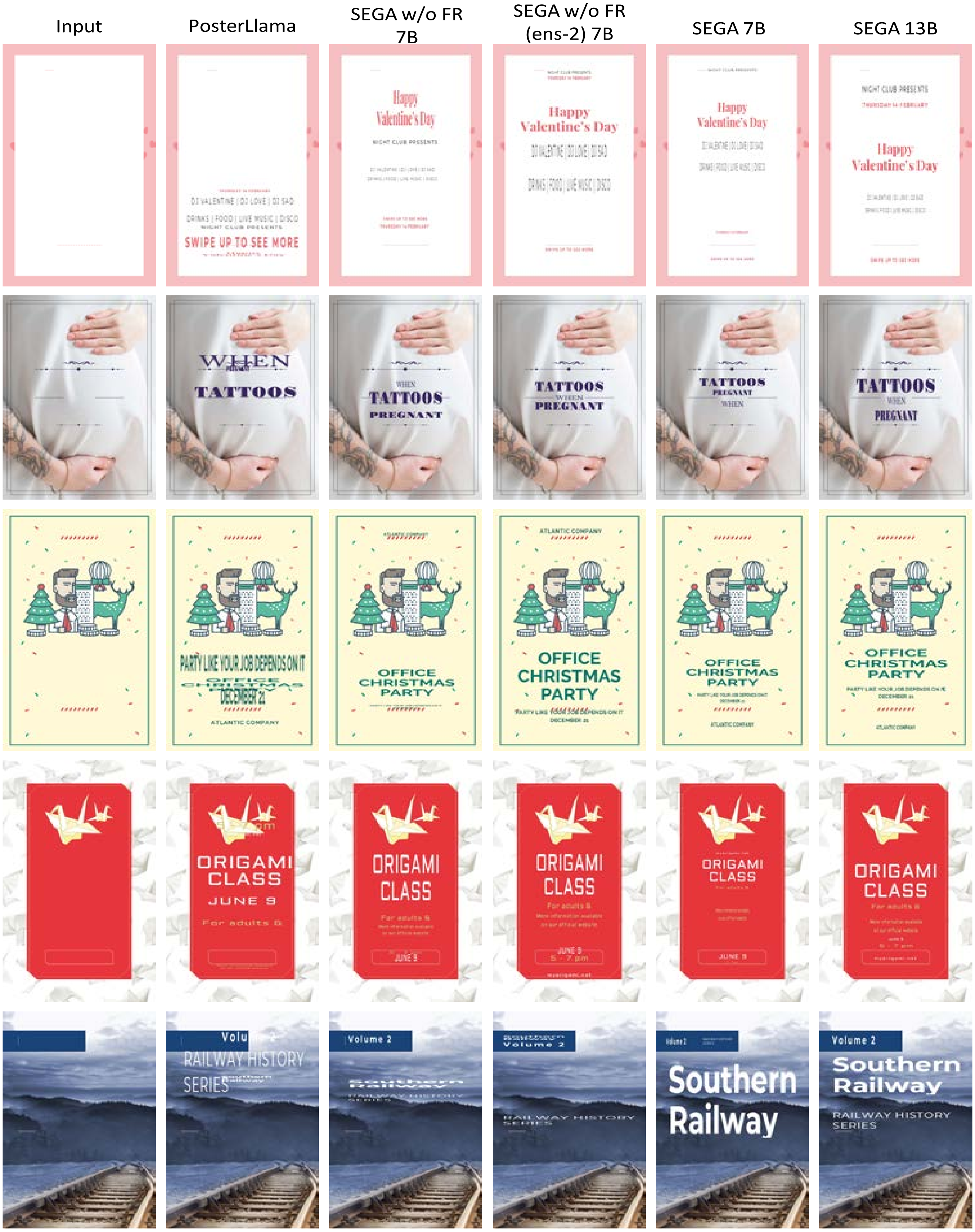


Figure 8. **Qualitative results in the Crello dataset.**

14287–14296, 2023. 4

[7] Peidong Jia, Chenxuan Li, Zeyu Liu, Yichao Shen, Xingru Chen, Yuhui Yuan, Yinglin Zheng, Dong Chen, Ji Li, Xiaodong Xie, et al. Cole: A hierarchical generation framework for graphic design. *arXiv preprint arXiv:2311.16974*, 2023. 4

[8] Jianan Li, Jimei Yang, Jianming Zhang, Chang Liu, Christina Wang, and Tingfa Xu. Attribute-conditioned layout gan for automatic graphic design. *IEEE Transactions on Visualization and Computer Graphics*, 27(10):4039–4048, 2020. 3

[9] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 5

[10] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024. 2

[11] Xiaoran Liu, Hang Yan, Shuo Zhang, Chenxin An, Xipeng Qiu, and Dahua Lin. Scaling laws of rope-based extrapolation. *arXiv preprint arXiv:2310.05209*, 2023. 2

[12] Jaejung Seol, Seojun Kim, and Jaejun Yoo. Posterllama: Bridging design ability of langauge model to contents-aware layout generation. *arXiv preprint arXiv:2404.00995*, 2024. 3, 5, 6

[13] Kota Yamaguchi. Canvasvae: Learning to generate vector graphic documents. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5481–5489, 2021. 3

[14] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023. 5

[15] Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*, 2024. 3

[16] Min Zhou, Chenchen Xu, Ye Ma, Tiezheng Ge, Yuning Jiang, and Weiwei Xu. Composition-aware graphic layout gan for visual-textual presentation designs. *arXiv preprint arXiv:2205.00303*, 2022. 3, 4

[17] Min Zhou, Chenchen Xu, Ye Ma, Tiezheng Ge, Yuning Jiang, and Weiwei Xu. Composition-aware graphic layout gan for visual-textual presentation designs. *arXiv preprint arXiv:2205.00303*, 2022. 6