InnoAds-Composer: Efficient Condition Composition for E-Commerce Poster Generation

Supplementary Material

1. Related Work

1.1. Poster Generation

Poster generation aims to automatically produce visually appealing layouts that integrate images, text, and design elements to effectively convey information and aesthetic appeal. Recent advances such as COLE [24], Posta [9] and PosterCraft [14] leverage MLLMs to enable multi-stage control and iterative optimization, generating posters with high artistic quality and visual coherence. However, these methods are primarily designed for general or artistic compositions and are less suitable for visually appealing promotional images that must effectively present product information and attract consumer attention in e-commerce scenarios.

To address the specific requirements of e-commerce poster generation, several tailored approaches [18, 19, 81] have been proposed. DreamPainter [81] and Repainter [19] introduce inpainting-based frameworks to customize both product and background regions, enabling controlled and coherent visual synthesis. PosterMaker [18] further extends this line of work by combining prompt, subject, and text conditions to achieve fine-grained customization of background, product, and textual elements. Nonetheless, its reliance on prompt-based background generation often leads to results that deviate from desired visual or semantic constraints.

1.2. Text Rendering

Text rendering aims to generate visually coherent and legible text within images, often requiring fine-grained control over font, layout, and contextual consistency. Early text rendering methods [10, 12, 70, 84] primarily focused on generating Latin characters such as English text, but struggled to generalize to non-Latin scripts like Chinese due to the lack of corresponding text representations. To address this limitation, subsequent approaches introduced glyph-based representations [25, 33, 34, 39, 53, 54] to bridge the gap between different languages. For instance, AnyText [53] integrates glyph images as conditional inputs through a ControlNet [73] structure, enabling controllable rendering of multilingual text, while Glyph-ByT5 [33] employs a customized multilingual text encoder trained on glyph representations to generate non-Latin characters effectively.

More recent studies [3, 20, 29, 30, 35, 36, 55, 67] have adopted Diffusion Transformer (DiT) [41] architectures to achieve higher-quality and more contextually consistent text generation. TextFlux [67], for example, uses Flux-Fill [28] as its backbone and leverages in-context learning to better capture glyph structure and spatial dependencies. Building on this line of work, FluxText [29] further explores multiple condition fusion strategies to enhance the fidelity and controllability of generated text. Inspired by these advances, we adopt a DiT-based backbone in our framework to improve the quality, clarity, and contextual alignment of text generation in e-commerce poster synthesis.

1.3. Multi-Condition Control Generation

Controllable image generation aims to incorporate multiple conditioning signals—such as text, layout, or structural guidance—into the generative process to achieve fine-grained control over visual content. Earlier approaches typically relied on ControlNet [73] or IP-Adapter [71] architectures to inject additional conditions through feature modulation or adapter networks. More recently, DiT-based methods [51, 62, 66, 74] have demonstrated strong potential for multi-condition control by integrating conditioning tokens directly into the denoising process. For example, OmniControl [51] and UNO [66] concatenate textual or semantic tokens with noisy image tokens to achieve unified conditional generation, while IC-Edit [78] and insertanything [49] performs spatial concatenation and leverages in-context learning to support diverse conditional editing tasks.

However, as the number of conditions increases, the corresponding growth in token count leads to higher attention computation costs and reduced efficiency. To mitigate this issue, several studies [1, 22, 31, 40, 42, 52, 59, 60, 76] have explored more efficient conditioning mechanisms. OmniControl2 [52], for instance, computes condition token features only once and reuses them across denoising steps, while FullDiT2 [22] introduces a dynamic token selection mechanism to adaptively identify and retain the most informative context tokens during generation. Although these methods have achieved promising results in general visual synthesis tasks, applying multi-condition control to e-commerce poster generation remains challenging, as it requires simultaneously maintaining background style consistency, accurate text rendering, and product integrity while ensuring high-quality and efficient image generation.

2. Implementation Details

InnoAds-Composer is developed based on the FLUX model [28], which is pretrained on a large-scale text rendering dataset AutoPP1M [15] and possesses an in-

Figure 8. Additional qualitative results generated by our method.

trinsic awareness of Chinese characters. The model is further optimized under tri-conditional control using our InnoComposer-80K dataset through a two-stage training strategy. (1) In *Stage I*, we fine-tune all MM-DiT blocks using LoRA modules with a rank of 256. A constant learning rate of 2×10^{-5} is adopted, and the training process requires approximately 1.1k GPU hours. (2) In *Stage II*, We removed selected tokens and fine-tuned the network to minimize performance degradation. During this stage, the learning rate was set to 1×10^{-6}, and training was conducted for approximately 100 GPU hours based on the checkpoint from *Stage I*. All training processes were conducted at a resolution of 800 x 800, using Ascend 910B. During the inference phase, to ensure a fair comparison with open-source models, we used the A100 for evaluating performance and inference latency. Besides, we have supplemented the pseudocode for TFEM as follows:

Algorithm: Text Feature Enhancement Module (TFEM)

Input: Glyph image I_g, Single-glyph crops $\{C_i\}_{i=1}^{N}$
Output: Enhanced glyph tokens h^c

$h^{c1} \leftarrow \text{Patchify}(\text{VAE\_Encode}(I_g))$ // Global structure branch
for each crop C_i **in** $\{C_i\}_{i=1}^{N}$ **do**
  $f_i \leftarrow \text{OCR\_Backbone}(C_i)$
  $p_i \leftarrow \text{Add\_Positional\_Encodings}(f_i, \text{abs\_pos}, \text{font\_size}, \text{local\_pos})$
end for
$h^{c2} \leftarrow \text{Concat}(\{p_i\}_{i=1}^{N})$ // Local semantic branch
// Character Encoder Fusion via Cross-Attention
$h^c \leftarrow \text{Softmax}\left(\frac{(h^{c1}W_Q)(h^{c2}W_K)^{\top}}{\sqrt{d}}\right)(h^{c2}W_V) + h^{c1}$
return $\text{LayerNorm}(\text{FFN}(h^c))$

3. More Experiments

More Cases. Fig. 8 presents additional generation results produced by our method. As shown, our approach not only maintains high-fidelity subject consistency for various products, but also delivers accurate and visually coherent text rendering. Moreover, the method produces realistic and diverse background styles, demonstrating its strong ability to integrate multiple conditions into cohesive, high-quality product posters.

Table 3. Comparison of Stage II performance under different token pruning ratios. In the pruning ratio column, [x, y, z] denote the token pruning ratios for the glyph, subject, and style conditions, respectively.

Pruning Ratio (%)	**NED** ↑	**MSE** ↓	**CLIP-I** ↑
[0,0,0]	0.976	0.056	0.582
[50,20,30]	0.971	0.057	0.581
[70,40,50]	0.970	0.057	0.562
[80,50,60] (ours)	0.969	0.058	0.594
[90,60,70]	0.582	0.066	0.451

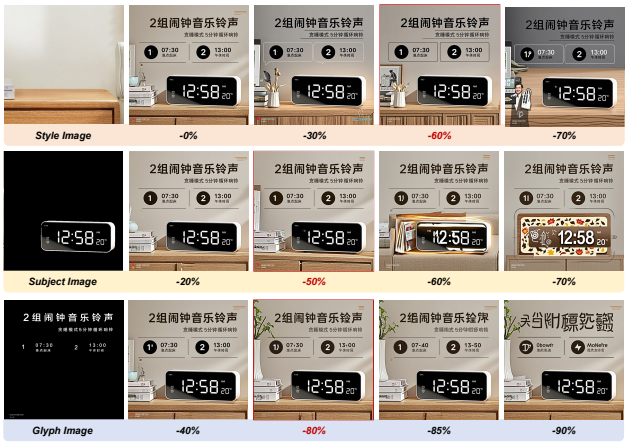

Figure 9. Qualitative results under different condition token pruning ratios.

Different Token Pruning Ratios. To validate the effectiveness of our importance-based token pruning ratios, we first evaluate alternative pruning proportions during *Stage I* inference, with qualitative results shown in Fig. 9. As illustrated, removing fewer tokens than the selected ratio preserves high-quality backgrounds, text rendering, and subject fidelity, while more aggressive pruning leads to a clear decline in generation quality. We further conduct *Stage II* training under these alternative ratios, with quantitative results summarized in Table 3. The table shows that pruning fewer tokens produces performance comparable to our chosen ratio, whereas pruning beyond it results in noticeable degradation across all metrics.