UniLayDiff: A Unified Diffusion Transformer for Content-Aware Layout Generation

Supplementary Material

A. Datasets

In this section, we provide more details for PKU and CGL datasets.

A.1 Dataset Details

The PKU dataset includes three element categories (logo, text, and underlay), while the CGL dataset also contains an additional "embellishment" category. In terms of scale, CGL is larger, with 60,548 annotated posters (each with both a layout and a corresponding image) and 1,000 unannotated canvases (images only). PKU contains 9,974 annotated posters and 905 unannotated canvases.

A.2 Data Splits

Both the PKU and CGL datasets are divided into an annotated split and an unannotated split. To ensure a fair evaluation, we follow the data splitting protocol of RALF [2] and LayoutDiT [5]. The training, validation, and test sets are all partitioned from the annotated split; for the PKU dataset, this comprises 7734, 1000, and 1000 samples, while for the CGL dataset, it comprises 48544, 6002, and 6002 samples for training, validation, and testing, respectively. The unannotated split serves exclusively as an unannotated test set. Consequently, due to the lack of ground-truth layout annotations, this unannotated test set cannot be used for calculating metrics like FID and is only suitable for evaluating unconditional generation tasks.

A.3 Data Preprocessing

The original PKU and CGL annotated datasets lack high-quality inpainted images and saliency maps. Therefore, subsequent works [2, 5] commonly redo data preprocessing. The standard procedure involves using advanced inpainting algorithms to remove original graphic elements from the posters to create clean canvases [2, 5]. For saliency detection, two methods are typically employed: ISNet [7] and BASNet [6]. This results in two types of saliency maps: a more precise map that accurately captures object contours (from ISNet) and a slightly blurrier map that covers a broader salient area (from BASNet). The final saliency map $\mathbf{S}$ fed into the model is generated by taking the element-wise maximum of these two maps.

We would like to extend our gratitude to the authors of LayoutDiT [5] for making their preprocessed data publicly available. In our work, we utilize their provided inpainted canvases, saliency maps, and saliency bounding boxes.

Module	Params
Image encoder (ViT)	20.16 M
Layout-mask MM-DiT	14.34 M
Layout-image MM-DiT	24.43 M
LoRA	0.13 M
Utils	0.25 M
Total	59.31 M

Table A. The number of parameters of each module.

B. Implementation Details

B.1 Architecture Details

MM-DiT. Our MM-DiT architecture consists of two parallel branches, each with a depth of 12. These branches are responsible for modeling layout-mask and layout-image interactions, respectively. For the layout-mask MM-DiT, the input dimensions are $d_{\text{layout}} = 128$ and $d_{\text{mask}} = 128$. For the layout-image MM-DiT, the input dimensions are $d_{\text{image}} = 256$ and $d_{\text{layout}} = 128$. In each branch, the inputs are projected to a shared dimension of $d = 512$ before the MM-Attention operation. The feed-forward network (FFN) dimension within each MM-DiT block is 1024.

Image encoder. We use a ViT-based image encoder with an embedding dimension of 256, a depth of 12, and an FFN dimension of 2048. It takes an input image of size 384×256 and uses a patch size of 32.

LoRA. We apply LoRA for fine-tuning, targeting only the layout-mask branch of our model. The adaptation is applied specifically to the linear projection matrices of the MM-Attention module within this branch, using a LoRA rank of $r = 4$ and a scaling factor of $\alpha = 3$.

Parameters summary. The parameters for each key module of UniLayDiff are summarized in Tab. A. The "Utility Modules" category includes the Layout Encoder, Mask Encoder, and the final output MLP.

B.2 Constraints Details

In this section, we present the details of the layout constraints, including the layout optimization loss and the weights of the losses.

Layout optimization loss. To encourage the generation of layouts that adhere to established design paradigms, we incorporate a layout optimization loss, denoted as $\mathcal{L}_{\text{lyt}}$. This objective aligns with the aesthetic constraints proposed in LACE [1], as both prioritize the rational internal arrange-

ment of layout elements. Specifically, tailored for the unique underlay elements in content-aware layout generation, $\mathcal{L}_{\text{lyt}}$ explicitly models the spatial constraints between underlays and other components. This encourages underlay elements to properly encompass their associated non-underlay counterparts.

For each underlay bounding box $\mathbf{b}_i$, we calculate its maximum coverage ratio, $C_{\max}(\mathbf{b}_i)$, over all non-underlay elements.

$$C_{\max}(\mathbf{b}_i) = \max_{j \in \mathcal{O}} \frac{\text{Area}(\mathbf{b}_i \cap \mathbf{b}_j)}{\text{Area}(\mathbf{b}_j)}, \tag{1}$$

where $C_{\max}(\mathbf{b}_i)$ is the maximum coverage ratio for a given underlay box $\mathbf{b}_i$; $\mathcal{O}$ is the set of all non-underlay elements; and $\mathbf{b}_j$ is an element from that set. All bounding boxes and their classifications (e.g., as underlay or non-underlay) are extracted from the predicted clean layout $\tilde{\mathbf{x}}_0$.

The final loss then penalizes incomplete coverage by averaging the coverage shortfall across all underlay elements.

$$\mathcal{L}_{\text{lyt}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left(1 - C_{\max}(\mathbf{b}_i)\right), \tag{2}$$

where $\mathcal{L}_{\text{lyt}}$ is the final loss value, and $\mathcal{B}$ is the set of all underlay elements, with $|\mathcal{B}|$ being its cardinality.

Hyperparameters for additional constraints. For the total objective function $\mathcal{L}_{\text{LoRA}}$ used in the LoRA fine-tuning stage, as formulated in Eq. 11 of the main text, we configure the weighting coefficients as follows: $\lambda_1 = 0.2$ for the relational constraint term $\mathcal{L}_{\text{rel}}$, $\lambda_2 = 0.4$ for the content constraint term $\mathcal{L}_{\text{ctn}}$, and $\lambda_3 = 1$ for the layout optimization term $\mathcal{L}_{\text{lyt}}$. The internal hyperparameters for these loss functions are listed in Tab. B. Moreover, the additional constraints are applied only during the first 30% of the DDPM steps (the low-noise stage), while for the remaining 70% of the steps, only $\mathcal{L}_{\text{diff}}$ is applied.

B.3 Training Details

We summarize the hyperparameters for both pre-training and fine-tuning UniLayDiff in Table C. Moreover, the sampling proportions of different tasks for the pre-training and fine-tuning stages are shown in Table D. For optimization, we employ the Adam optimizer with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The diffusion model (DDPM) is trained for a total of 1000 steps. For the final layout generation during inference, we use Denoising Diffusion Implicit Models (DDIM) sampling with 100 steps. In order to facilitate reproducibility, we fixed the random seed to 1 for all experiments (including both training and testing). All experiments were conducted on a single NVIDIA RTX 3090 GPU.

Hyperparameter	Symbol	Value
Relational Tolerance Margin	α	0.1
Relational Temperature	τ_{rel}	0.6
Sigmoid Temperature	τ_{sig}	1.0
Stability Constant	δ_{eps}	1e-6

Table B. Hyperparameters included in the loss functions.

Dataset	Stage	Learning-rate	Batch-size	Epoch
CGL	Pre-training	1e-4	32	500
	LoRA Fine-tuning	2e-3	32	300
PKU	Pre-training	2e-4	128	500
	LoRA Fine-tuning	3e-3	128	300

Table C. Training hyperparameters for UniLayDiff on the PKU and CGL datasets.

Uncond	C $\rightarrow$ S + P	C + S $\rightarrow$ P	Completion	Relationship
2/5	1/5	1/5	1/5	–
1/3	1/6	1/6	1/6	1/6

Table D. Sampling proportions of different tasks during pre-training and fine-tuning.

	#Param	Inference Memory	Inference Time
Autoregressive			
RALF	43M	8.4GB	15ms/sample
Scan-and-Print	26M	1.4GB	5.8ms/sample
Diffusion-based			
LayoutDiT	48M	1.3GB	57ms/sample
UniLayDiff	59M	1.7GB	93ms/sample
LLM-based			
PosterLlama	7B	39GB	6380ms/sample
PosterO	8B	23GB	3881ms/sample

Table E. Comparison of computational cost.

C. Additional Experimental Results

C.1 Computational Cost

We compare the computational cost of UniLayDiff against several state-of-the-art (SOTA) models, including the autoregressive-based RALF [2] and Scan-and-Print [4], the diffusion-based LayoutDiT [5], and the LLM-based PosterLlama [8] and PosterO [3] in Tab E. The comparison evaluates model parameters, inference time, and memory usage.

The parameter count and inference time of UniLayDiff are slightly higher than those of the diffusion-based LayoutDiT. However, UniLayDiff is designed as a unified model, offering superior parameter efficiency. Simultaneously, our model holds a significant advantage over LLM-based approaches regarding both inference speed and model size. Additionally, autoregressive models achieve faster infer-

Method	#Params	PKU annotated					CGL annotated				
		Content		Graphic			Content		Graphic		
		Occ ↓	Rea ↓	Und_L ↑	Und_S ↑	Ove ↓	Occ ↓	Rea ↓	Und_L ↑	Und_S ↑	Ove ↓
LLM-Based											
PosterLlama	7B	0.163	0.0285	0.998	0.992	0.0006	0.468	0.0555	0.997	0.977	0.0014
LayoutPrompter	8B	0.254	0.0412	0.405	0.162	0.0010	0.457	0.0644	0.323	0.114	0.0026
PosterO	8B	0.144	0.0248	0.985	0.924	**0.0004**	0.291	0.0365	0.981	0.903	**0.0004**
Non-LLM-based											
CGL-GAN	41M	0.213	0.0256	0.718	0.299	0.1034	0.496	0.0604	0.652	0.129	0.2478
DS-GAN	30M	0.179	0.0228	0.764	0.464	0.0368	0.407	0.0533	0.804	0.349	0.0831
ICVT	50M	0.295	0.0254	0.420	0.301	0.3167	0.464	0.0502	0.466	0.306	0.1959
LayoutDM†	43M	0.153	0.0220	0.645	0.235	0.3051	0.432	0.0499	0.730	0.539	0.0979
RALF	43M	0.142	0.0190	0.965	0.878	0.0106	**0.258**	**0.0319**	0.986	0.935	0.0572
LayoutDiT	48M	0.124	0.0161	**0.999**	**0.994**	**0.0004**	0.339	0.0358	0.998	0.989	0.0023
Scan-and-Print	26M	**0.116**	0.0181	0.983	0.970	0.0090	0.336	0.0323	0.985	0.974	0.0157
UniLayDiff(Ours)	59M	0.118	**0.0155**	**0.999**	**0.994**	0.0009	0.358	0.0335	**0.999**	**0.994**	0.0008

Table F. Unconstrained generation results on the PKU and CGL unannotated test split.

Method	PKU annotated							CGL annotated						
	Occ↓	Rea↓	Und_L↑	Und_S↑	Ove↓	FID ↓	Vio%↓	Occ↓	Rea↓	Und_L↑	Und_S↑	Ove↓	FID ↓	Vio%↓
RALF	0.124	0.0138	0.960	0.869	0.0085	2.41	34.2	0.126	0.0152	0.975	0.954	0.0060	**0.73**	36.1
UniLayDiff(Ours)	**0.119**	**0.0131**	**0.971**	**0.883**	**0.0049**	**2.15**	**22.4**	**0.121**	**0.0137**	**0.984**	**0.958**	**0.0043**	1.74	**21.6**

Table G. Generation results for relation task on the PKU and CGL annotated test split.

Hybrid Tasks	Content		Graphic			
	Occ ↓	Rea ↓	Und_L ↑	Und_S ↑	Ove ↓	FID ↓
Comp + (C → (S + P))	0.121	0.0130	0.981	0.951	0.003	2.34
Comp + ((C + S) → P)	0.125	0.0132	0.974	0.894	0.005	0.76
Comp + Relation	0.122	0.0129	0.974	0.894	0.0060	2.04

Table H. Quantitative evaluation for the hybrid tasks on the PKU annotated test split.

ence in this context. This discrepancy arises because layout generation tasks typically involve a limited number of elements, resulting in fewer autoregressive iterations compared to the fixed sampling steps required by diffusion models.

C.2 Generalization and Robustness

To validate the robustness of UniLayDiff, we conduct evaluations across three distinct dimensions: hybrid tasks, unannotated data, and cross-dataset scenarios.

Generalization to hybrid tasks. We present the quantitative results on a set of hybrid generation tasks to further demonstrate the versatility and robustness of UniLayDiff. It is worth noting that previous task-specific models are incapable of handling these hybrid scenarios, as they are typically trained exclusively on single tasks. In contrast, by flexibly combining the Partial Constraint Mask $\mathbf{M}_P$ and the relation matrix $\mathbf{M}_R$, our unified framework can seamlessly adapt to these complex scenarios. Three hybrid tasks could be defined as follows:

Comp + (C → (S + P)). This task combines layout completion with category-conditioned generation. Specifically, a subset of elements is fully specified (including category, size, and position), serving as the completion context. For the remaining elements, only their categories are provided.

Comp + ((C + S) → P). This task integrates completion with category-and-size-conditioned generation. In addition to a set of fully defined elements (completion part), the categories and sizes of the remaining elements are given.

Comp + Relation. This task imposes relational constraints on top of the standard completion task.

The quantitative results for these hybrid tasks are summarized in Tab. H. These results highlight the versatility of UniLayDiff, demonstrating that our unified training strategy and MM-DiT architecture effectively empower the model to support a diverse range of constraint combinations.

Train	Test	Method	Occ ↓	Rea ↓	Und_L ↑	Und_S ↑	Ove ↓
CGL	PKU-unanno	RALF	0.143	0.0170	0.989	0.958	0.0347
		LayoutDiT	0.115	0.0178	**0.998**	0.980	0.0025
		Ours	**0.107**	**0.0165**	**0.998**	**0.986**	**0.0012**
	PKU-anno	RALF	0.140	**0.0113**	0.995	0.976	0.0078
		LayoutDiT	0.108	0.0142	0.995	0.987	0.0015
		Ours	**0.105**	**0.0113**	0.996	**0.990**	**0.0009**
PKU	CGL-unanno	RALF	0.342	0.0425	0.927	0.773	0.0309
		LayoutDiT	**0.328**	0.0331	**1.000**	0.992	0.0053
		Ours	0.343	**0.0275**	0.999	**0.993**	**0.0044**
	CGL-anno	RALF	0.150	0.0170	0.974	0.902	0.0065
		LayoutDiT	0.128	0.0147	**0.999**	**0.995**	0.0040
		Ours	**0.123**	**0.0138**	**0.999**	0.992	**0.0035**

Table I. Cross-dataset evaluation setup: We train a model on PKU and test it on CGL, or vice versa.

Results on unannotated splits. Tab. F reports unconditional generation performance on the unannotated test set. The results demonstrate that our method maintains strong

Scheme	Occ ↓	Rea ↓	Und$_S$ ↑	Ove ↓	FID ↓	Vio%↓
All	**0.119**	0.0135	**0.885**	0.0053	2.43	25.1
Layout-only	0.120	0.0135	0.876	**0.0047**	2.39	24.7
Mask-only	0.122	0.0133	0.874	0.0052	2.65	32.3
Diag(Ours)	**0.119**	**0.0131**	0.883	0.0049	**2.15**	**22.4**

Table J. Ablation study on relative positional bias for relation injection using the PKU annotated dataset, evaluated using the relationship task.

Scheme	Occ ↓	Rea ↓	Und$_S$ ↑	Ove ↓	FID ↓	Vio% ↓
Dual Branch FT	**0.115**	**0.0128**	0.879	0.0053	2.44	23.5
Mask-Layout FT	0.119	0.0131	**0.883**	**0.0049**	**2.15**	**22.4**

Table K. Ablation study on LoRA fine-tuning strategy for relation using the PKU annotated dataset.

Salbox	Image	Occ↓	Rea↓	Und$_S$ ↑	Ove ↓	FID ↓
✗	✗	0.127	0.0131	0.994	0.0006	4.61
✓	✗	0.124	0.0126	0.983	0.0007	3.61
✓	✓	0.122	0.0120	0.992	0.0005	3.37

Table L. Ablation study on the impact of different visual guidance components (Saliency Box and Image) injected via adaLN-Zero on the PKU annotated dataset, evaluated using the unconditional generation task with pre-trained models.

performance on the challenging unannotated data.

Cross-dataset Evaluation. The generalization of our method is verified through cross-dataset experiments. The model is trained on PKU and tested on CGL, and subsequently, trained on CGL and tested on PKU. As shown in Tab. I, UniLayDiff exceeded baselines on most metrics. These results confirm its robustness and adaptability to diverse real-world images.

C.3 More Results for Relation Task

Due to space constraints in the main text, we did not include the relation violation rate for the relation task in the main table. We report the full relation generation performance including the violation rate in Tab. G as a supplement. Experimental results show that our unified model outperforms RALF in terms of relation violation rate as well as most of the other metrics, demonstrating the generative advantages of our model.

C.4 More Results on Ablation Study

Relative positional bias for relation injection. To validate the effectiveness of our masked relative positional bias strategy, we compare it with several alternative schemes in Tab. J. The schemes are:

(a) *All*: The relative positional bias is applied to all attention pairs across modalities.

(b) *Layout-only*: The relative positional bias is restricted solely to the attention between layout elements.

λ_1	Occ ↓	Rea ↓	Und$_S$ ↑	Ove ↓	FID ↓	Vio%↓
0.05	0.115	0.0127	0.884	0.0040	2.24	29.6
0.1	0.123	0.0129	0.885	0.0042	2.41	25.2
0.2(Ours)	0.119	0.0131	0.883	0.0049	2.15	22.4
0.5	0.122	0.0135	0.876	0.0056	2.96	21.3

Table M. Ablation study on different coefficient values for $\mathcal{L}_{\text{rel}}$ term on the PKU annotated dataset, evaluated using the relationship task.

λ_2	Occ ↓	Rea ↓	Und$_S$ ↑	Ove ↓	FID ↓
0.2	0.135	0.0129	0.964	0.0015	2.54
0.4**(Ours)**	0.128	0.0129	0.963	0.0018	3.09
1	0.125	0.0125	0.958	0.0023	3.42

Table N. Ablation study on different coefficient values for $\mathcal{L}_{\text{ctn}}$ term on the PKU annotated dataset, using the C → S + P task.

λ_3	Occ ↓	Rea ↓	Und$_S$ ↑	Ove ↓	FID ↓
0.2	0.114	0.0118	0.992	0.0006	3.36
0.5	0.113	0.0119	0.994	0.0007	3.28
1**(Ours)**	0.115	0.0114	0.996	0.0005	3.15
2	0.116	0.0115	0.998	0.0009	3.44

Table O. Ablation study on different coefficient values for $\mathcal{L}_{\text{und}}$ term. on the PKU annotated dataset, evaluated using the unconditional generation task.

(c) *Mask-only*: The relative positional bias is restricted solely to the attention between mask elements.

(d) *Diag* : Our proposed masked relative positional bias, which effectively models the relationships within each modality as shown in Eq. 6 of the main text.

As shown in Tab. J, the *Diag* scheme achieves the best performance, indicating its strong capability in relationship modeling. Compared to the *All* scheme, our *Diag* approach is based on a clearer premise, as the concept of relative position between elements of different modalities is not always meaningful.

LoRA fine-tuning strategy for relation.

In UnilayDiff, LoRA fine-tuning is specifically applied to the Partial Constraint Mask branch ($\mathbf{F}_M$-$\mathbf{F}_L$) rather than both branches of the MM-DiT (Sec. 3.5). To validate this design, we compare two different LoRA application strategies:

(a) *Dual Branch FT*: LoRA fine-tuning is applied to both the Image-Layout ($\mathbf{F}_I$-$\mathbf{F}_L$) and Mask-Layout ($\mathbf{F}_M$-$\mathbf{F}_L$) branches.

(b) *Mask-Layout FT* (Ours): LoRA is applied exclusively to the Mask-Layout interaction branch ($\mathbf{F}_M$-$\mathbf{F}_L$).

The results are summarized in Tab. K. We observe that the *Dual* strategy does not yield performance gains for the relationship task (Vio%) compared to our *Mask-only* strategy. Although *Dual* shows a slight improvement in the Overlay

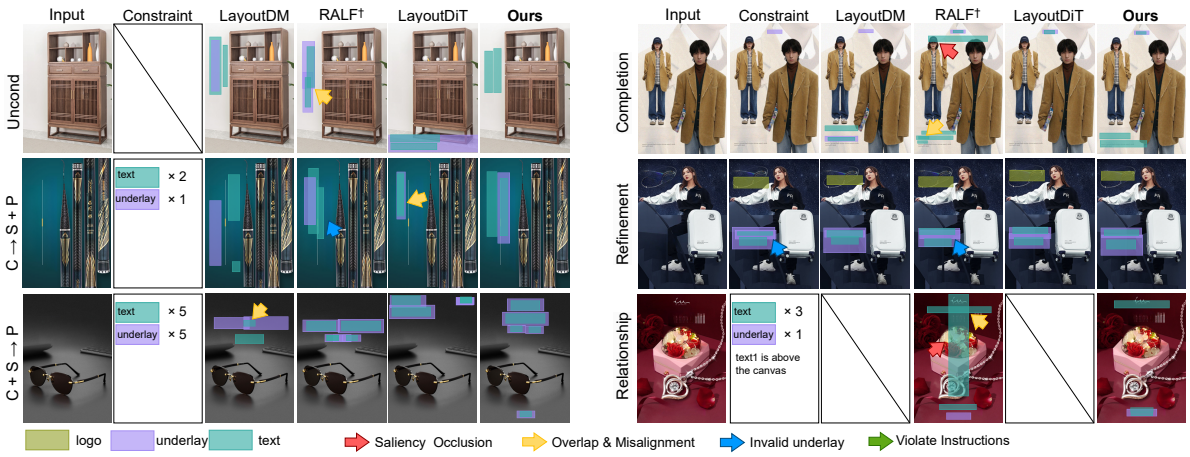

Figure A. Visual comparison with baselines on the CGL annotated test split.

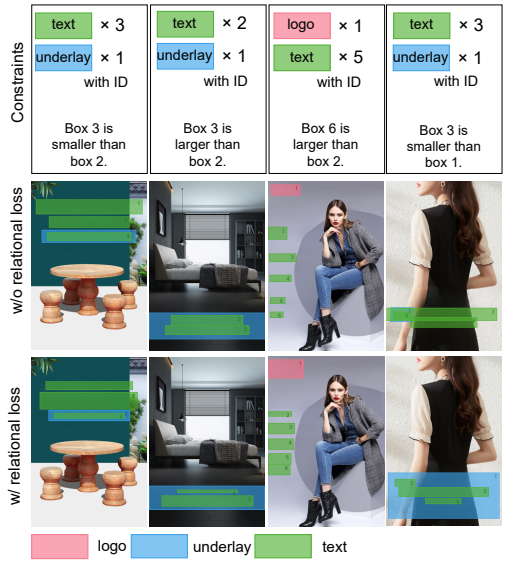

Figure B. Ablation study on the impact of $\mathcal{L}_{\text{rel}}$. The IDs in the figure represent the natural order of the layout elements, following the setting of RALF.

metric (Ove), it leads to performance degradation in other metrics. This suggests that relational constraints are inherently structural and are most effectively injected through the mask branch.

Impact of adaLN-Zero. As introduced in Sec. 3.3, we utilize adaLN-Zero to inject global visual guidance, comprising saliency bounding boxes $\mathbf{B}$ and global image features. To validate this, we evaluate three configurations on the PKU dataset: (1) time embedding only; (2) adding $\mathbf{B}$; and (3) our full method with both.

Results in Tab. L show that the baseline lacks visual cues, yielding suboptimal FID. Incorporating $\mathbf{B}$ significantly improves generation quality and content harmony, indicating that explicit structural guidance helps the model avoid occlusion. Furthermore, integrating the global image feature achieves the best performance across all metrics (FID 3.37, Occ 0.122), validating that global visual context is essential for generating high-quality layouts.

Layers	Occ ↓	Rea ↓	Und$_S$ ↑	Ove ↓	FID ↓
6	0.123	0.0130	0.990	0.0022	4.48
8	0.125	0.0124	**0.993**	0.0015	5.05
10	0.124	0.0132	0.992	**0.0005**	4.13
12(Ours)	**0.122**	**0.0120**	0.992	**0.0005**	**3.37**
14	0.127	0.0124	0.988	0.0012	4.26
16	0.124	0.0126	0.989	0.0008	4.44

Table P. Ablation studies of model scale on the PKU annotated dataset, in which we experimentally evaluate the models from pre-training on the unconditional generation task.

Impact of additional constraints. The ablation study results for the coefficients of additional constraints are shown in Tab. M, Tab. N and Tab. O. With our chosen coefficients, the model achieves a balanced performance across various metrics. Moreover, We conduct visual ablation study on relation training methods, shown in Fig. B. Fig. B demonstrates that $\mathcal{L}_{\text{rel}}$ significantly enhances the model's adherence to size relations.

Model scale. We further evaluate the effect of model complexity by experimenting with different MM-DiT depths. As demonstrated in Tab. P, the model achieves its best performance at a depth of 12 layers.

C.5 Visual Comparison

We provide additional visual examples of UniLayDiff in Fig. A, Fig. C, Fig. D, and Fig. E. Among these, Fig. A shows a set of results generated by the model for each subtask on the CGL annotated split. In Fig. C and Fig. D, we provide additional generated results on the PKU and CGL annotated splits to supplement the visualizations in Fig. A and the main text. Furthermore, we demonstrate the generation performance of UniLayDiff on real-world data (the unannotated split) in Fig. E. The visual results show that UniLayDiff can generate high-quality and diverse layouts.

References

[1] Jian Chen, Ruiyi Zhang, Yufan Zhou, Rajiv Jain, Zhiqiang Xu, Ryan Rossi, and Changyou Chen. Towards aligned layout generation via diffusion model with aesthetic constraints. In *The Twelfth International Conference on Learning Representations*, 2024. 1

[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*, 2024. 1, 2

[3] HsiaoYuan Hsu and Yuxin Peng. Postero: Structuring layout trees to enable language models in generalized content-aware layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 2

[4] HsiaoYuan Hsu and Yuxin Peng. Scan-and-print: Patch-level data summarization and augmentation for content-aware layout generation in poster design. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2025. 2

[5] Yu Li, Yifan Chen, Gongye Liu, Fei Yin, Qingyan Bai, Jie Wu, Hongfa Wang, Ruihang Chu, and Yujiu Yang. Layoutdit: exploring content-graphic balance in layout generation with diffusion transformer. *arXiv preprint arXiv:2407.15233*, 2024. 1, 2

[6] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7479–7489, 2019. 1

[7] Xuebin Qin, Hang Dai, Xiaobin Hu, Deng-Ping Fan, Ling Shao, and Luc Van Gool. Highly accurate dichotomous image segmentation. In *ECCV*, 2022. 1

[8] Jaejung Seol, Seojun Kim, and Jaejun Yoo. Posterllama: Bridging design ability of language model to content-aware layout generation. In *Proceedings of the European Conference on Computer Vision*, pages 451–468, 2024. 2

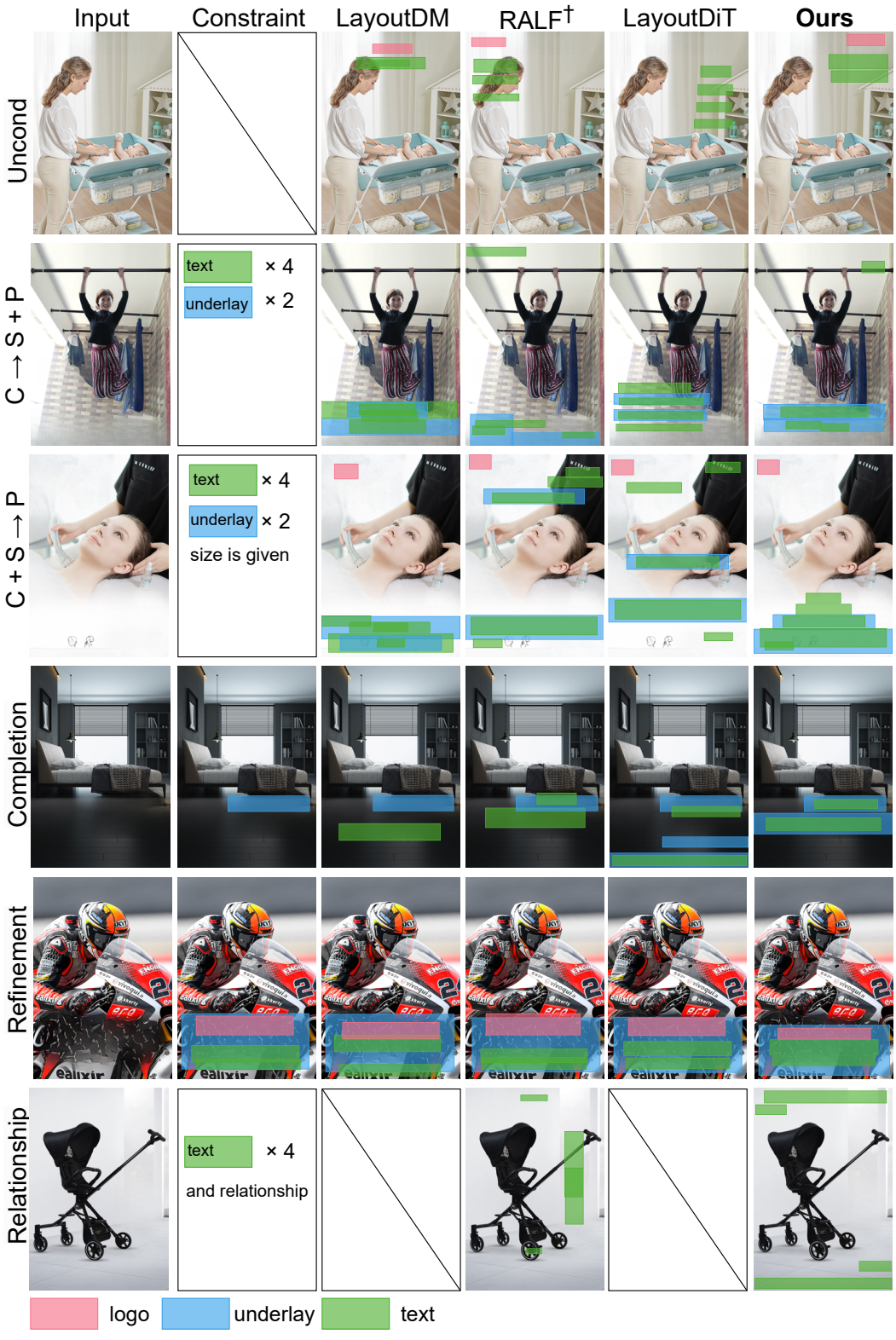

Figure C. Visual comparison with baselines on the PKU annotated test split.

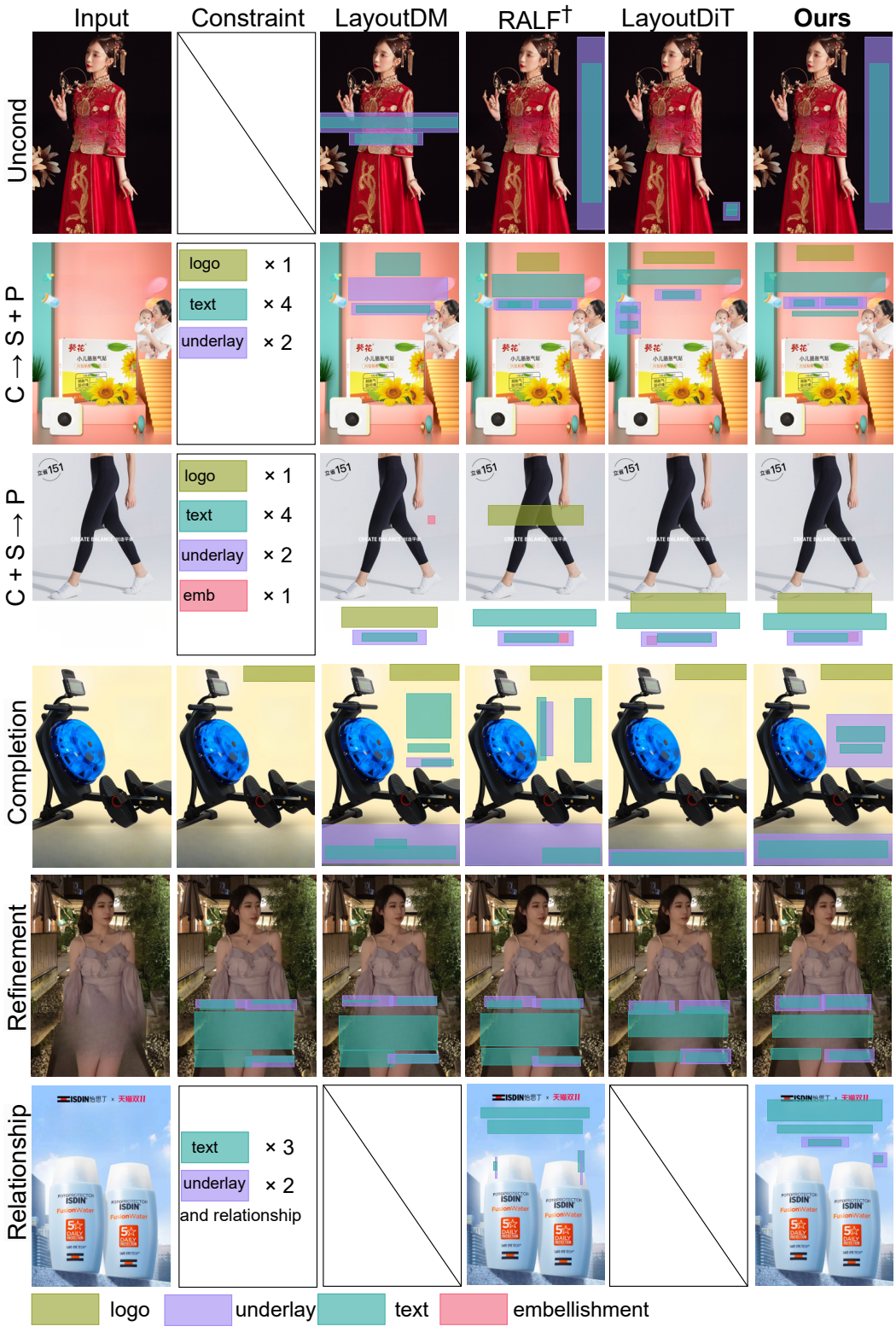

Figure D. More visual comparison with baselines on the CGL annotated test split.

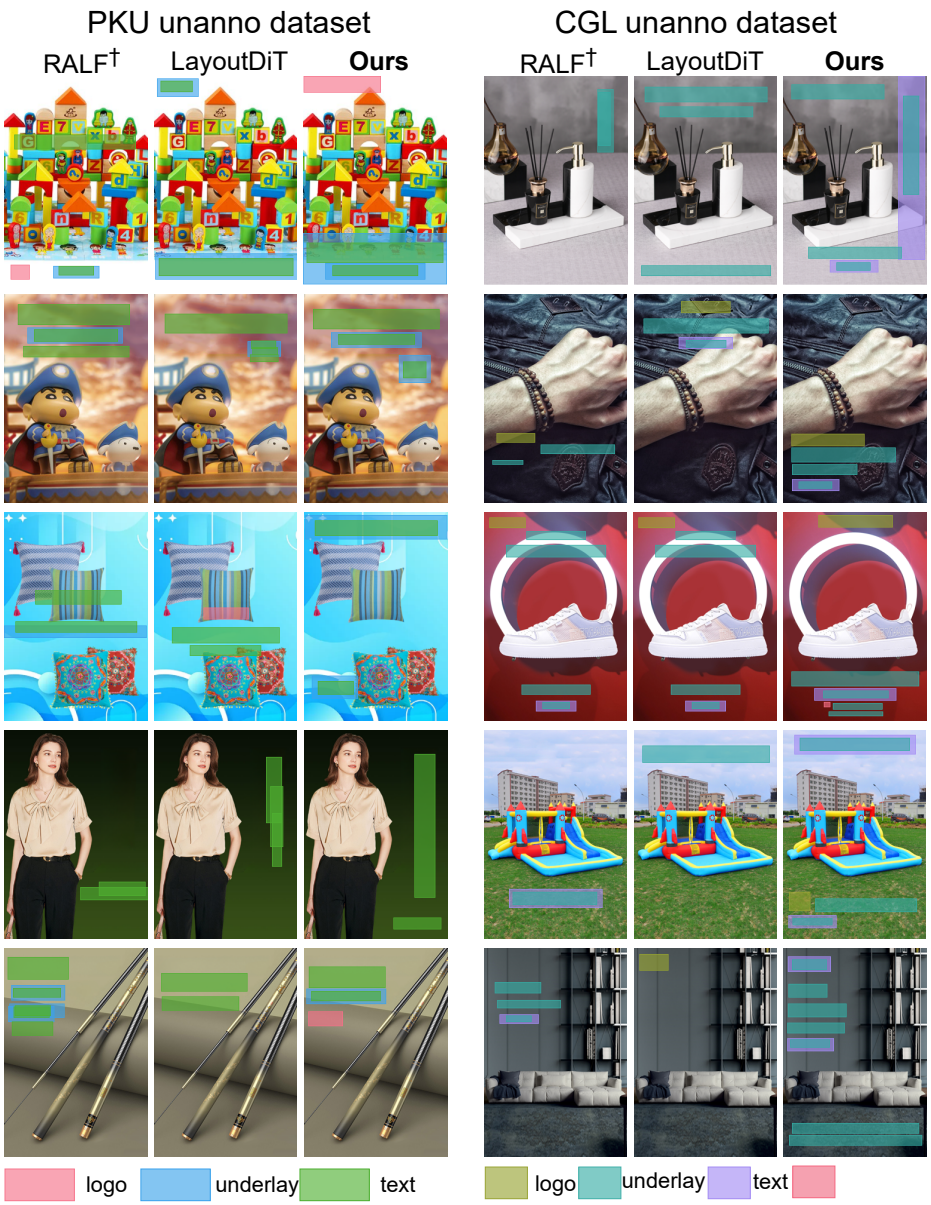

Figure E. Visual comparison with baselines on PKU and CGL unannotated test splits.