A ADDITIONAL ABLATION RESULTS

We compared LACE with LayoutDM and LayoutGAN++ in terms of overlap and alignment. LayoutGAN++ is a task-specific model that also uses constraints in training and post-processing. We directly adopt their results in the original paper. In addition, we apply post-processing to LayoutDM to demonstrate the advantage in using constraint during training for diffusion model. Results are shown in Table A.1 and Table A.2. Both LayoutDM and LayoutGAN++ show a notable FID increase after post-processing. In contrast, LACE maintains a stable FID and achieves lower overlap and alignment scores before post-processing, which are further improved afterward.

	Task	C→S+P		
Model	Metric	FID↓	Align↓	Overlap.↓
Task-specific models				
NDN-none		61.1	0.350	16.5
LayoutGAN++		24.0	0.190	22.80
LayoutGAN++ w/ $\mathcal{C}$		22.3	0.160	14.27
LayoutGAN++ w/ $\mathcal{C}$ & post		26.2	0.160	1.18
Diffusion-based models				
LayoutDM		7.95	0.106	16.43
LayoutDM w/ post		15.2	0.083	6.076
LACE w/o $\mathcal{C}$		6.12	0.054	1.636
LACE (local)		4.88	0.043	1.638
LACE (global)		5.14	0.046	1.791
LACE (local) w/ post		4.63	0.010	1.211
LACE (global) w/ post		4.56	0.009	0.906
Validation data		6.25	0.021	0.117

Table A.1: FID, overlap and alignment results in the C→S+P task on the PubLayNet dataset.

Additionally, LACE, even without post-processing, outperforms LayoutDM with post-processing in alignment and overlap scores. This serves as proof of the effectiveness of the constraint loss in our approach.

	Task	C+S→P		Completion		U-Cond	
Model	Metric	Align↓	Overlap.↓	Align↓	Overlap.↓	Align↓	Overlap.↓
Diffusion-based models							
LayoutDM		0.119	18.91	0.107	15.04	0.195	13.43
LayoutDM w/ post		0.117	6.506	0.073	5.220	0.200	4.641
LACE w/o $\mathcal{C}$		0.065	3.062	0.054	3.223	0.238	7.533
LACE (local)		0.061	3.309	0.040	2.772	0.141	3.615
LACE (global)		0.061	3.439	0.042	3.056	0.185	4.140
LACE (local) w/ post		0.016	1.400	0.014	1.723	0.032	0.586
LACE (global) w/ post		0.017	1.363	0.017	1.573	0.074	0.768
Validation data		0.021	0.117	0.021	0.117	0.021	0.117

Table A.2: Overlap and alignment results on PubLayNet for three generation tasks.

B IMPLEMENTATION SPECIFICATIONS

B.1 TIME-DEPENDENT CONSTRAINT WEIGHT

The time-dependent constraint weight is critical for effective model convergence and output quality. Without this weight, the model struggles to converge, leading to a high Fréchet Inception Distance (FID) score, typically remaining above 100, which indicates poor layout quality. We choose $\omega_t = (1 - \bar{\alpha}_t)$ of a constant β schedule as the constraint weight series. The β schedule is set empirically such that the weight activates the constraint only when t is small when the corruption process has not introduced too much overlap, as demonstrated in Figure B.1. Thus, in the reverse process, the coarse structure of the layout has emerged. Figure B.2 demonstrates the local minimum induced by the constraint functions at noisy steps, hindering convergence.

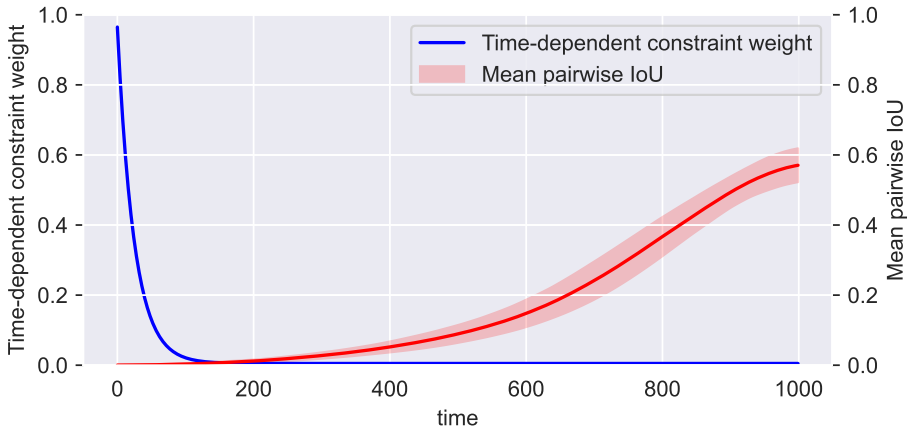

Figure B.1: Time-dependent constraint weight and Mean Pairwise IoU in the forward process.

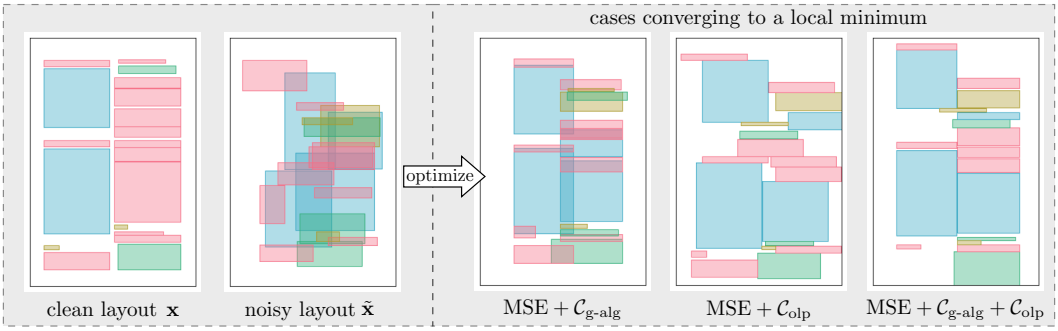

Figure B.2: Examples of convergence to local minimum with alignment and overlap constraints

B.2 Post-processing threshold

We use the global alignment and overlap (only on PubLayNet) constraints to optimize the raw output of LACE. Since there is no target layout to compute the ground truth alignment mask matrix in Eq. (9), we use a threshold value to compute an alignment mask matrix using the coordinate difference matrix of the generated layout. To determine the threshold for enhanced visual quality post-processing, we first scaled the normalized canvas according to its width/height ratio. We then tested various threshold values, including 1/16, 1/32, 1/64, 1/128, and 1/256. The optimal threshold was empirically determined to be 1/64 of the scaled normalized canvas size. This setting aligns with real dataset observations, where only 0.5% of unaligned coordinate pairs have a smaller difference.

B.3 Model architecture

As illustrated in Figure B.3, we adopt a transformer architecture that is implemented in the source code of LayoutDM Inoue et al. (2023) to predict the noise term in Eq. (5). We also choose similar hyper-parameter settings for a fair comparison: 4 layers, 16 attention heads, 2048 hidden dimension in the FNN (feed-forward networks), and the embedding dimension is 1024 for PubLayNet and 512 for Rico. In addition, we add two FNNs to encode and decode element vectors. Time embeddings are injected by an modified adaptive layer normalization Dumoulin et al. (2017). Specifically, the layer normalization is:

$$\mathbf{y} = (\mathbf{1} + f_\gamma(\mathbf{v}_t)) \odot \left(\frac{\mathbf{x} - \boldsymbol{\mu}}{\boldsymbol{\sigma}} \right) + f_\beta(\mathbf{v}_t), \tag{B.1}$$

where $\mathbf{x}$, $\mathbf{y}$ are the input and output of the normalization function, f_γ, f_β are FFN that encode the time-dependent scale and shift, $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$ are $\mathbf{x}$'s mean and standard deviation, $\mathbf{v}_t$ is the time embedding, $\mathbf{1}$ is a vector of all ones represents a residue connection. $\odot$ is the Hadamard product.

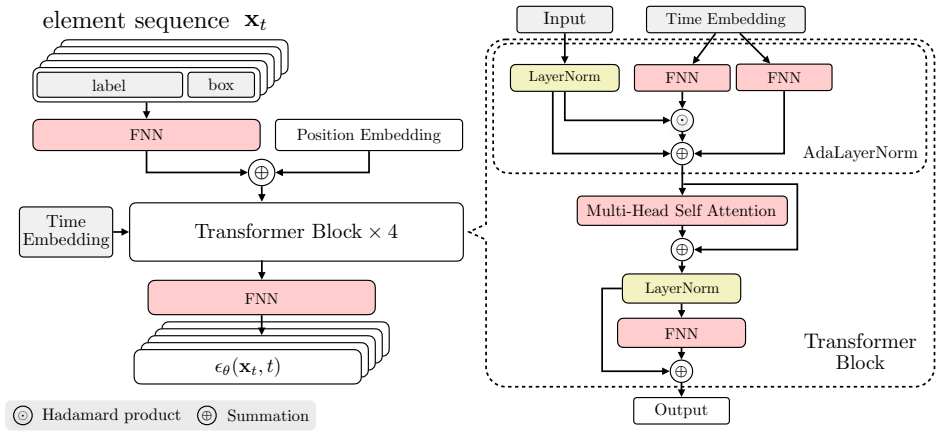

Figure B.3: Neural network architecture for the layout generation diffusion models. The the network takes sequence of layout elements the time variables as input and output the predicted noise. The pink blocks in the figure represent the trainable network components, gray blocks represent input tensors.

B.4 Training details

We train the model using the Adam optimizer. The batch size is 256. We used a learning rate schedule that included a warmup phase followed by a half-cycle cosine decay. The initial learning rate is set to 0.001. Training is divided into two phases: initially, the model is trained without constraints ($\omega_t = 0$) until convergence is observed in the FID score. In the second phase, constraints are added to the total loss, and training continues until convergence is achieved in both alignment and FID scores. For the Rico dataset, the overlap constraint is excluded due to prevalent overlap patterns in real data. However, in the PubLayNet dataset training, the overlap constraint is applied to prevent undesirable overlaps in publication layouts. The diffusion model employs a total of 1000 forward steps. For efficient generation, we use DDIM sampling with 100 steps.

C QUALITATIVE COMPARISON

Figure C.1: Qualitative comparison between LACE w/ post-processing (left), real (middle), and LayoutDM (right) in conditional generation tasks (C+S $\rightarrow$ P) on the PublayNet dataset.

Figure C.2: Qualitative comparison between LACE w/ post-processing (left), real (middle), and LayoutDM (right) in conditional generation tasks (C+S $\rightarrow$ P) on the Rico dataset.

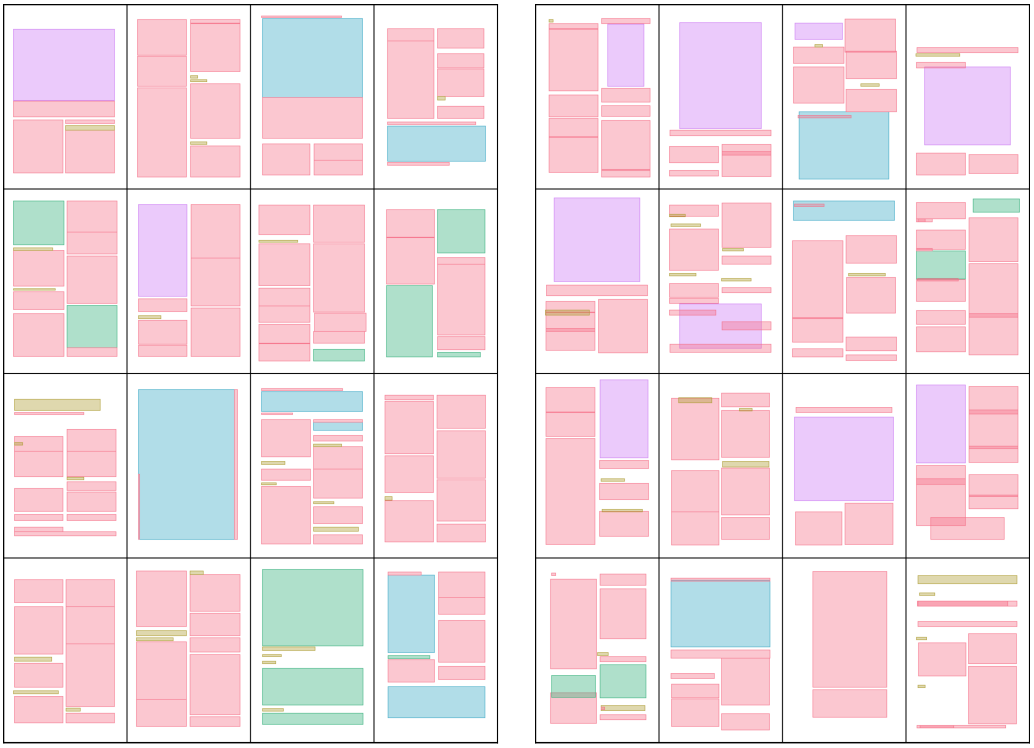

Figure C.3: Qualitative comparison between LACE w/ post-processing (left) and LayoutDM (right) in unconditional generation tasks on the PublayNet dataset.

Figure C.4: Qualitative comparison between LACE w/ post-processing (left) and LayoutDM (right) in unconditional generation tasks on the Rico dataset.