Generating Handwritten Mathematical Expressions From Symbol Graphs: An End-to-End Pipeline (Supplementary Material)

Yu Chen[1,*], Fei Gao[2,*], Yanguang Zhang[3], Maoying Qiao[4], Nannan Wang[5,†]
[1] Beijing Waiyan Online Digital Technology ; [2] Hangzhou Institute of Technology, Xidian University
[3] Hangzhou Dianzi University; [4] University of Technology, Sydney (UTS); [5] Xidian University
{fgao, nnwang}@xidian.edu.cn, ppak1991@outlook.com, maoying.qiao@uts.edu.au

Abstract

In this paper, we solve a novel challenging generation task, i.e. Handwritten Mathematical Expression Generation (HMEG) from symbolic sequences. Since symbolic sequences are naturally graph-structured data, we formulate HMEG as a graph-to-image (G2I) generation problem. Unlike the generation of natural images, HMEG requires critic layout clarity for synthesizing correct and recognizable formulas, but has no real masks available to supervise the learning process. To alleviate this challenge, we propose a novel end-to-end G2I generation pipeline (i.e. $graph \rightarrow layout \rightarrow mask \rightarrow image$), which requires no real masks or nondifferentiable alignment between layouts and masks. Technically, to boost the capacity of predicting detailed relations among adjacent symbols, we propose a Less-is-More (LiM) learning strategy. In addition, we design a differentiable layout refinement module, which maps bounding boxes to pixel-level soft masks, so as to further alleviate ambiguous layout areas. Our whole model, including layout prediction, mask refinement, and image generation, can be jointly optimized in an end-to-end manner. Experimental results show that, our model can generate high-quality HME images, and outperforms previous generative methods. Besides, a series of ablations study demonstrate effectiveness of the proposed techniques. Finally, we validate that our generated images promisingly boosts the performance of HME recognition models, through data augmentation. Our code will be released after peer review.

1. Analysis of Pixel-level Mask

In the proposed method, we propose a Sequential BBox-to-Mask Transformation (B2M) module, to alleviate a. transform *hard* object-level layouts to *soft* pixel-level masks. Figure 1 shows several examples of the predicted layouts, masks, and generated images. Obviously, there are typically overlaps among symbols in a formula. While, in soft masks, the overlapped areas are become less ambiguous. Namely, there are less overlaps between adjacent symbols. Besides, masks are consistent with generated images, in terms of structure. Recall that in the ablation study, removing B2M seriously decrease either the quantitative and qualitative performance. Such results demonstrate the significance of B2M in boosting image generation performance.

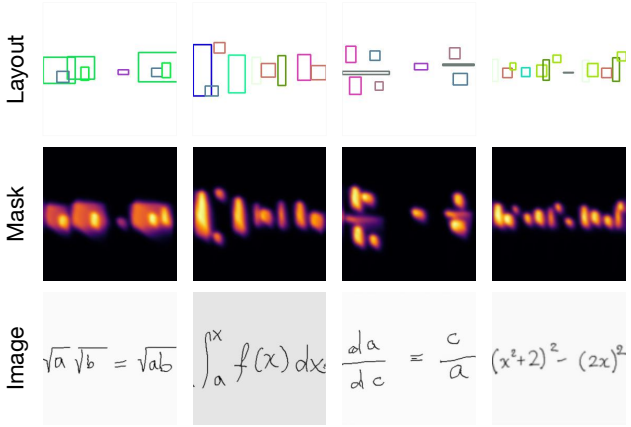

Figure 1. Illustrations of predicted layouts, masks, and generated images

2. Detailed Network Architectures

Layout Prediction The architectures of our layout predictor G_{box} and the layout discriminator D_{box} are shown in Table 1 and Table 2, respectively.

Grid-to-Mask Projection The architecture of our Grid-to-Mask Projector in the Sequential BBox-to-Mask Transformation (B2M) module is shown in Table 3.

Image Decoder The architecture of our image decoder follows Cascaded Refinement Network (CRN). The architecture is shown in Table 4.

Image Discriminator The architecture of our image discriminator is shown in Table 5.

Graph Embedding Model	
input:	concat($\mathbf{h}_i^{(0)}$(128), $\mathbf{e}_{ij}$(128), triple(3))
1.	GCN(128, 512, avg), Relu
2.	GCN(512, 512, avg), Relu
3.	GCN(512, 512, avg), Relu
4.	GCN(512, 512, avg), Relu
output:	$\mathbf{h}_i^{(T)}$ (128)
5.	GCN(512, 128, avg), Relu
output:	$\mathbf{h}_G$(128)
Box Regression Network f_{bbox}	
input:	concat($\mathbf{h}_G$(128), $\mathbf{h}_i^{(T)}$(128), noise $\mathbf{z}$ (64))
1.	Attn(128, 128)
2.	Linear(320, 512), BN, Relu
3.	Linear(512, 4), BN, Relu
output:	boxes(4)

Table 1. The architectures of our layout predictor. $\mathbf{h}_i^{(0)}$ and $\mathbf{e}_{ij}$ are encoded node/edge features, respectively. *concat* denoted concatenation.

input:	concat($\mathbf{h}_G$(128), boxes(4))
1.	Linear(128, 64), BatchNorm, Relu
2.	GCN(132, 64, avg), Relu
3.	GCN(64, 32, avg), Relu
4.	Attn(32, 32)
5.	Linear(32, 32)
6.	Linear(32, 1)
output:	P(1)

Table 2. The architecture of our layout discriminator D_{box}.

input:	grid(4), obj_features(128)
1.	Conv(4, 64, s=2, k=4), BN, Relu
2.	Conv(64, 1, s=2, k=4), BN, Relu
3.	Matmul with obj_features
4.	TransConv(128, 128, s=2, k=4), BN, Relu
5.	TransConv(128, 128, s=2, k=4), BN, Relu
output:	layout_feature(128)

Table 3. The architecture of our Grid-to-Mask Projector. *TransConv* denotes Transposed Convolutional Layer.

Symbol Discriminator The architecture of our symbol discriminator is shown in Table 6.

3. More Results

3.1. Comparison with SOTAs

Figure 2 illustrates more results about comparison with existing methods, including CycleGAN [3], FormulaGAN [2], and Sg2im [1].

input:	layout feature(128)
1.	CRN(128, 1024, k=2)
2.	CRN(1024, 512, k=4)
3.	CRN(512, 256, k=8)
4.	CRN(256, 128, k=16)
5.	CRN(128, 64, k=32)
6.	Conv(64, 64, k=1), LeakyRelu
7.	Conv(64, 3, k=1)
output:	image(3)

Table 4. The architecture of our image decoder.

input:	Image(3)
1.	Conv(3, 64, s=2, k=4), BN, LeakyRelu
2.	Conv(64, 128, s=2, k=4), BN, LeakyRelu
3.	Conv(128, 256, s=2, k=4)
4.	Conv(256, 1)
output:	P(1)

Table 5. The architecture of our image discriminator.

input:	Image Crops (3)
1.	Conv(3, 64, s=2, k=4), BN, LeakyRelu
2.	Conv(64, 128, s=2, k=4), BN, LeakyRelu
3.	Conv(128, 256, s=2, k=4)
4.	Conv(256, 1)
output:	P(1)

Table 6. The architecture of our symbol discriminator.

3.2. Ablation

Figure3 illustrates more results about the ablation study. Obviously, without using the proposed LiM or B2M, there would be more overlapped symbols or unrecognisable symbols. Such results demonstrates the effectiveness of our proposed techniques.

References

[1] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1219–1228, 2018. 2, 3

[2] Matthias Springstein, Eric Müller-Budack, and Ralph Ewerth. Unsupervised training data generation of handwritten formulas using generative adversarial networks with self-attention. In *Proceedings of the 2021 Workshop on Multi-Modal Pre-Training for Multimedia Understanding*, pages 46–54, 2021. 2, 3

[3] J. Zhu, T. Park, P. Isola, and A. A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2242–2251, 2017. 2, 3

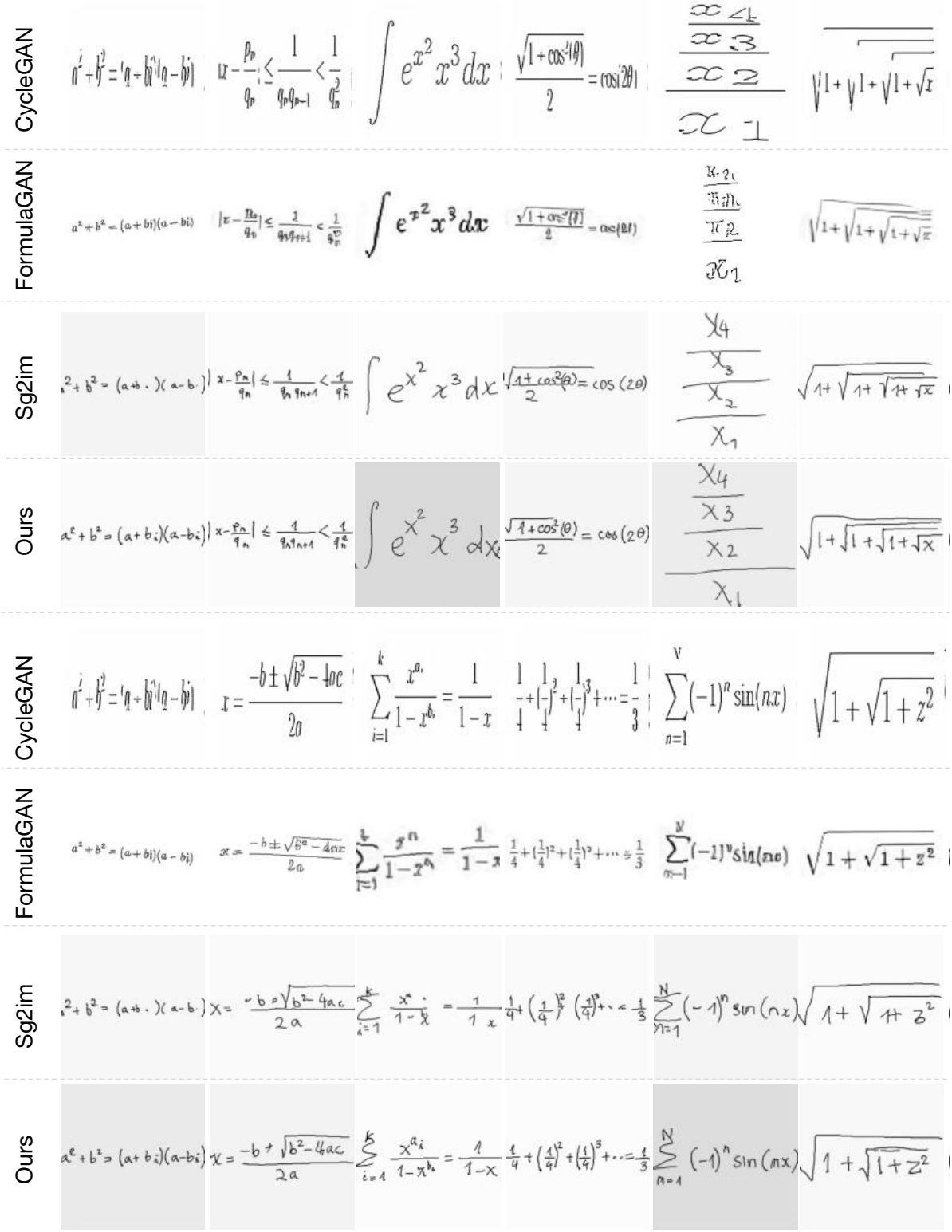

Figure 2. More comparison results with existing methods, including CycleGAN [3], FormulaGAN [2], and Sg2im [1].

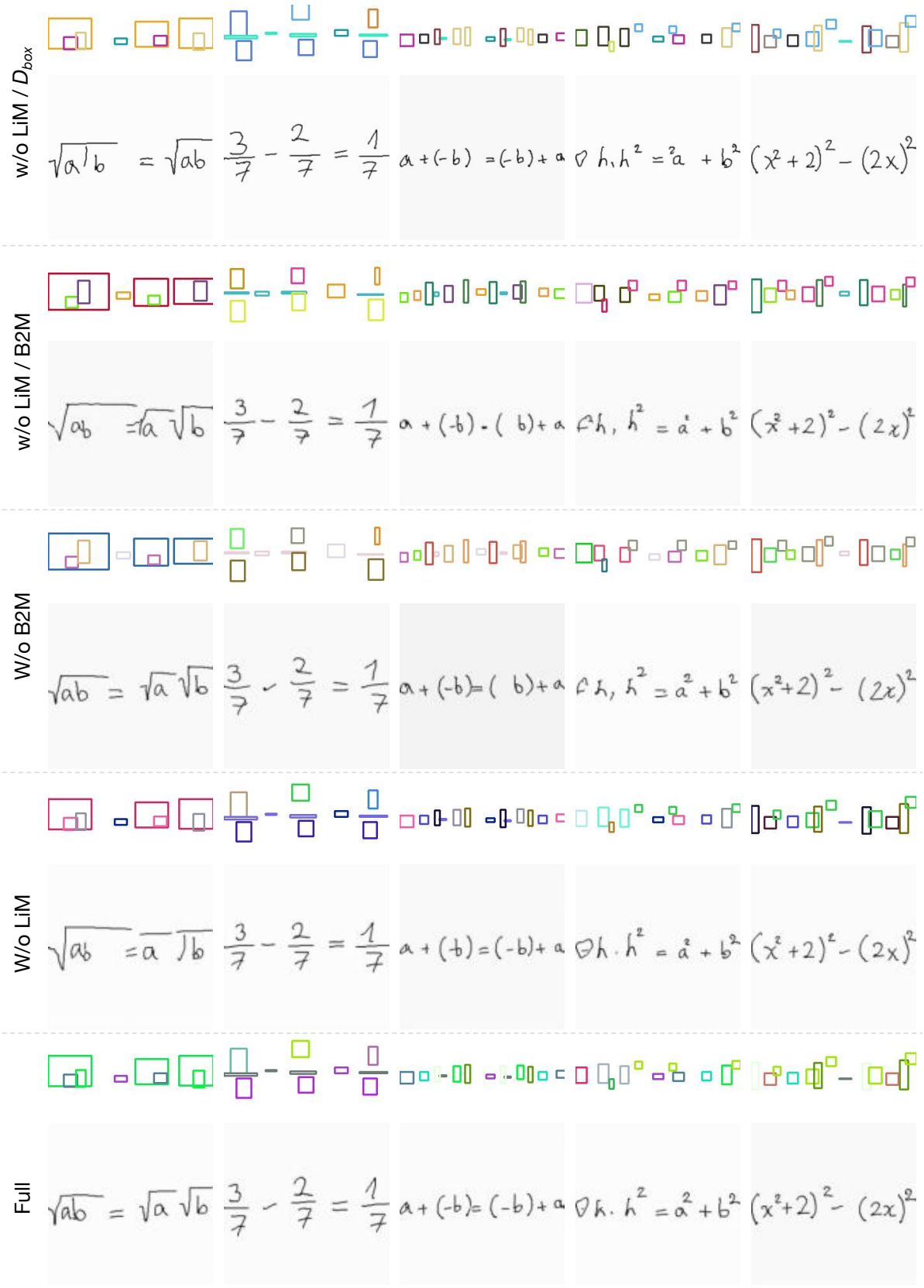

Figure 3. More results about the ablation study. LiM denotes our Less-is-More training strategy for layout prediction. B2M denotes the proposed sequential BBox-to-Mask transformation module. For each model variant, the predicted layouts and the generated images are shown in parallel.