Generating Handwritten Mathematical Expressions From Symbol Graphs: An End-to-End Pipeline

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

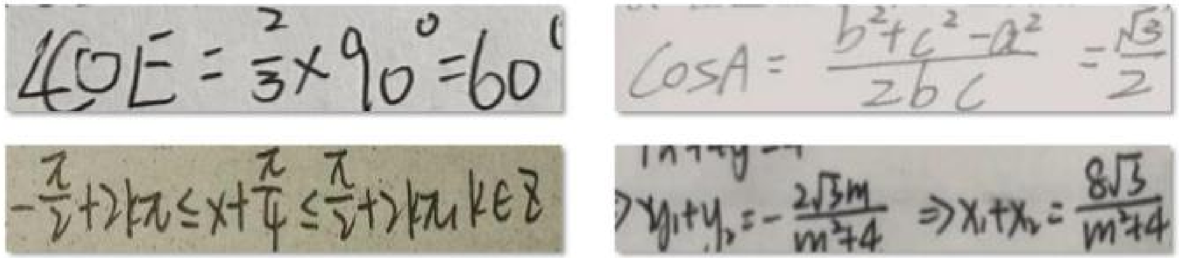
* Equal Contributions.

{fgao, nnwang}@xidian.edu.cn

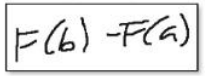
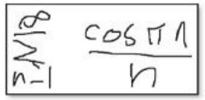
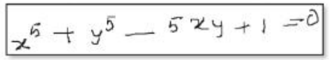
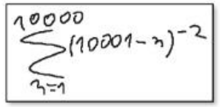
Introduction

Introduction

- **Handwritten Mathematical Expressions (HMEs)**
  - Complex structures
  - Serious deformations
  - Diverse writing styles



- **HME Recognition (HMER) a grand challenge in the OCR community.**

Input Image	DWAP (Baseline)	CAN-DWAP (Ours)
	\log g	\log
	F (b) - F (G)	F (b) - F (a)
	\sum _ { n = 1 } ^ { \infty } \frac { \cos \pi } { n }	\sum _ { n = 1 } ^ { \infty } \frac { \cos \pi n } { n }
	x ^ { 5 } + y ^ { 5 } - x y + 1 = 0	x ^ { 5 } + y ^ { 5 } - 5 x y + 1 = 0
	\sum _ { n = 1 } ^ { 1 0 0 0 0 0 0 0 0 1 - n) ^ { - 2 }	\sum _ { n = 1 } ^ { 1 0 0 0 0 } (1 0 0 0 1 - n) ^ { - 2 }

Method	CROHME 2014			CROHME 2016			CROHME 2019		
	ExpRate ↑	≤1 ↑	≤2 ↑	ExpRate ↑	≤1 ↑	≤2 ↑	ExpRate ↑	≤1 ↑	≤2 ↑
Without data augmentation									
DWAP-MSA [40]	52.80	68.10	72.00	50.10	63.80	67.40	47.70	59.50	63.30
WS-WAP [24]	53.65	–	–	51.96	64.34	70.10	–	–	–
MAN [28]*	54.05	68.76	72.21	50.56	64.78	67.13	–	–	–
BTTR [46]	53.96	66.02	70.28	52.31	63.90	68.61	52.96	65.97	69.14
ABM [1]	56.85	73.73	81.24	52.92	69.66	78.73	53.96	71.06	78.65
DWAP (baseline)†	51.48	67.01	73.30	50.65	63.30	70.88	50.04	65.39	69.39
CAN-DWAP (ours)	**57.00**	74.21	80.61	**56.06**	71.49	79.51	**54.88**	71.98	79.40
ABM (baseline)†	56.04	73.10	79.90	53.36	70.01	78.12	53.71	71.23	78.23
CAN-ABM (ours)	**57.26**	74.52	82.03	**56.15**	72.71	80.30	**55.96**	72.73	80.57

CAN Li B, Yuan Y, Liang D, et al. When counting meets HMER: counting-aware network for handwritten mathematical expression recognition[C]//European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2022: 197-214.

Introduction

- Synthetic Data Augmentation ✓

- No effective generative model for generating high-quality HME images ✘
  - Recomposion of real online HMEs
  - FormulaGAN: Image-to-Image (I2I) Translation

- Solution
  - HME Generation (HMEG) from symbol graphs of symbolilc sequences.
  - Graph-to-Image (G2I) generation.
    [J. Johnson, et al. CVPR'18]

[Truong et al. PRL'22]

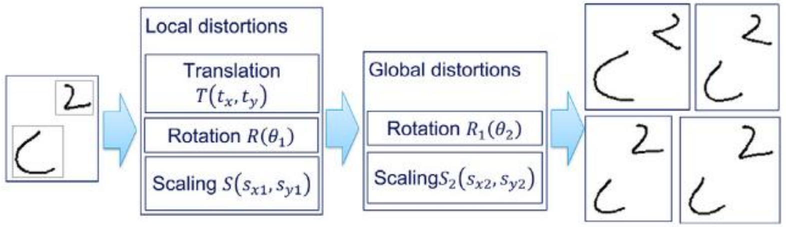


Fig. 2. Pattern augmentation using local and global distortions.

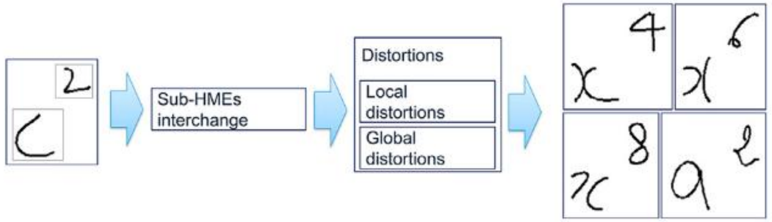
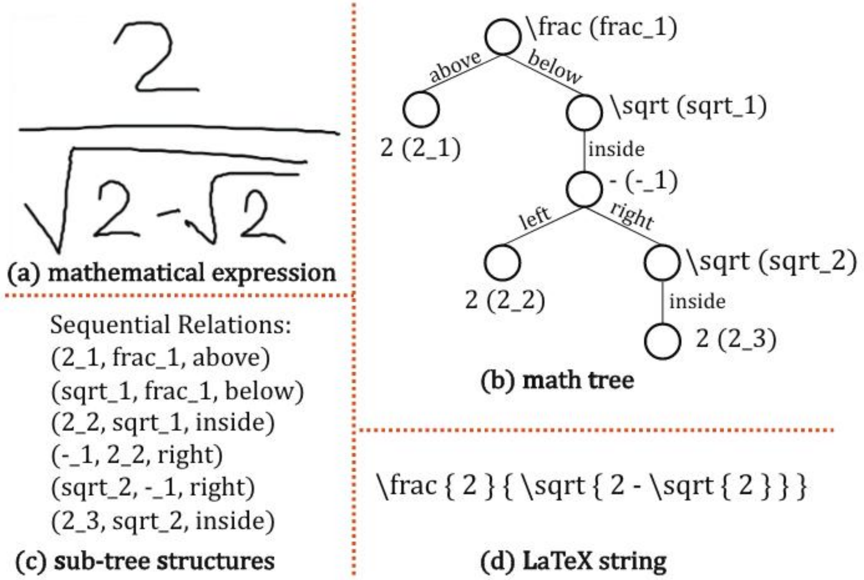


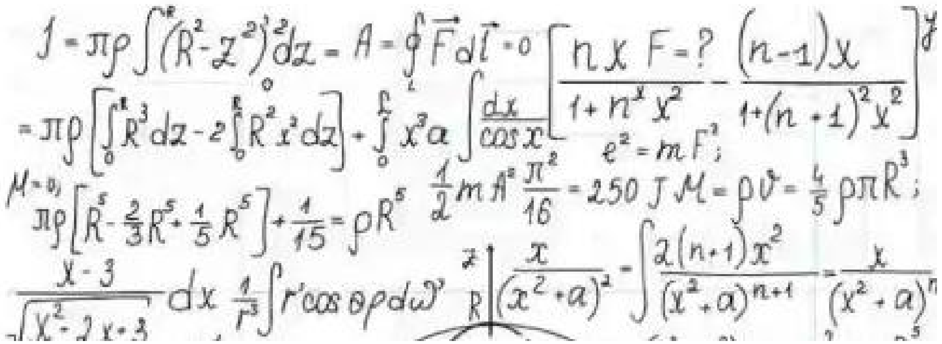
Fig. 3. Process of replacing sub-HMEs and applying the distortions.

[Zhang et al. TMM'20]



Challenges

- **Critical (unambiguous) layout clarity**
  - including the size of each symbol &
  - the positional relations between symbols
  - *e.g.* x^2, $x2$, x_2



- **MEs have infinite possible layouts and structures**
  - It's impossible to collect completed training data for learning the layout predictor.

- **Solutions:**
  - Less-is-More (LiM) training strategy
  - Sequential BBox-to-Mask Transformation (B2M)

a novel end-to-end pipeline of
graph → layout → mask → image
for G2I generation

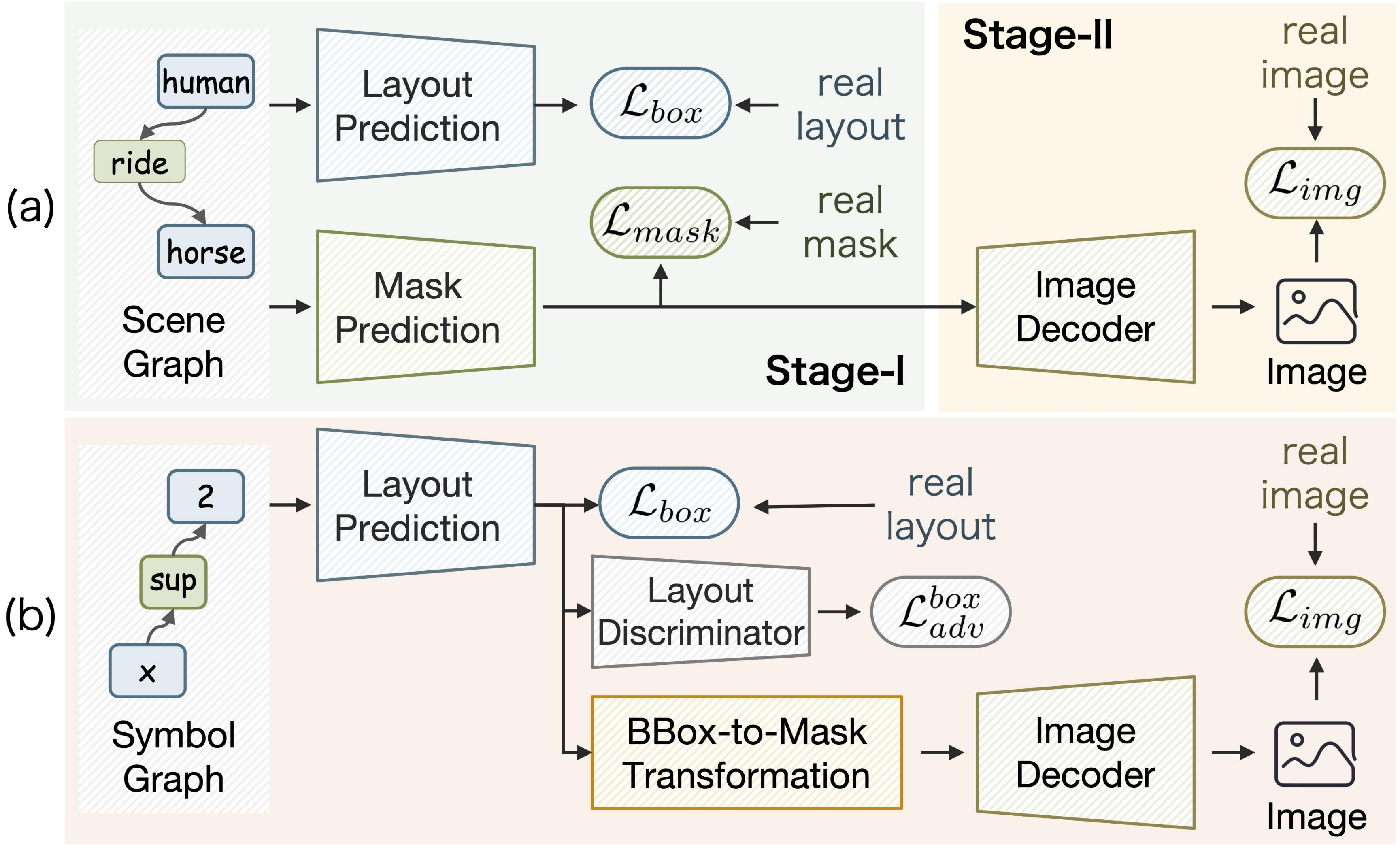
Solution

- ## Differences

Figure 1. Differences between (a) typical two-stage graph-to-image generation pipeline and (b) our end-to-end pipeline.

(a) In previous methods, the real masks are available as the input.

(b) In contrast, we propose a novel end-to-end pipeline of graph → layout → mask → image, and requires no real masks or non-differentiable alignment between layouts and masks.

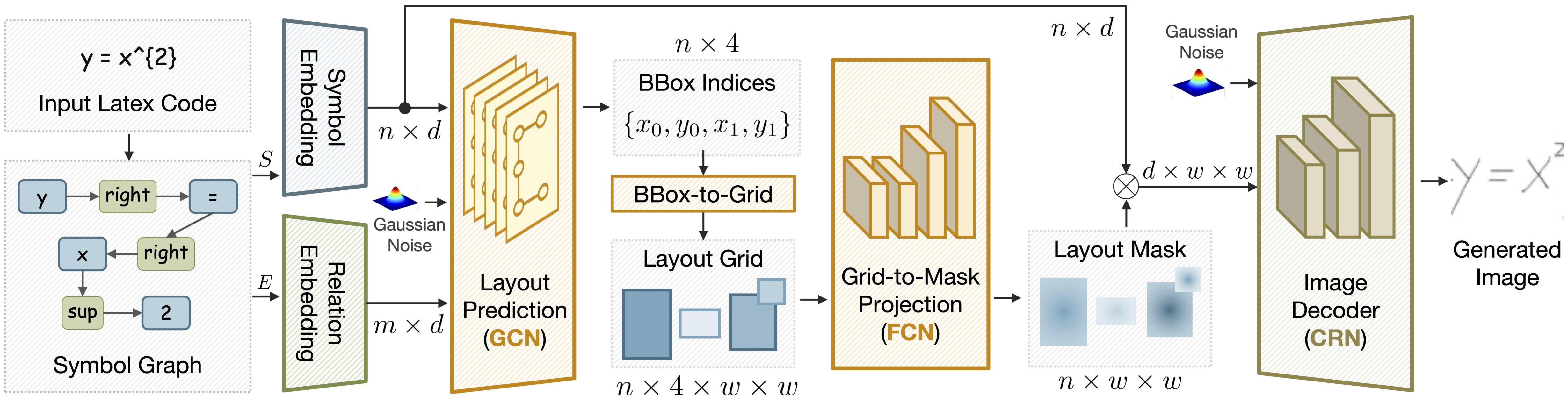


Proposed HMEG Method

Overview

- Pipeline

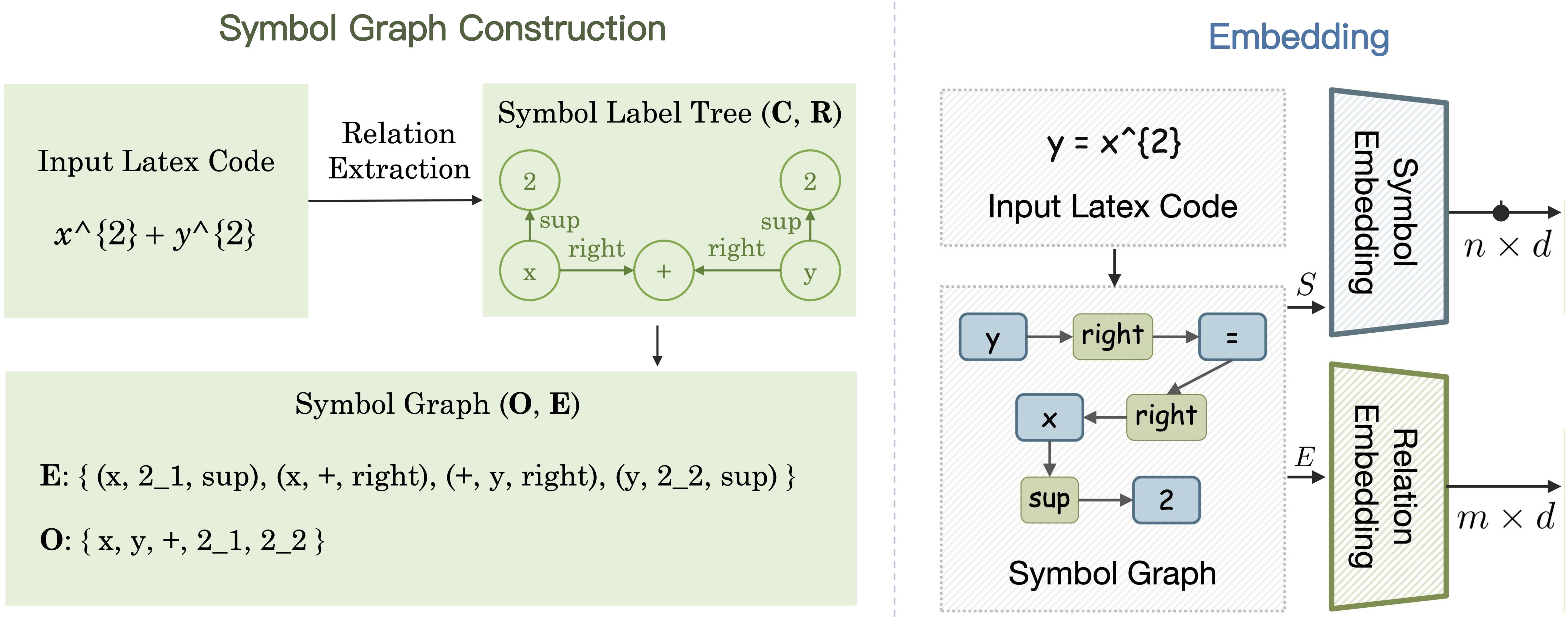
Graph Coinstruction → Layout Prediction → Mask Refinement → Image Generation



Overview of our handwritten mathematical expression generator (HMEG)

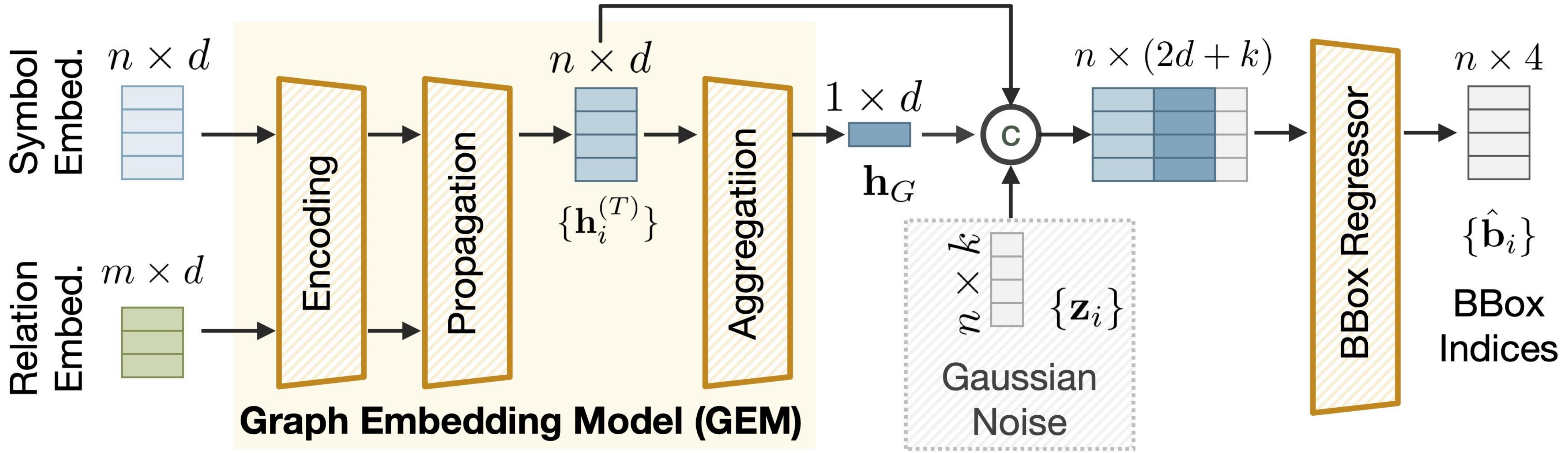
Symbol Graph Construction & Embedding

- The input LaTeX sequence is converted to a symbol graph, and embedded into high-dimensional feature vectors



Adversarial Layout Prediction

- Layout Predictor / Discriminator via GEM



$$\mathbf{h}_G = f_G\left(\sum_i \sigma(f_{\text{gate}}(\mathbf{h}_i^{(T)})) \odot f_{\text{update}}(\mathbf{h}_i^{(T)})\right),$$

$$\hat{\mathbf{b}}_i = f_{\text{bbox}}(\mathbf{h}_i^{(T)}, \mathbf{h}_G, \mathbf{z}_i), \forall i \in S.$$

Layout Discriminator

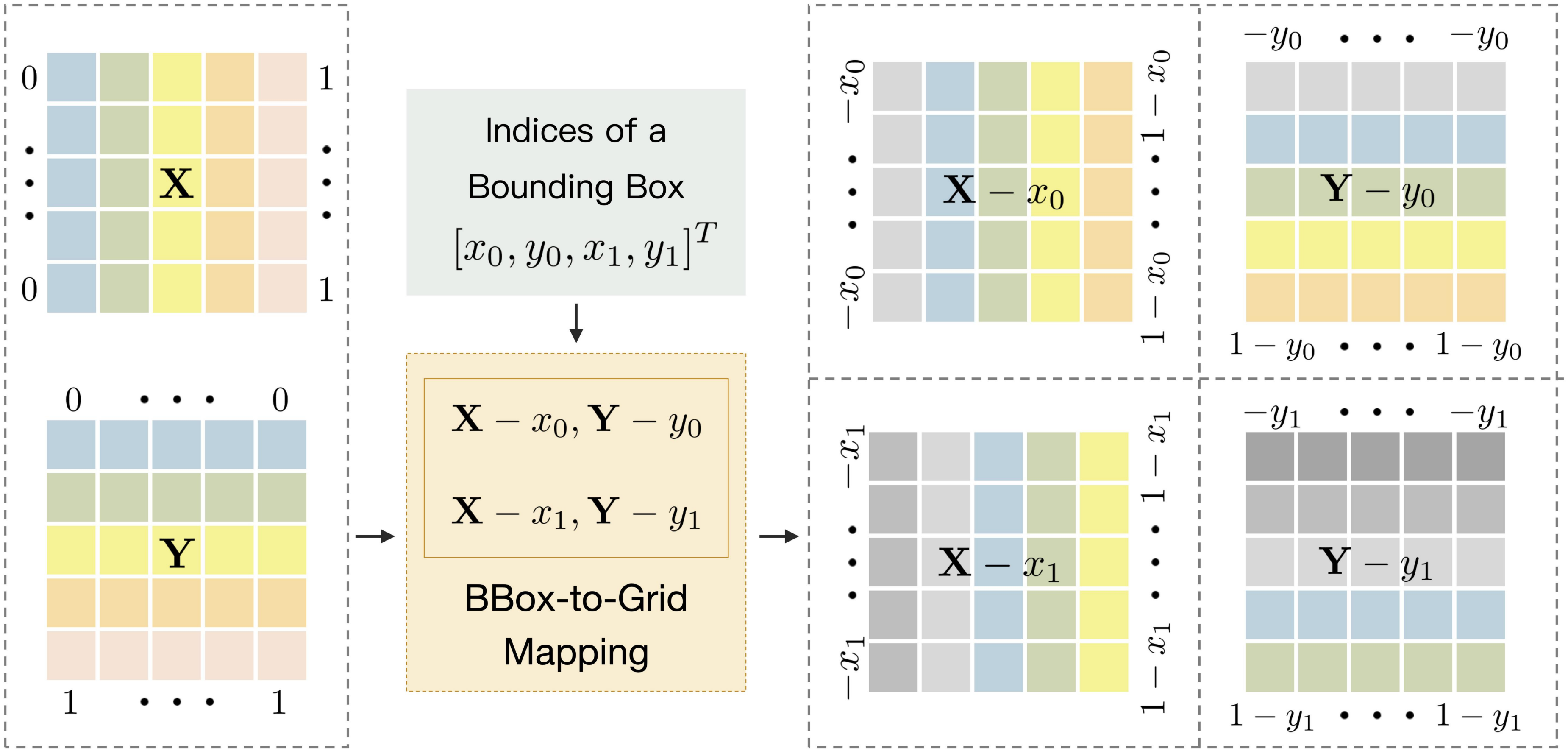
$$\mathcal{L}_{adv}^{box} = ||D_{box}(\mathbf{V}, B)||_2^2 + ||1 - D_{box}(\mathbf{V}, \hat{B})||_2^2.$$

Yujia Li, Chenjie Gu, Thomas Dullien, Oriol Vinyals, and Pushmeet Kohli. Graph matching networks for learning the similarity of graph structured objects. In International conference on machine learning, pages 3835 - 3845. PMLR, 2019.

Sequential BBox–to–Mask Transformation (B2M)

- (1) BBox–to–Grid Mapping.

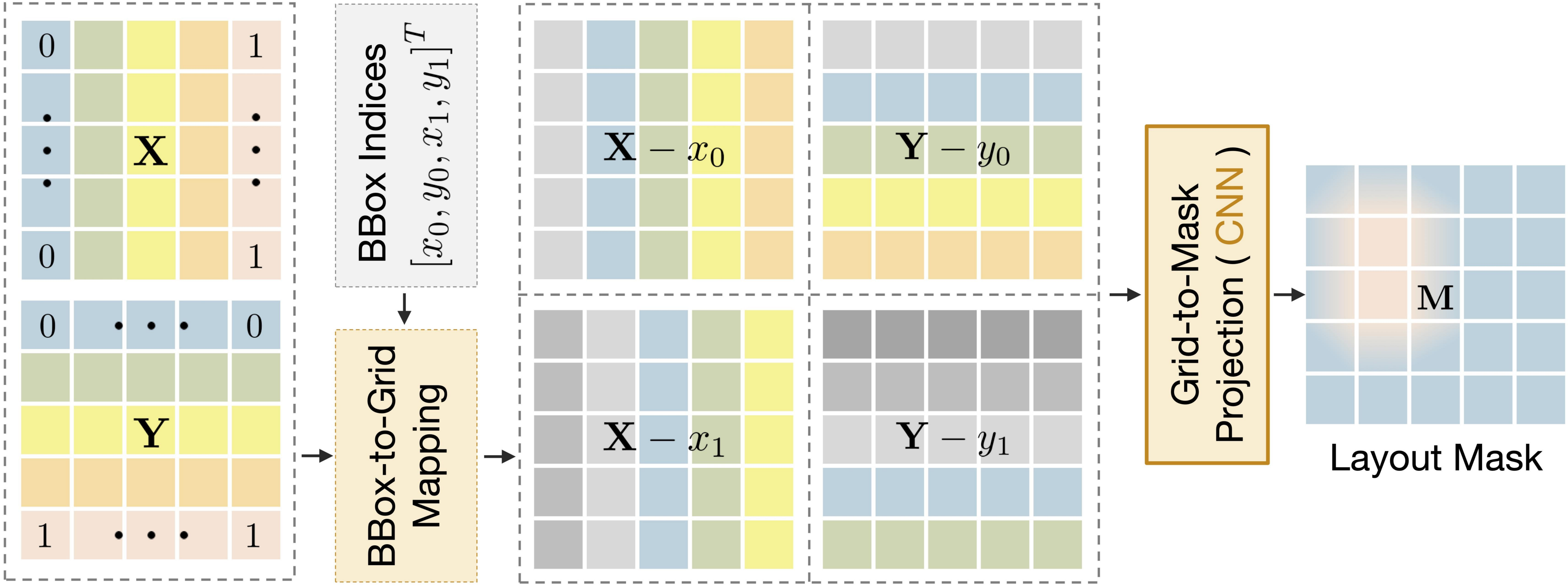
$$\mathbf{M}_i = \phi_{\text{mask}}\boxed{(\mathbf{X} - x_{i,0}, \mathbf{X} - x_{i,1}, \mathbf{Y} - y_{i,0}, \mathbf{Y} - y_{i,1})}$$



Sequential BBox-to-Mask Transformation (B2M)

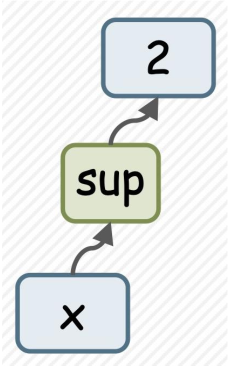
- (2) Grid-to-Mask Projection

$$\mathbf{M}_i = \boxed{\phi_{\text{mask}}}(\mathbf{X} - x_{i,0}, \mathbf{X} - x_{i,1}, \mathbf{Y} - y_{i,0}, \mathbf{Y} - y_{i,1}).$$



Less–is–More Learning (LiM) Strategy

- **Challenge**
  - arbitrary number of symbols or relations between symbols
  - a complete training set
  - the difficulty in learning an effective model



1–Degree Symbol Graph

- **Solution: Less–is–More (LiM) learning strategy**
  - **training:** merely use 1–degree symbol graphs
    - emphasis on local structures
  - **testing:** the learned GCN model can be applied to arbitrary symbol graphs, with arbitrary degrees of connections

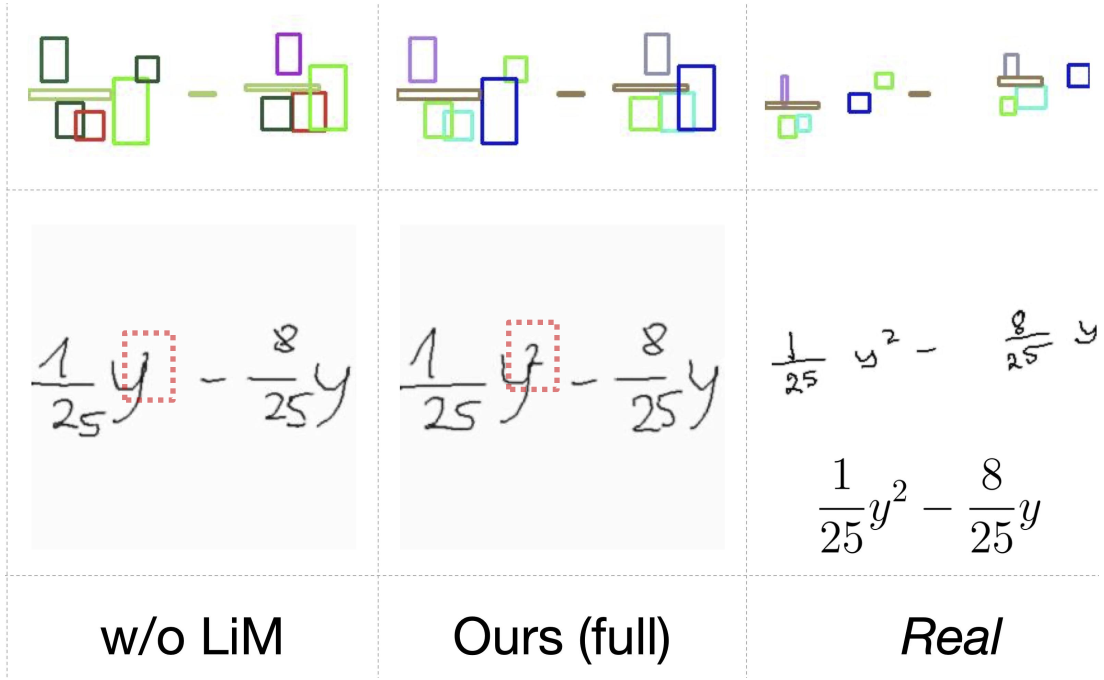


Image Generation

- **Image Decoder: CRN**
  - Three generated images, at the resolution of 64 × 64, 128 × 128, and 256 × 256, respec-tively. All these images are used to calculate the losses for optimizing our model.
- **Image and Symbol Discriminators**
  - Adverarial loss and symbol classification loss [AC-GAN, ICML'17]
- **Loss Functions**

$$\mathcal{L}_{img} = \lambda_1 \mathcal{L}_{pix} + \lambda_2 \mathcal{L}_{adv}^{img} + \lambda_3 \mathcal{L}_{adv}^{sym} + \lambda_4 \mathcal{L}_{aux}^{sym},$$

- *Pixel loss* $\mathcal{L}_{pix}$: First, we use the L1 distance between a generated image and the target formula image and as the pixel loss: $\mathcal{L}_{pix} = ||I - \hat{I}||_1$.
- *Image adversarial loss* $\mathcal{L}_{adv}^{img}$: Second, we use the global image discriminant loss from D_{img}, to encourage the generated HME images in the real handwritten style.
- *Symbol adversarial loss* $\mathcal{L}_{adv}^{sym}$: Third, we use the symbol discriminative loss from D_{sym}, to improve the fidelity of generated symbols.
- *Auxiliary Symbol Recognition Loss* $\mathcal{L}_{aux}^{sym}$: Finally, we use the auxiliary symbol recognition loss from D_{sym}, to en-force the generator producing recognizable symbols.

Experiments

Settings

- Data: CROHME2014/2016/2019
  - 126 different symbol categories, and
  - 8 positional relations, i.e.

    start, left superscript, superscript, subscript, below, above, right, end

- Implementation Details.
  - 1st stage – layout prediction: a batch size 64 and a learning rate 5e5, for 40,000 iterations.
  - 2nd stage – mask refinement & image decoding: a batch size 8, a learning rate 1e4, and train for 600,000 iterations.
  - Our codes are implemented by using Pytorch and a NVIDIA TITAN XP GPU.
  - We use Adam as the optimizer for all networks.

Comparison with SOTAs

Qualitative Comparison

- realistic handwritten styles
- clear strokes

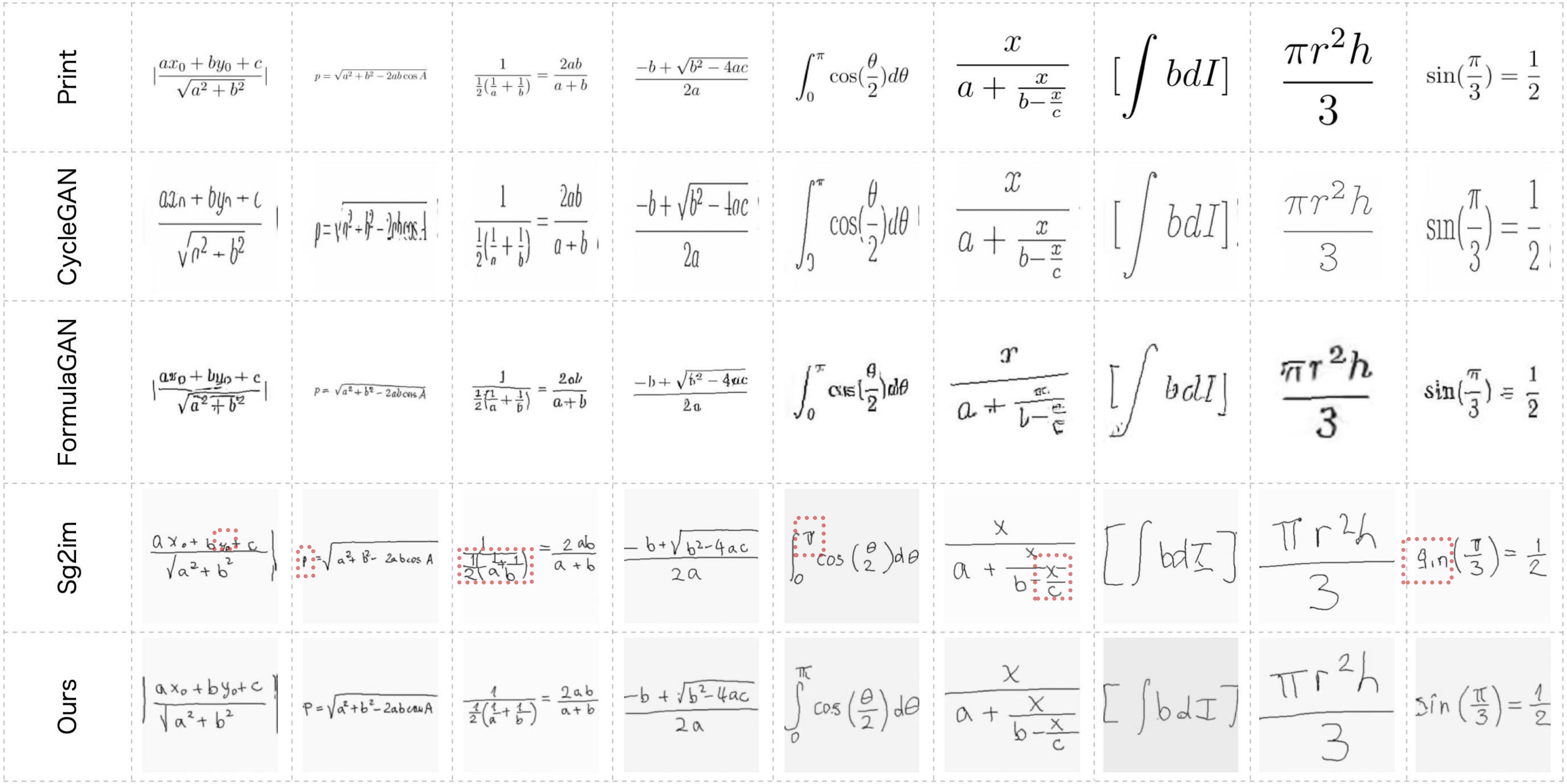


Figure 5. Handwritten mathematical expressions generated by CycleGAN [71], FormulaGAN [41], Sg2im [24], and our method.

Comparison with SOTAs

• Quantitative Comparison

Table 1. Comparison with existing methods on CHROME2019.

	SSIM ↑	FID ↓	WER ↓	ExpRate ↑
CycleGAN [71]	0.757	84.14	0.671	0.026
FormulaGAN [41]	0.724	74.68	0.601	0.066
Sg2im [24]	<u>0.787</u>	**10.02**	<u>0.393</u>	<u>0.219</u>
Ours	**0.793**	<u>10.98</u>	**0.326**	**0.316**

The handwritten expressions generated by our method, are significantly better than Sg2im, in terms of clarity and structure

Comparison with SOTAs

● Diffusion Models

- These diffusion models cannot generate high–quality HMEs with realistic handwritten styles or recognizable symbols.
- Besides, their computational complexity is much heavier than ours.

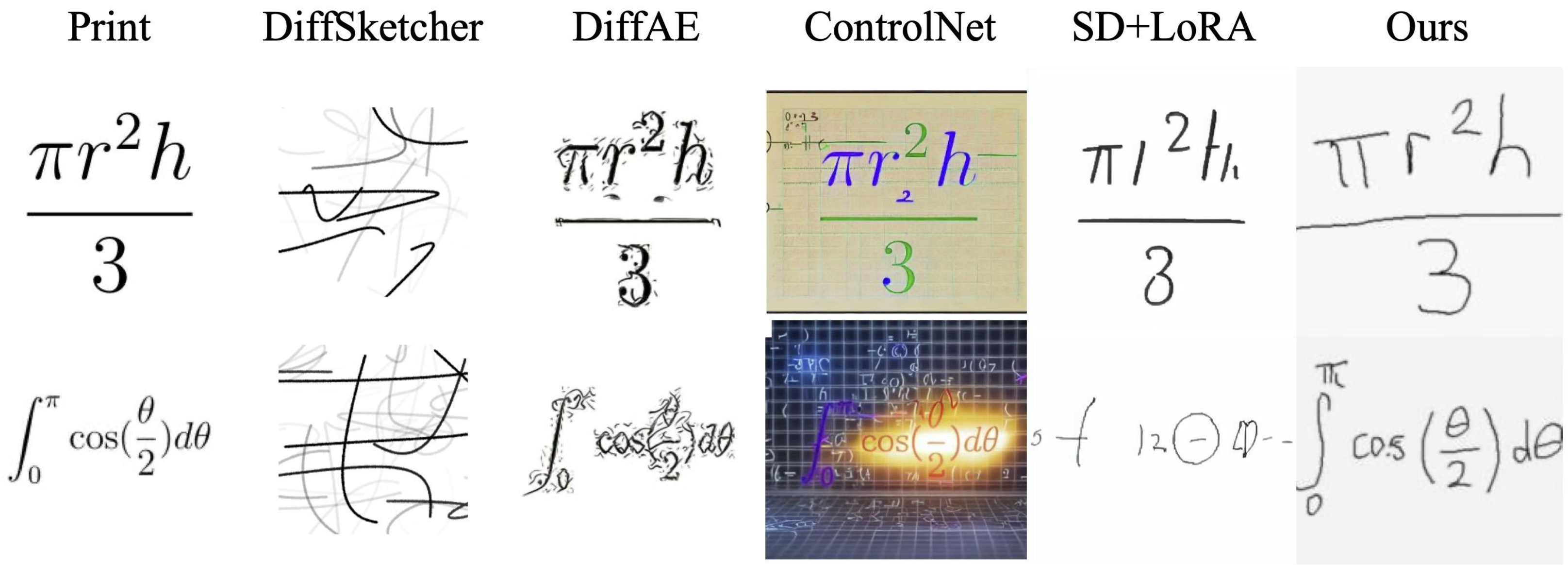


Figure 7. Comparison with diffusion models. DiffSketcher and DiffAE generate HMEs conditioned on print formulas; while ControleNet and SD+LoRA take both the print formula and a textual prompt ("*a handwritten mathematical expression of* `latex code`") as input. We use official diffusion models, and fine-tune DiffAE and SD+LoRA using HMEs.

Ablation Study

- Qualitative Results

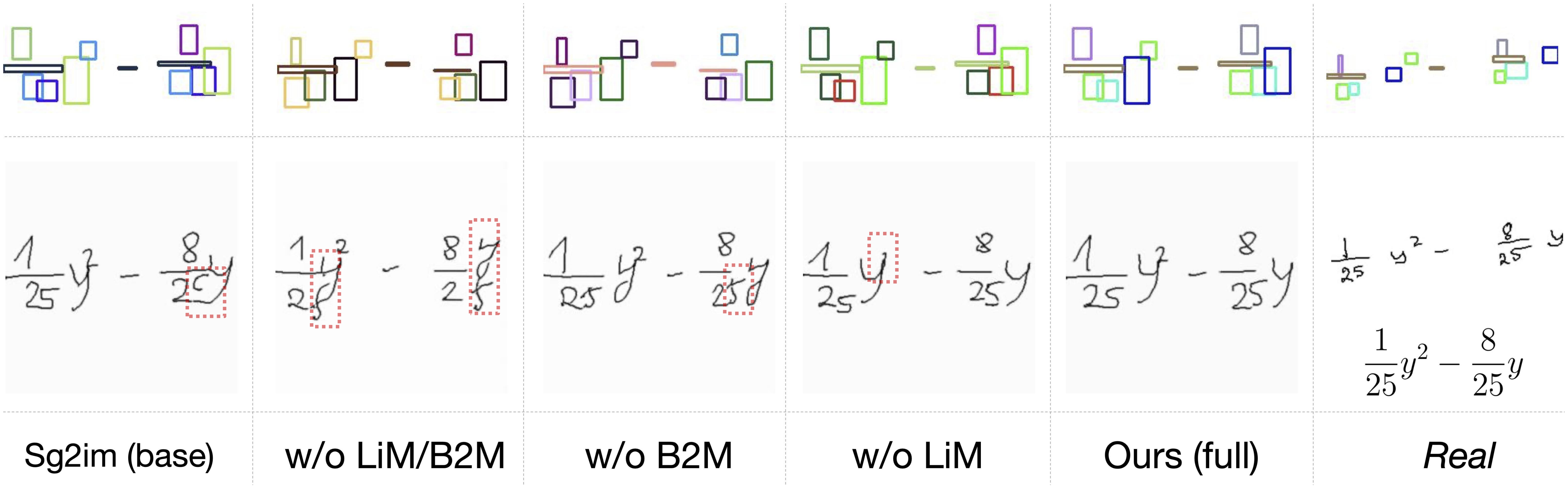


Figure 6. Illustration of the layouts and formula images generated by model variants, in the ablation study. The input formula is $\frac{1}{25}y^2 - \frac{8}{25}y$.

Ablation Study

- Quantitative Results

Table 2. Performance indices w.r.t. the ablation study on CHROME19. The model variant in the last row is our baseline.

	SSIM ↑	FID↓	WER↓	ExpRate ↑	mIOU ↑
Ours (full)	**0.793**	<u>10.98</u>	**0.326**	**0.316**	**0.364**
w/o LiM	<u>0.790</u>	11.55	<u>0.327</u>	<u>0.315</u>	0.335
w/o B2M	<u>0.790</u>	11.46	0.332	0.308	<u>0.354</u>
w/o LiM/B2M	0.786	13.30	0.365	0.279	0.338
Sg2im (base)	0.787	**10.08**	0.393	0.219	0.324

The LiM, B2M, and layout discriminator contribute significantly to the quality of layout prediction and image generation

Applications

• Mathematical Expression Manipulation

- edit in symbol graphs
- consistent chage with the editing operation.
- the manipulated images present
  - natural layouts,
  - recognizable symbols, and
  - clear strokes.

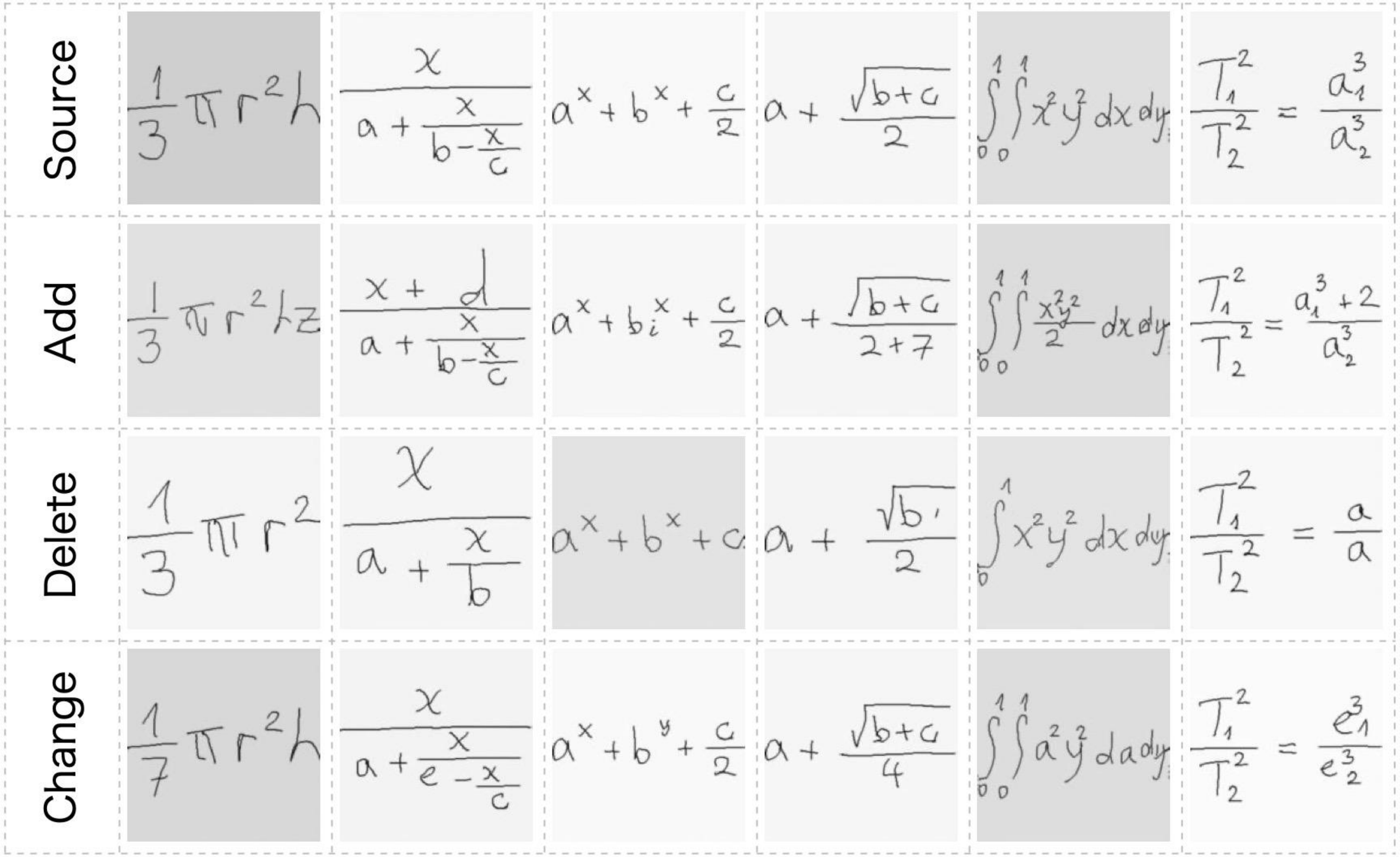


Figure 8. Illustration of expression manipulation using symbol graphs (LaTeX sequences).

Data Augmentation for HMER

- CROHME 2014/2016/2019
  - base HMER model: CAN [Li et al., ECCV'22]
  - 6000 additional generated samples, train for 120 epochs

Table 3. Results on CROHME14/16/19 testing sets and our generated images. *, †, *† denote using previous data augmentation, our synthetic data augmentation, and both, respectively.

	HMEG (*generated*)				**HMER** (*real*)			
	SSIM	CAN*	CAN†	StruRate	CAN	CAN*	CAN†	CAN*†
CROHME14	0.789	52.1	54.1	98.1	44.7	<u>52.9</u>	50.2	**55.4**
CROHME16	0.798	51.4	55.8	96.5	42.8	52.4	<u>53.9</u>	**57.6**
CROHME19	0.793	55.4	56.4	97.7	39.3	48.4	<u>49.6</u>	**58.5**

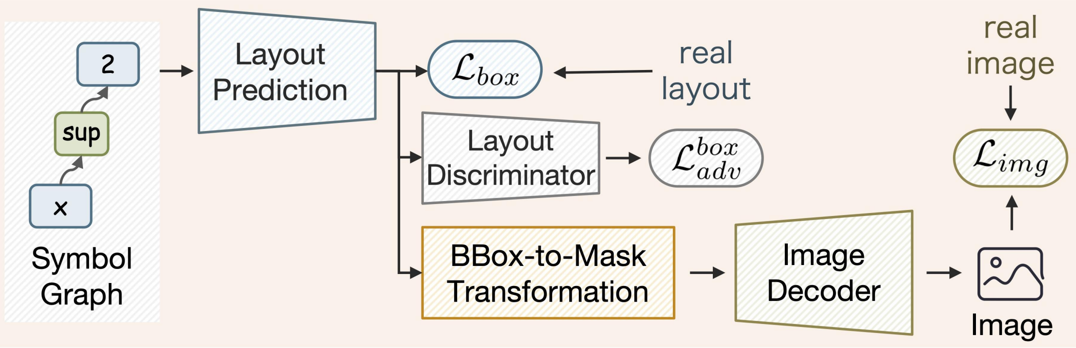
CAN Li B, Yuan Y, Liang D, et al. When counting meets HMER: counting-aware network for handwritten mathematical expression recognition[C]//European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2022: 197-214.

Conclusions

Conclusions

Summary

- Handwritten Mathematical Expression Generation (HMEG) from symbolic sequences
- end-to-end Graph-to-Image (G2I) generation
- Less-is-More (LiM) learning strategy
- differentiable layout refinement module
- https://github.com/AiArt-HDU/HMEG



Future Work

- boost the generation quality by via advanced networks (e.g. VQ-GAN or diffusion models)
- extend the proposed techniques to natural/artistic image generation
- boost the HMER by using {LaTeX, predicted layout, generated image} as pseudo labeled data

Generating Handwritten Mathematical Expressions From Symbol Graphs: An End-to-End Pipeline

THANKS

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

* Equal Contributions.

{fgao, nnwang}@xidian.edu.cn

GitHub