Supplementary Material
Towards Flexible Multi-modal Document Models

We describe more experimental details and results that are omitted due to the limited space in the main paper. Note that images are resized to meet the size limit. We will plan to provide high-resolution results in our project page.

A. Implementation details

A.1. Dataset

We show the details of multi-modal attributes used in our experiments for Rico and Crello datasets in Tab. 1. The number of elements for Rico and Crello is shown in Fig. 1a and Fig. 1b, respectively. The average number of elements in Rico is 16.2, which is much higher compared to the benchmark setup in the literature [4] (5.9 on average, 9 at maximum). The average number of elements in Crello is 9.7. However, design tasks in Crello dataset are also very challenging since the model should consider complex multi-modal relations such as images and texts. Crello dataset involves a large number of decorative elements, which also makes modeling hard.

A.2. Quantitative Evaluation

We describe implementation details for adapting existing task-specific models to our multi-task, multi-attribute, and arbitrary masking settings. Note that a learning schedule is similar in all the methods for a fair comparison.

A.2.1 LayoutGAN++

We follow the implementation described in LayoutGAN++ [4] and implement discriminator and auxiliary decoder modules. The generator part is precisely the same as our model. Training GANs on discrete data is non-trivial, as discussed in RelGAN [8]. It involves non-differentiable sampling operations on the multinomial distribution of the generator output. Following RelGAN, we use Straight Through Gumbel-Softmax trick [2] to enable the differentiable optimization. We set the number of Transformer layers to 4 and the size of their hidden states in each module to 256.

A.2.2 CVAE

Following LayoutVAE [3] and NDN [5], we implement a CVAE [10] variant for the multi-task, multi-attribute, and arbitrary masking settings. It predicts elements in an auto-regressive manner. At iteration j to predict j-th element, the condition c_j is modeled as:

$$t_j = \left(\{x^*_{i=1:j-1}\}, \{x_{i=j:S}\}\right), \tag{1}$$
$$c_j = g_{update}(t_j), \tag{2}$$

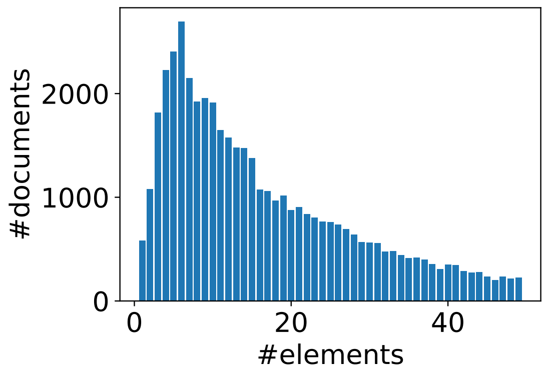
where x_j and x^*_j indicate a set of fields for the i-th element of the input and ground truth data, respectively. For g_{update}, we first use our Encoder $f^{\text{enc},k}$ to conduct element-wise aggregation and use our Transformer Blocks f^{trans} to obtain contextual information on each element for CVAE. We use an independent VAE module for each attribute:

$$z^k = g^{\text{enc},k}(x^k_j, c_j), \tag{3}$$
$$x^k_j = g^{\text{dec},k}(z^k, c_j), \tag{4}$$

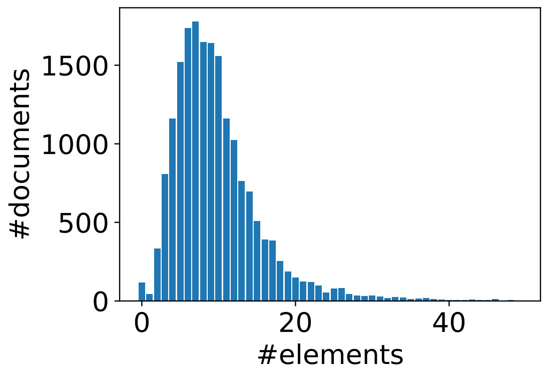
where $g^{\text{enc},k}$ and $g^{\text{dec},k}$ represent an encoder and a decoder consisting of fully connected layers for the k-th attribute. For $g^{\text{enc},k}$ and $g^{\text{dec},k}$, we follow NDN [5]. We train the model with conventional VAE loss consisting of a reconstruction loss $\mathcal{L}_{recon}$ and a KL loss $\mathcal{L}_{KL}$:

$$\mathcal{L}_{recon} = \sum_{(j,k)\in M} l^k(\hat{x}^k_j, {x^*}^k_j), \tag{5}$$
$$\mathcal{L}_{KL} = \sum_{(j,k)\in M} \mathbb{E}\left[D_{KL}(p(z^k|c_j, x^k_j)||p(z^k|c_j))\right] \tag{6}$$
$$\mathcal{L} = \mathcal{L}_{recon} + \lambda\mathcal{L}_{KL} \tag{7}$$

, where l^k is the loss function for attribute k and M is a set of tuples indicating the indices for masked tokens `[MASK]` in X and λ is a hyper-parameter. The model is trained with teacher forcing. At the test time, the model will use the estimated fields from previous steps (*i.e.*, $\hat{x}$ instead of x^* in Eq. (1)). The latent vector z is sampled from a conditional prior distribution $p(z|c_k)$, where p is a prior encoder.



(a) Rico dataset [1]



(b) Crello dataset [11]

Figure 1. The number of elements in vector graphic document datasets.

Table 1. Details of attributes for each element in vector graphic document used in our experiments. C and N is short for categorical and numerical, respectively.

Dataset	Group	Name	C/N	Size	Dim	Description
Crello	TYPE	Type	C	6	1	Element type, such as vector shape, image, or text.
	POS	Position	C	64	2	Left and top position each quantized to 64 bins.
	POS	Size	C	64	2	Width and height each quantized to 64 bins.
	ATTR	Opacity	C	8	1	Opacity quantized to 8 bins.
	ATTR	Color	C	16	3	RGB color each quantized to 16 bins (Only: text and svg fill)
	ATTR	Font	C	35	1	Font used to render texts. (Only: text)
	IMG	Image	N	1	768	Image feature extracted by pre-trained CLIP [9]. (Only: image)
	TXT	Text	N	1	768	Text feature extracted by pre-trained CLIP [9]. (Only: text)
Rico	TYPE	Type	C	13	1	Element type, such as text, image, icon, etc.
	POS	Position	C	64	2	Left and top position each quantized to 64 bins.
	POS	Size	C	64	2	Width and height each quantized to 64 bins.
	ATTR	Icon	C	59	1	Icon type, such as *arrow*, *close*, *home*.
	ATTR	Button	C	25	1	Text on button, such as *login* or *back*.
	ATTR	Clickable	C	2	1	Binary flag indicating if the element is clickable.

A.3. Comparison with Task-specific Baselines

A.3.1 Layout Generation

Although our model primarily aims at comprehensive design tasks, we can apply FlexDM to the conditional layout generation [4, 5, 7] where we only have labels as the inputs. We perform experiments on three major datasets, Rico [1], PubLaynet [13], and Magazine [12], and their performance in terms of the FID score, maximum IoU, alignment, and overlap metrics is summarized in Tab. 2. Since each metric has some variations, we follow [4] for details. We can see that our model is comparable to the existing layout generation models.

A.3.2 Single Text Box Placement

We use Crello dataset [11] for comparison but carefully select the attributes for a fair comparison. We use `Type`, `Left`, `Width`, `Right`, `Height`, `Text`, `Image`, and `Aspect Ratio`. `Aspect Ratio` is automatically computed by `Height/Width`. We mask the position and size attributes for the target text element and predict them by our masking model. For applying Li *et al.* [6]'s model, we render all the elements except the target one to a single image and use their model. `Color`, `Font`, and `Opacity` attributes are not used.

A.4. Qualitative Evaluation

We demonstrate that our framework generalizes to handle various samples in many design tasks. The results of

Table 2. Quantitative comparison of conditional layout generation. The values of Alignment and Overlap are multiplied by 100× for visibility. For reference, the FID and Max. IoU computed between the validation and test data, and the Alignment and Overlap computed with the test data are shown as *real data*.

Dataset	Rico				PubLayNet				Magazine			
Model	FID ↓	Max. IoU ↑	Align. ↓	Overlap ↓	FID ↓	Max. IoU ↑	Align. ↓	Overlap ↓	FID ↓	Max. IoU ↑	Align. ↓	Overlap ↓
LayoutGAN-W [7]	162.75±0.28	0.30±0.00	0.71±0.00	174.11±0.22	195.38±0.46	0.21±0.00	1.21±0.01	138.77±0.21	159.20±0.87	0.12±0.00	$\underline{0.74}$±0.02	188.77±0.93
LayoutGAN-R [7]	52.01±0.62	0.24±0.00	1.13±0.04	69.37±0.66	100.24±0.61	0.24±0.00	0.82±0.01	45.64±0.32	100.66±0.35	0.16±0.00	1.90±0.02	111.85±1.44
NDN-none [5]	**13.76**±0.28	0.35±0.00	$\underline{0.56}$±0.03	**54.75**±0.29	$\underline{35.67}$±0.35	$\underline{0.31}$±0.00	0.35±0.01	**16.50**±0.29	$\underline{23.27}$±0.90	$\underline{0.22}$±0.00	1.05±0.03	**30.31**±0.77
LayoutGAN++ [4]	$\underline{14.43}$±0.13	$\underline{0.36}$±0.00	0.60±0.12	$\underline{59.85}$±0.59	**20.48**±0.29	**0.36**±0.00	$\underline{0.19}$±0.00	22.80±0.32	**13.35**±0.41	**0.26**±0.00	0.80±0.02	$\underline{32.40}$±0.89
Ours	26.02±0.11	**0.42**±0.00	**0.12**±0.00	84.11±0.34	60.70±0.37	0.25±0.00	**0.11**±0.00	$\underline{16.85}$±0.13	33.84±0.69	$\underline{0.22}$±0.00	**0.66**±0.01	67.70±0.49
Real data	4.47	0.65	0.26	50.58	9.54	0.53	0.04	0.22	12.13	0.35	0.43	25.64

our model (Ours-EXP-FT) in Crello dataset are shown in Fig. 2 (`ATTR`), Fig. 5 (`IMG`), Fig. 8 (`TXT`), Fig. 11 (`POS`), and Fig. 14 (`ELEM`). We list typical failure cases for each task in Fig. 3 (`ATTR`), Fig. 6 (`IMG`), Fig. 9 (`TXT`), Fig. 12 (`POS`), and Fig. 15 (`ELEM`). We also show results on randomly sampled examples for each task in Fig. 4 (`ATTR`), Fig. 7 (`IMG`), Fig. 10 (`TXT`), Fig. 13 (`POS`), and Fig. 16 (`ELEM`). We can see that the design tasks are very challenging but our model constitutes a strong baseline. Note that we retrieve some rare element properties that affect the final appearance from ground truth for visualization, such as rotation and text alignment info inside the element (*e.g.*, center). Modeling such very fine-grained features would be a promising direction for a fully controllable generation.

References

[1] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. Rico: A mobile app dataset for building data-driven design applications. In *UIST*, 2017. 2

[2] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *ICLR*, 2017. 1

[3] Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. LayoutVAE: Stochastic scene layout generation from a label set. In *CVPR*, 2019. 1

[4] Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Constrained graphic layout generation via latent optimization. In *ACM MM*, 2021. 1, 2, 3

[5] Hsin-Ying Lee, Weilong Yang, Lu Jiang, Madison Le, Irfan Essa, Haifeng Gong, and Ming-Hsuan Yang. Neural design network: Graphic layout generation with constraints. In *ECCV*, 2020. 1, 2, 3

[6] Chenhui Li, Peiying Zhang, and Changbo Wang. Harmonious textual layout generation over natural images via deep aesthetics learning. *IEEE TMM*, 2021. 2

[7] Jianan Li, Jimei Yang, Aaron Hertzmann, Jianming Zhang, and Tingfa Xu. LayoutGAN: Generating graphic layouts with wireframe discriminators. In *ICLR*, 2019. 2, 3

[8] Weili Nie, Nina Narodytska, and Ankit Patel. Relgan: Relational generative adversarial networks for text generation. In *ICLR*, 2019. 1

[9] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2

[10] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. 2015. 1

[11] Kota Yamaguchi. CanvasVAE: Learning to generate vector graphics documents. In *ICCV*, 2021. 2

[12] Xinru Zheng, Xiaotian Qiao, Ying Cao, and Rynson WH Lau. Content-aware generative modeling of graphic design layouts. *ACM TOG*, 38(4), 2019. 2

[13] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. PubLayNet: largest dataset ever for document layout analysis. In *ICDAR*, 2019. 2

Figure 2. Success cases in ATTR prediction. We visualize the target fields assigned `[MASK]` using fixed default values (*e.g.*, black for text color and gray for solid fill). Best viewed with zoom and color.

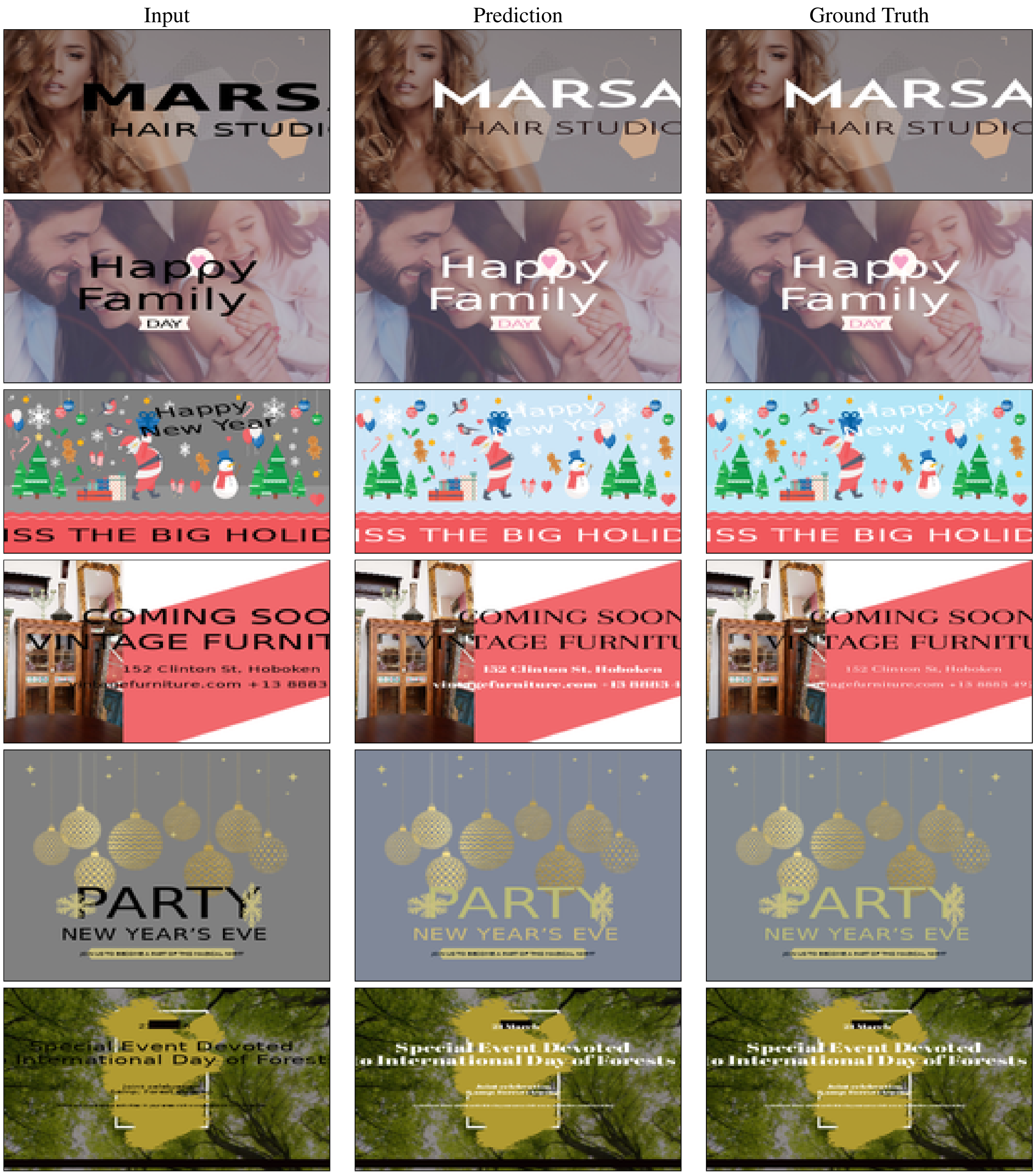
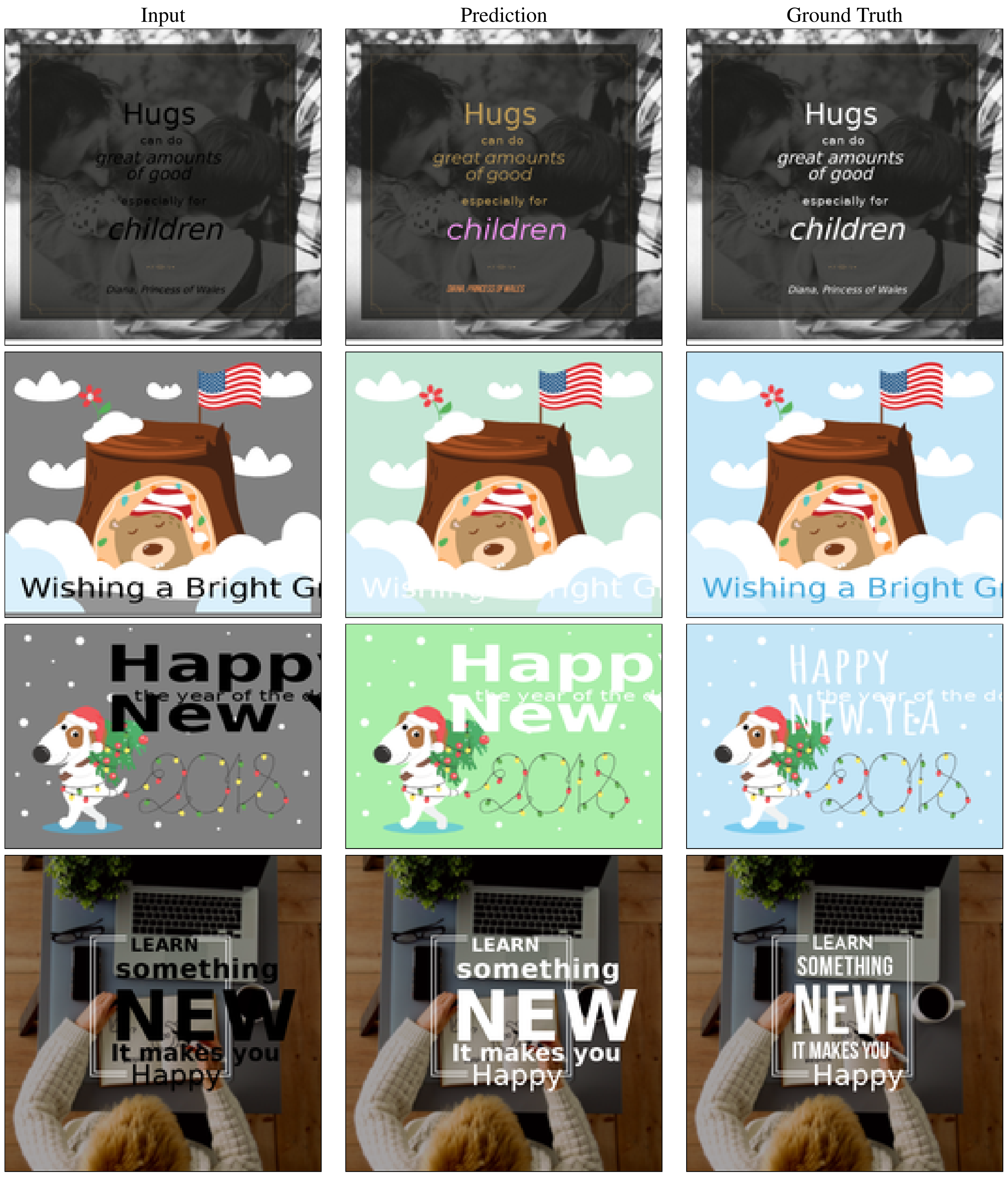


Figure 3. Failure cases in ATTR prediction. We visualize the target fields assigned [MASK] using fixed default values (*e.g.*, black for text color and gray for solid fill). First row: the model uses too many colors. Second row: text is invisible due to very weak contrast between the text and solid fill color. Third and fourth rows: the model does not pick appropriate fonts. Best viewed with zoom and color.



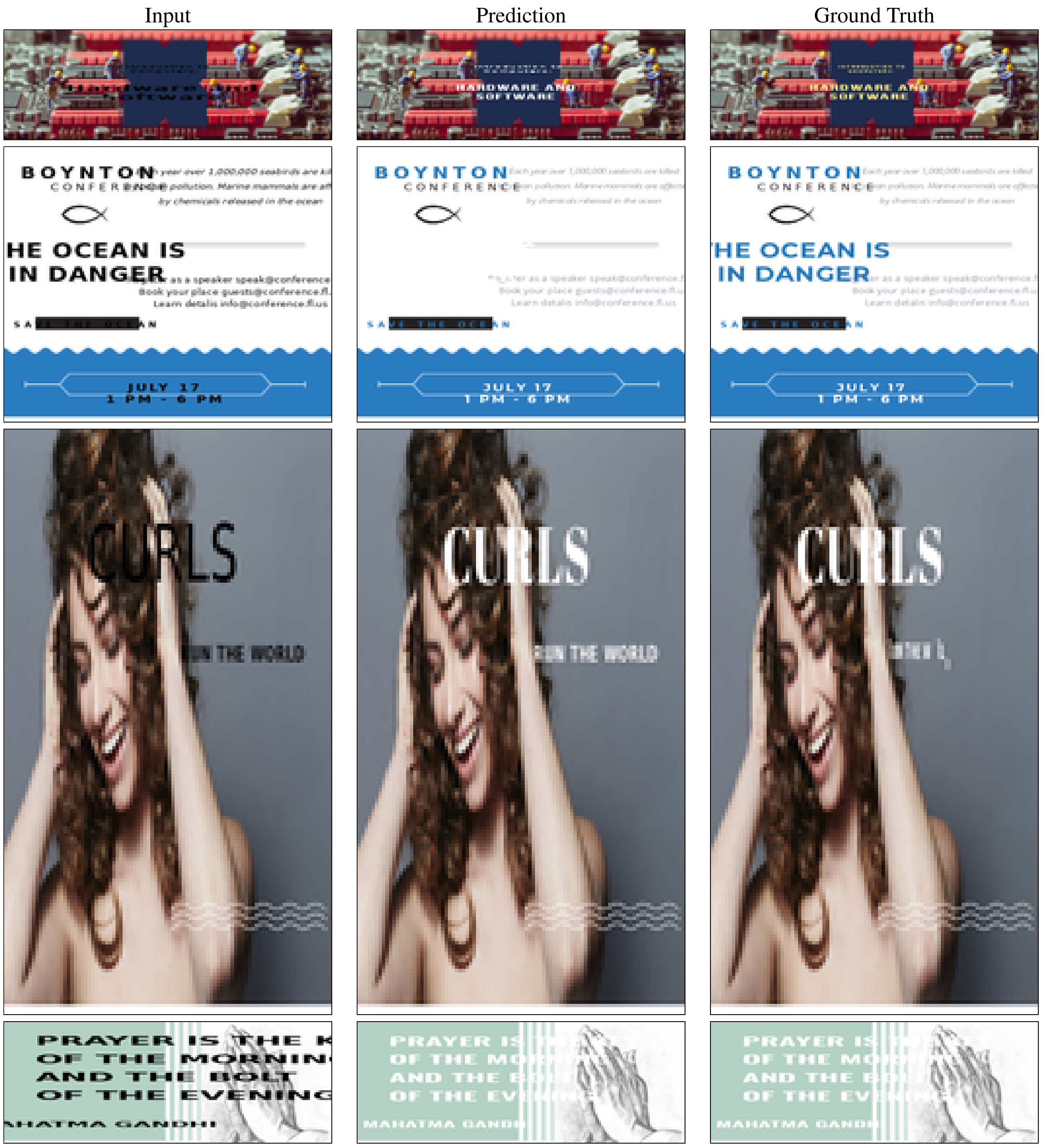


Figure 4. Randomly sampled examples in ATTR prediction. We visualize the target fields assigned `[MASK]` using fixed default values (*e.g.*, black for text color and gray for solid fill). Best viewed with zoom and color.

Figure 5. Success cases in IMG prediction. We visualize the target fields assigned `[MASK]` using fixed default values (*e.g.*, gray for image). Best viewed with zoom and color.

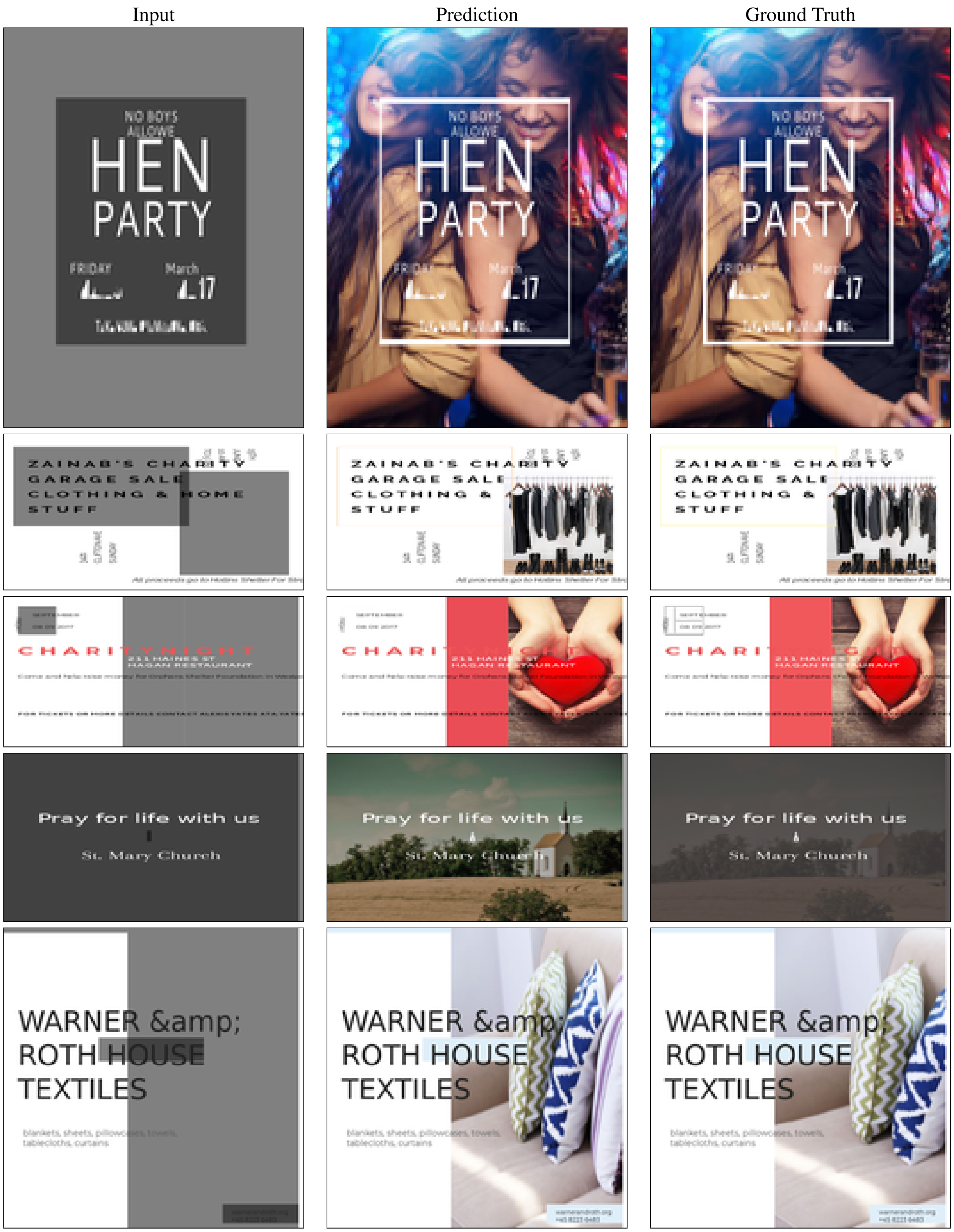

Input
Prediction
Ground Truth
NO BOYS ALLOWE
HEN PARTY
FRIDAY
March
17
ZAINAB'S CHARITY GARAGE SALE CLOTHING & HOME STUFF
CHARITY NIGHT
211 HAINES ST HAGAN RESTAURANT
Pray for life with us
St. Mary Church
WARNER &amp; ROTH HOUSE TEXTILES
blankets, sheets, pillowcases, towels, tablecloths, curtains

Figure 6. Failure cases in IMG prediction. We visualize the target fields assigned `[MASK]` using fixed default values (*e.g*., gray for image). First and second rows: the image does not match the texts. Third row: the model failed to infer symmetry. Fourth and fifth rows: predicting decorative elements are very challenging. Best viewed with zoom and color.

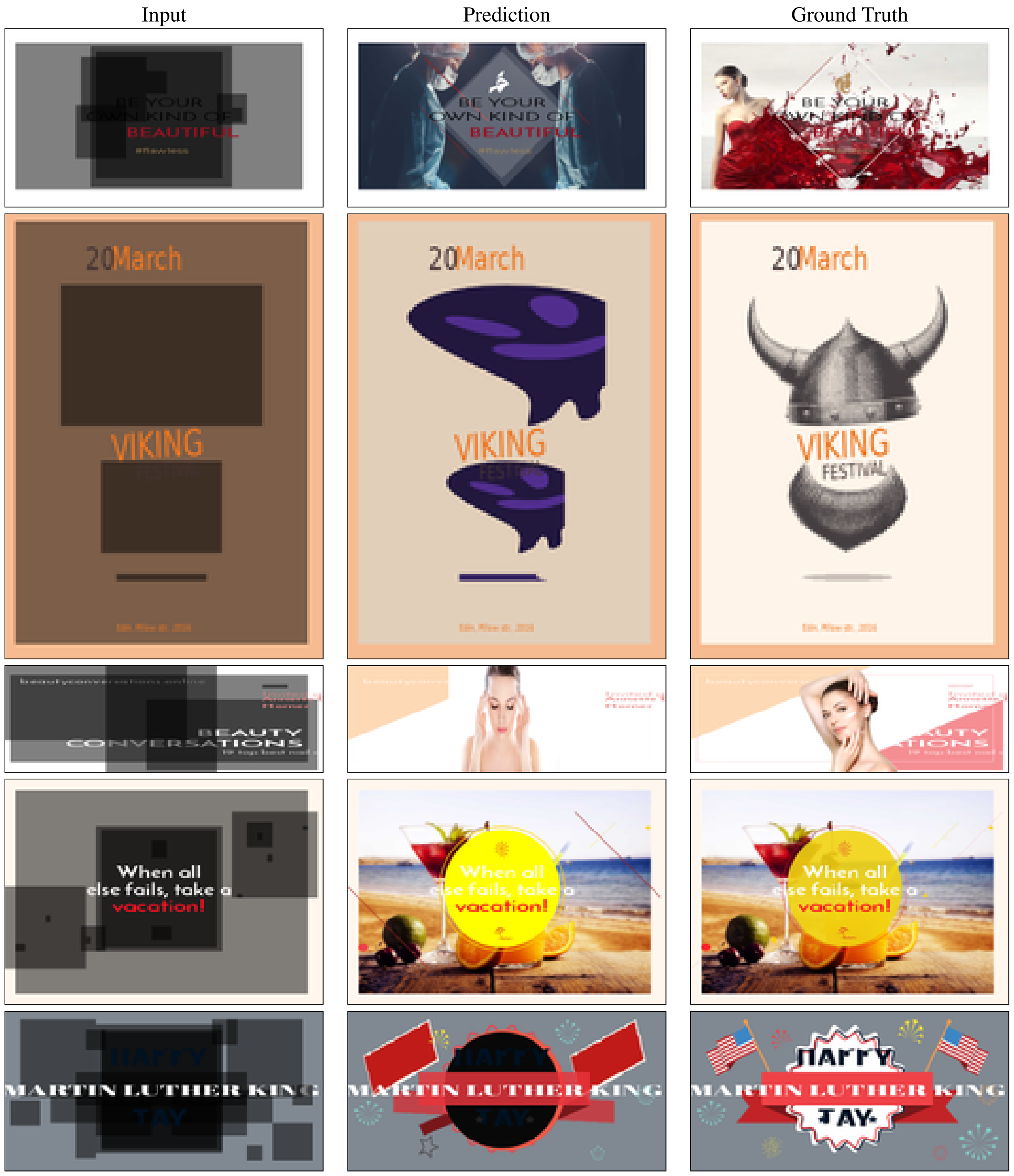


Figure 7. Randomly sampled examples in IMG prediction. We visualize the target fields assigned [MASK] using fixed default values (*e.g.*, gray for image). Best viewed with zoom and color.

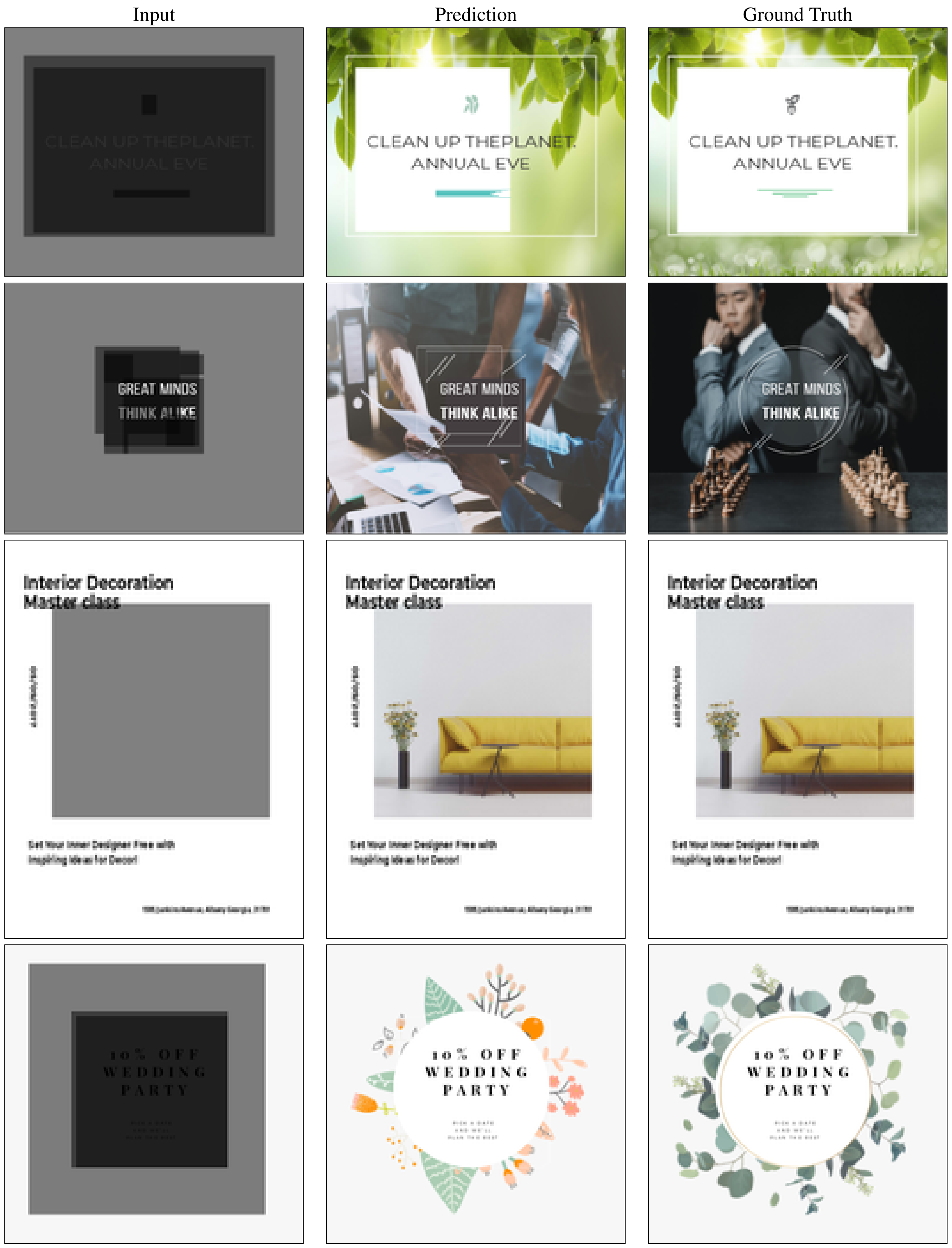


Figure 8. Success cases in TXT prediction. We visualize the target fields assigned [MASK] using fixed default values (*e.g.*, repeating 'TEXT' for text). Best viewed with zoom and color.

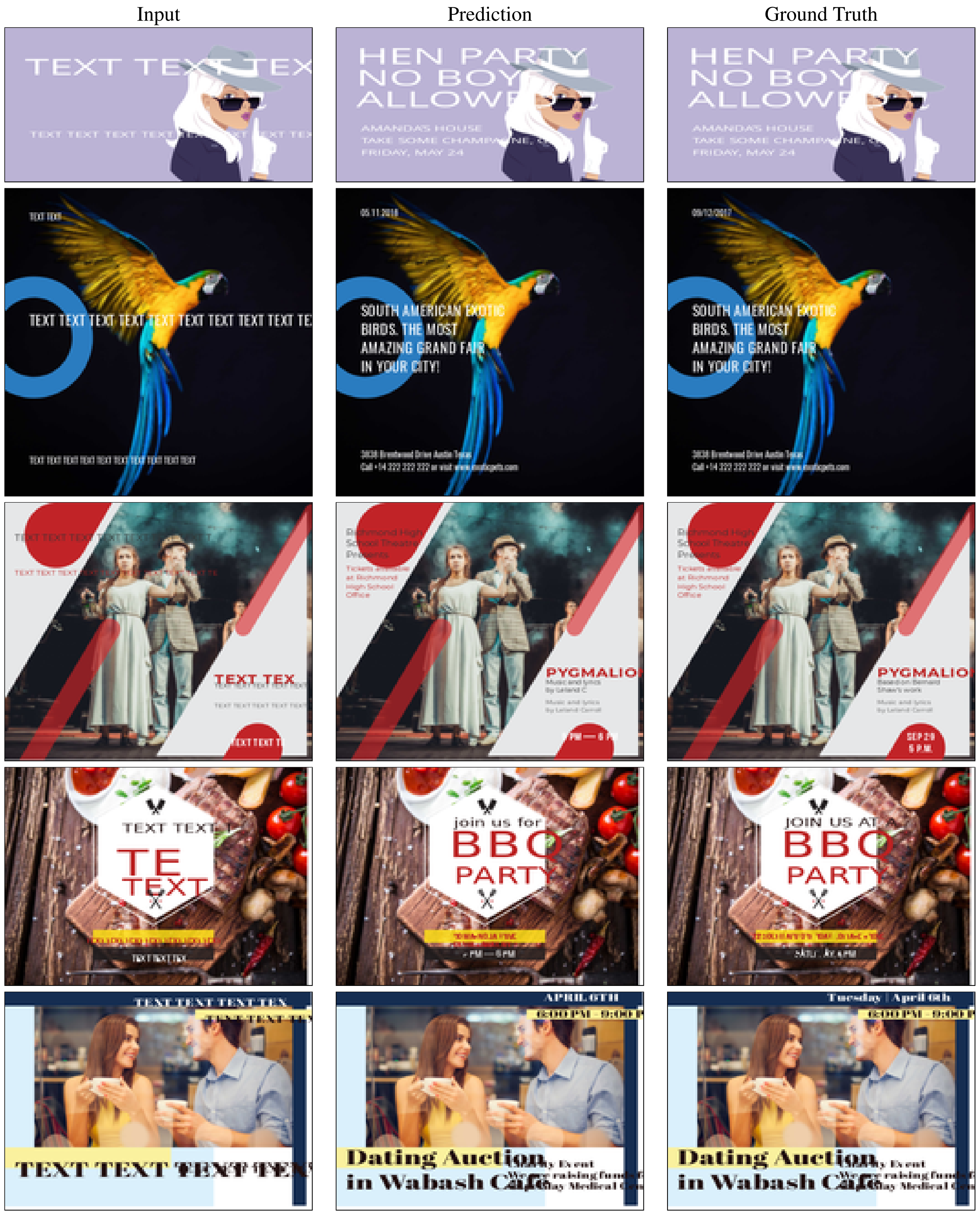


Figure 9. Failure cases in TXT prediction. We visualize the target fields assigned `[MASK]` using fixed default values (*e.g.*, repeating 'TEXT' for text). First row: the texts describe the background image a bit, but it does not match the ground truth. Second row: the model is unable to capture the context from the background images and decoration. Third row: when the model is not sure, it tends to predict features of some common words (e.g., 'than just a', 'this is', 'sale', …). Fourth row: infering complex messages behind the scene is difficult. Best viewed with zoom and color.

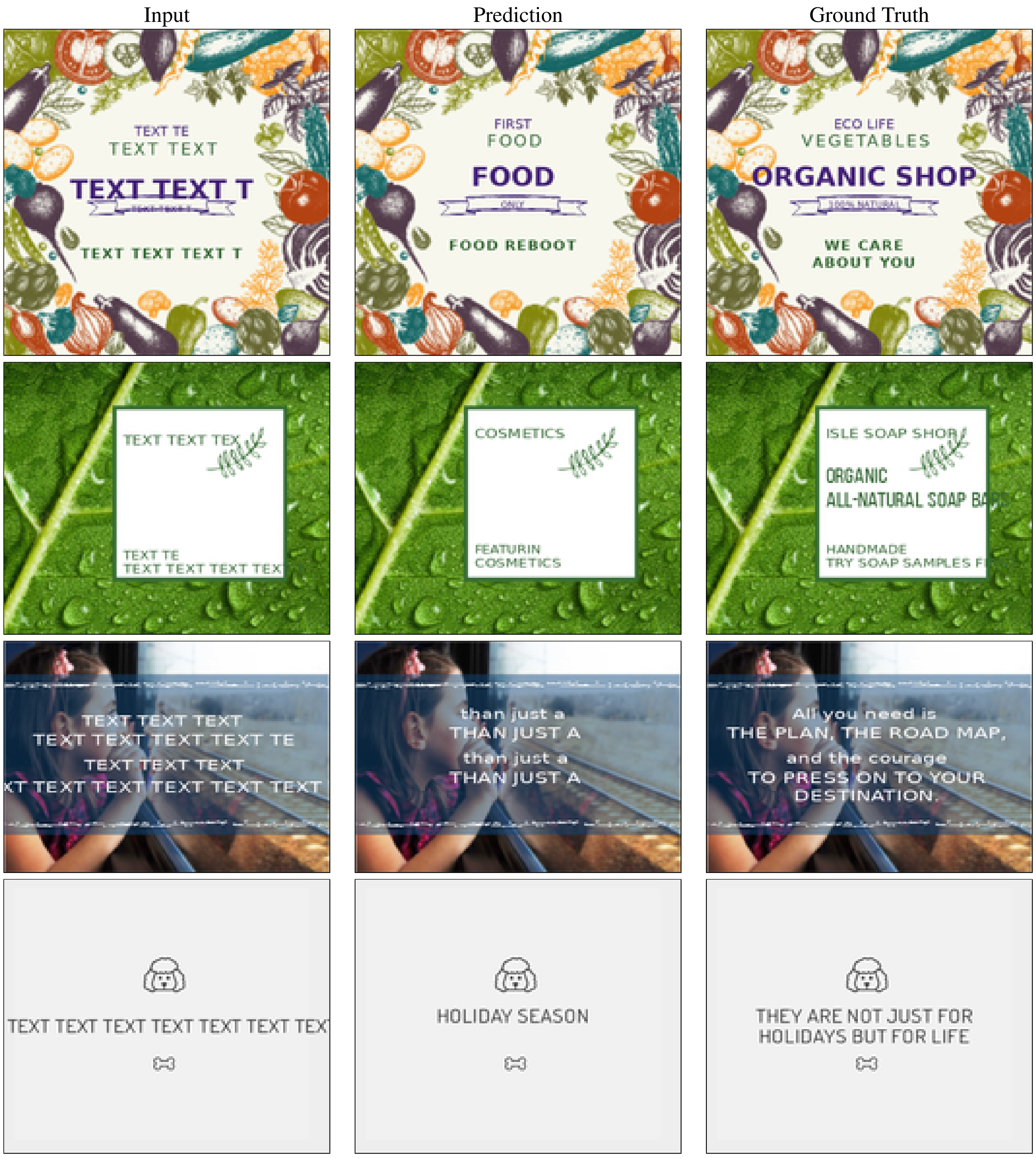


Figure 10. Randomly sampled examples in TXT prediction. We visualize the target fields assigned [MASK] using fixed default values (*e.g.*, repeating 'TEXT' for text). Best viewed with zoom and color.

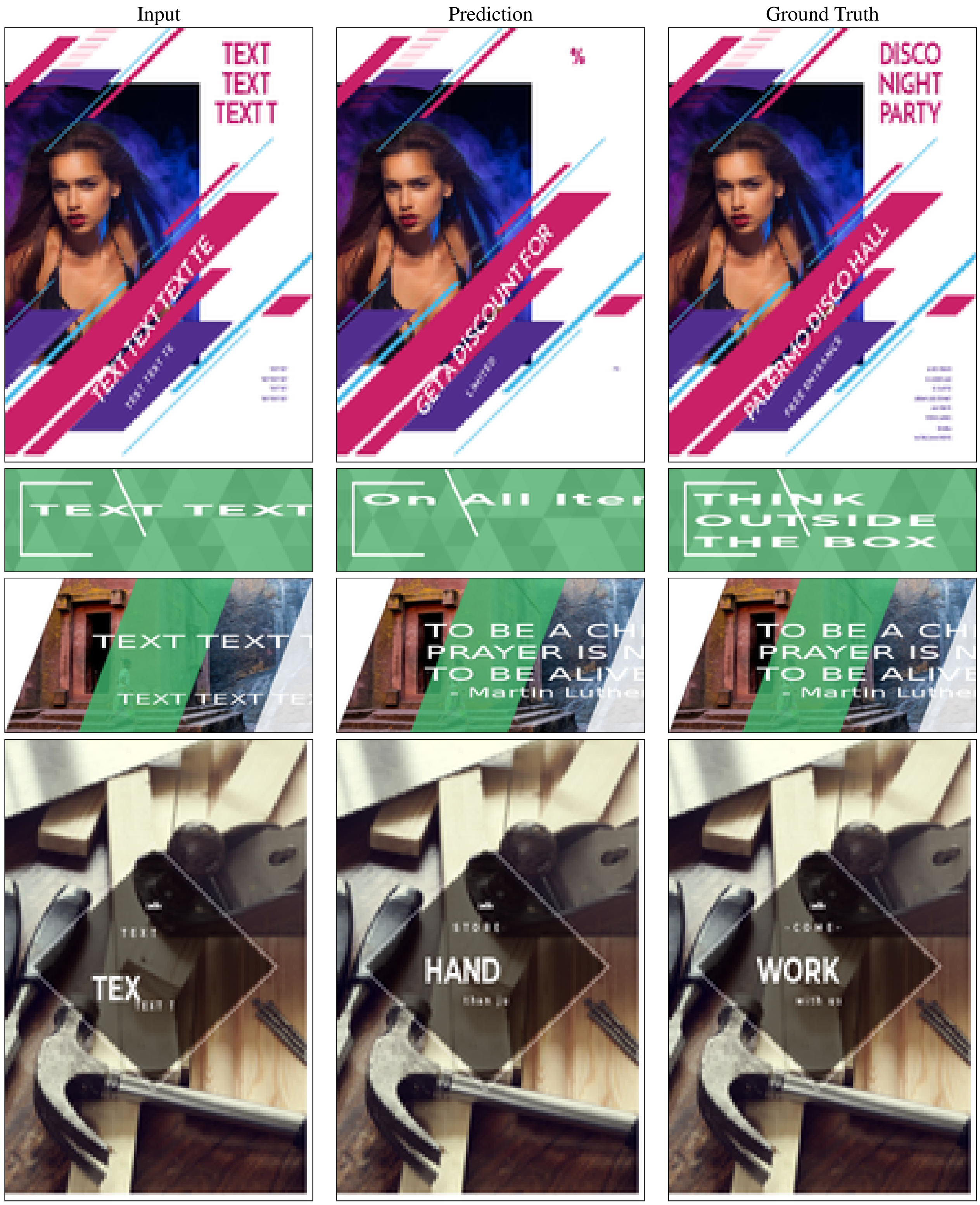
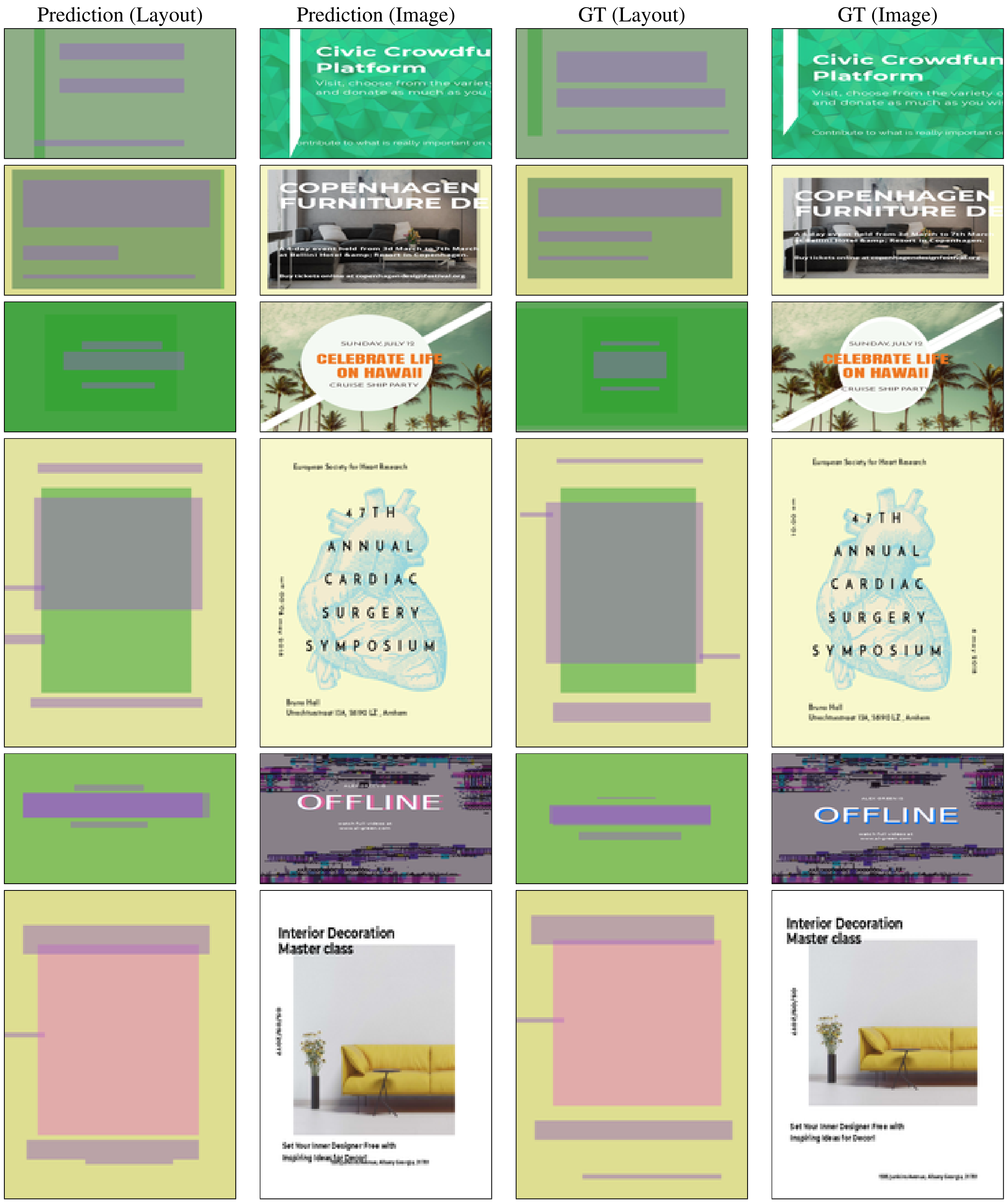


Figure 11. Success cases in POS prediction. The correspondence between the color and type of the element in the layout is as follows: green = *vector shape*, magenta = *image*, purple = *text*, yellow = *solid fill*. Best viewed with zoom and color.



Prediction (Layout)
Prediction (Image)
GT (Layout)
GT (Image)
Civic Crowdfunding Platform
COPENHAGEN FURNITURE DE
CELEBRATE LIFE ON HAWAII
SUNDAY, JULY 12
CRUISE SHIP PARTY
47TH ANNUAL CARDIAC SURGERY SYMPOSIUM
European Society for Heart Research
OFFLINE
Interior Decoration Master class
Set Your Inner Designer Free with Inspiring Ideas for Decor!

Figure 12. Failure cases in POS prediction. First row: text reading order is essential but the model fails to infer the order. Second row: the text is highly overlaping with the background image. Third and fourth rows: placing a large number of objects is very challenging. Fifth row: the model cannot capture view hierarchy. Sixth row: arranging decorative elements is difficult. The correspondence between the color and type of the element in the layout is as follows: green = *vector shape*, magenta = *image*, purple = *text*, yellow = *solid fill*. Best viewed with zoom and color.

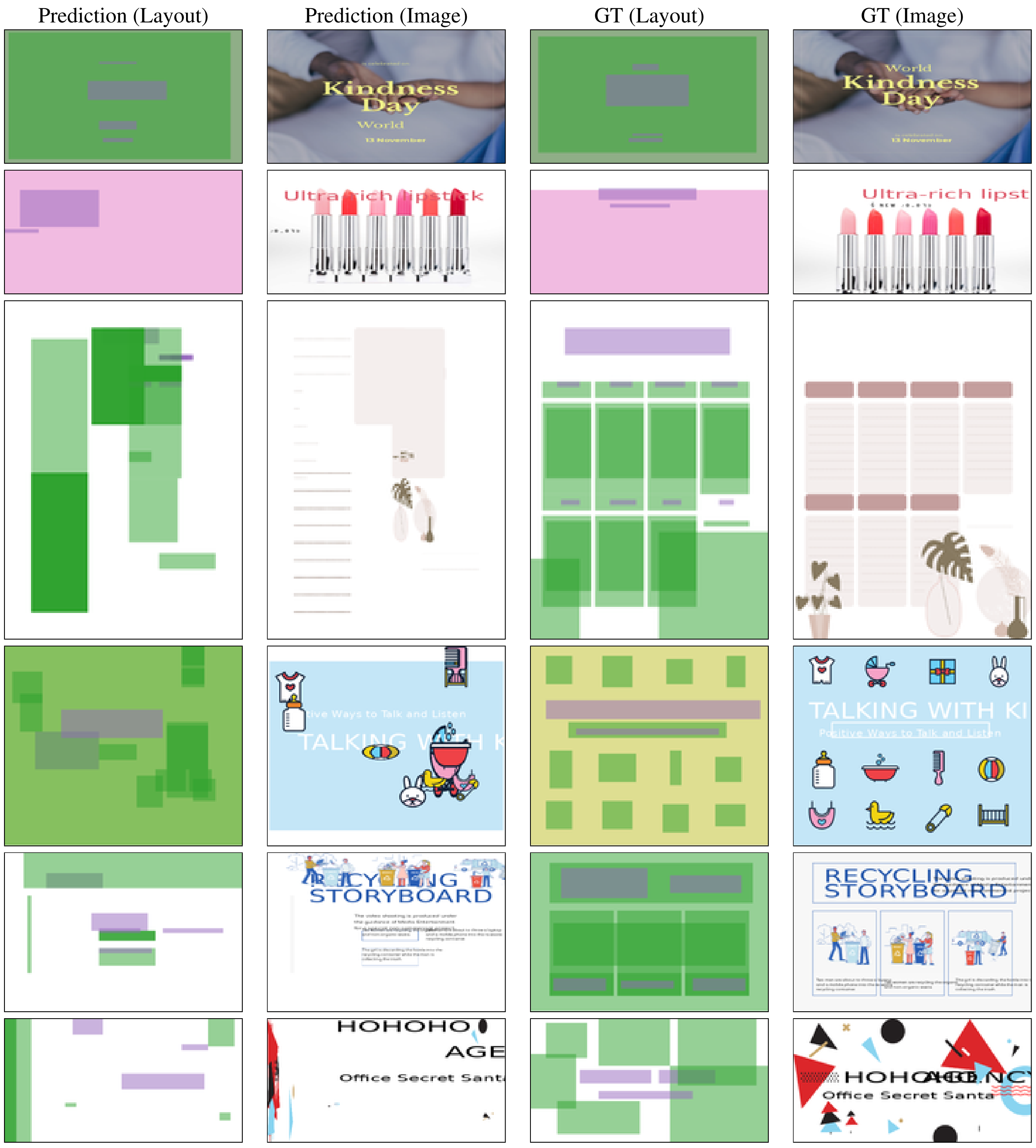


Figure 13. Randomly sampled examples in POS prediction. The correspondence between the color and type of the element in the layout is as follows: green = *vector shape*, magenta = *image*, purple = *text*, yellow = *solid fill*. Best viewed with zoom and color.

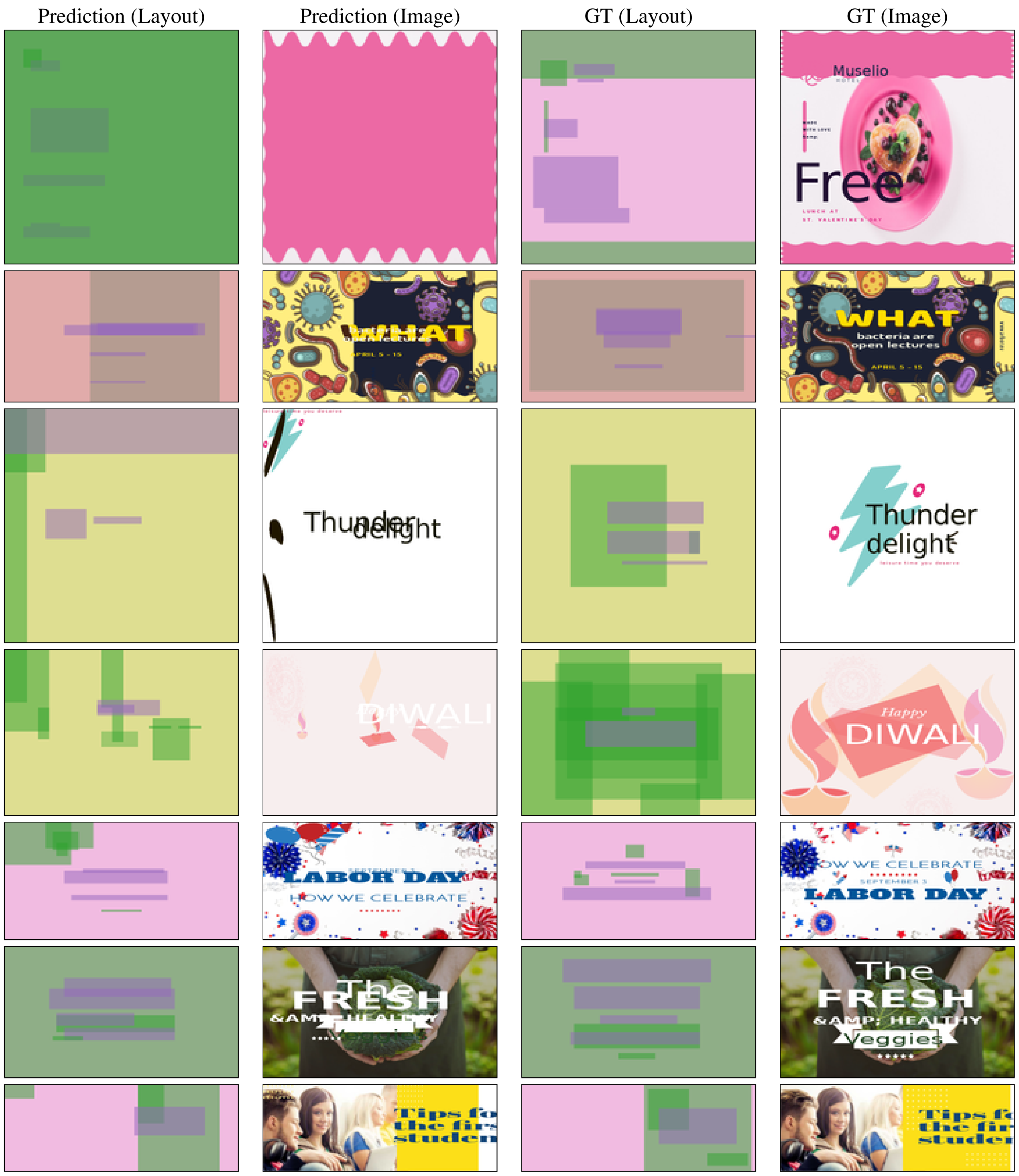


Figure 14. Success cases in ELEM filling. Best viewed with zoom and color.

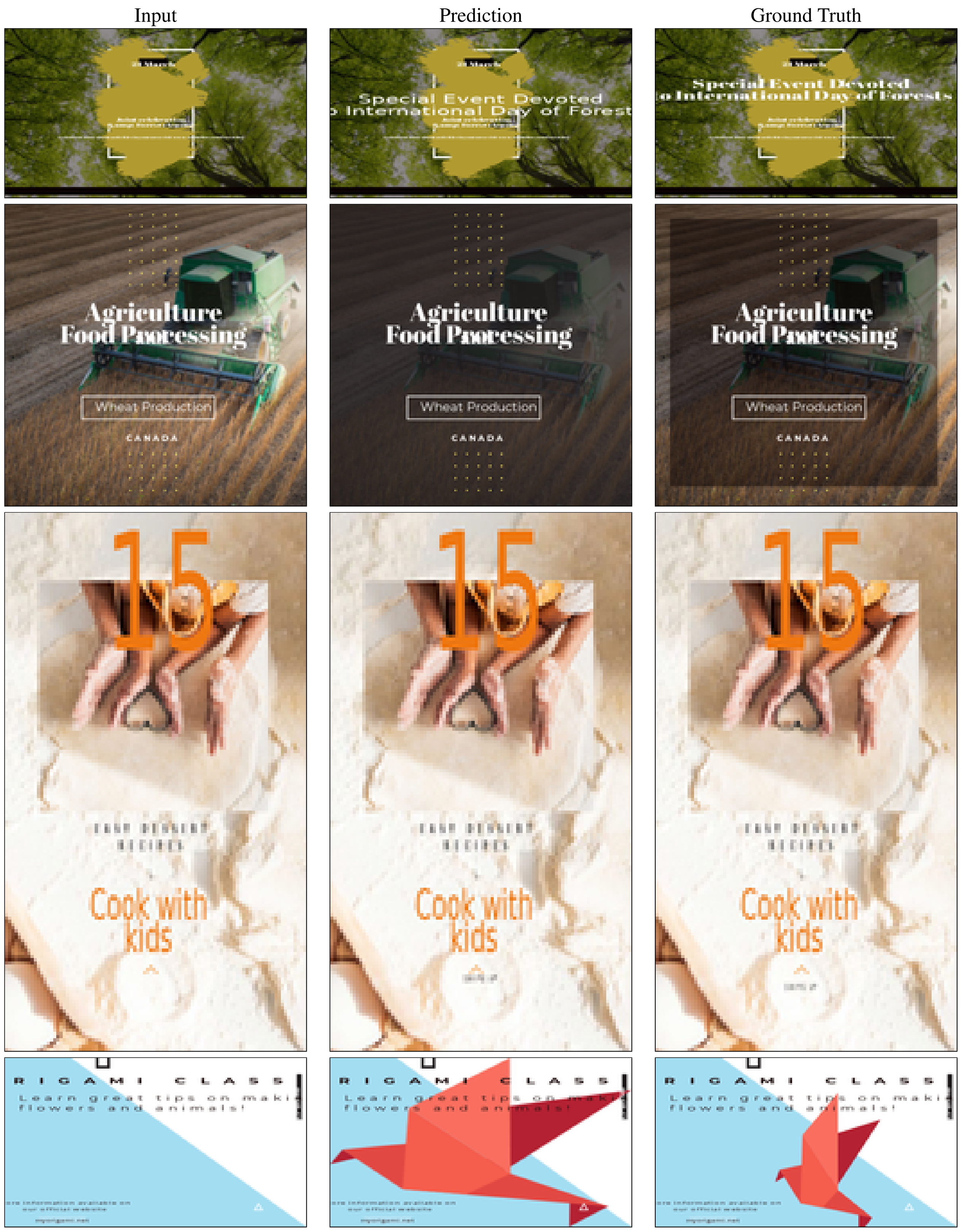
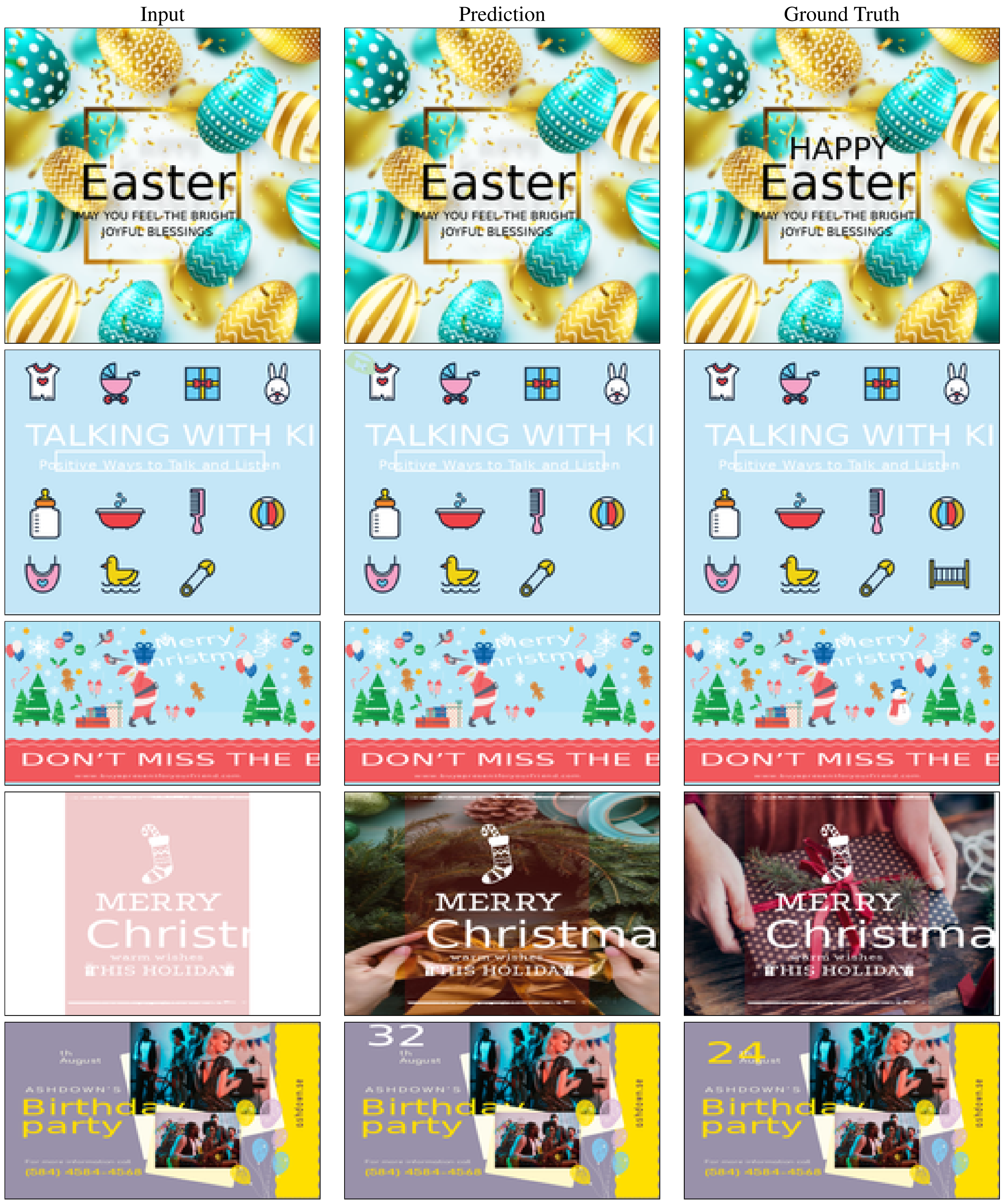


Figure 15. Failure cases in ELEM filling. First row: the model tries to place a vector shape behind texts, but the size is incorrect. Second row: the model tries to generate the main image but fails. Third and fourth rows: the model is unsure and retrieves some common words. Best viewed with zoom and color.



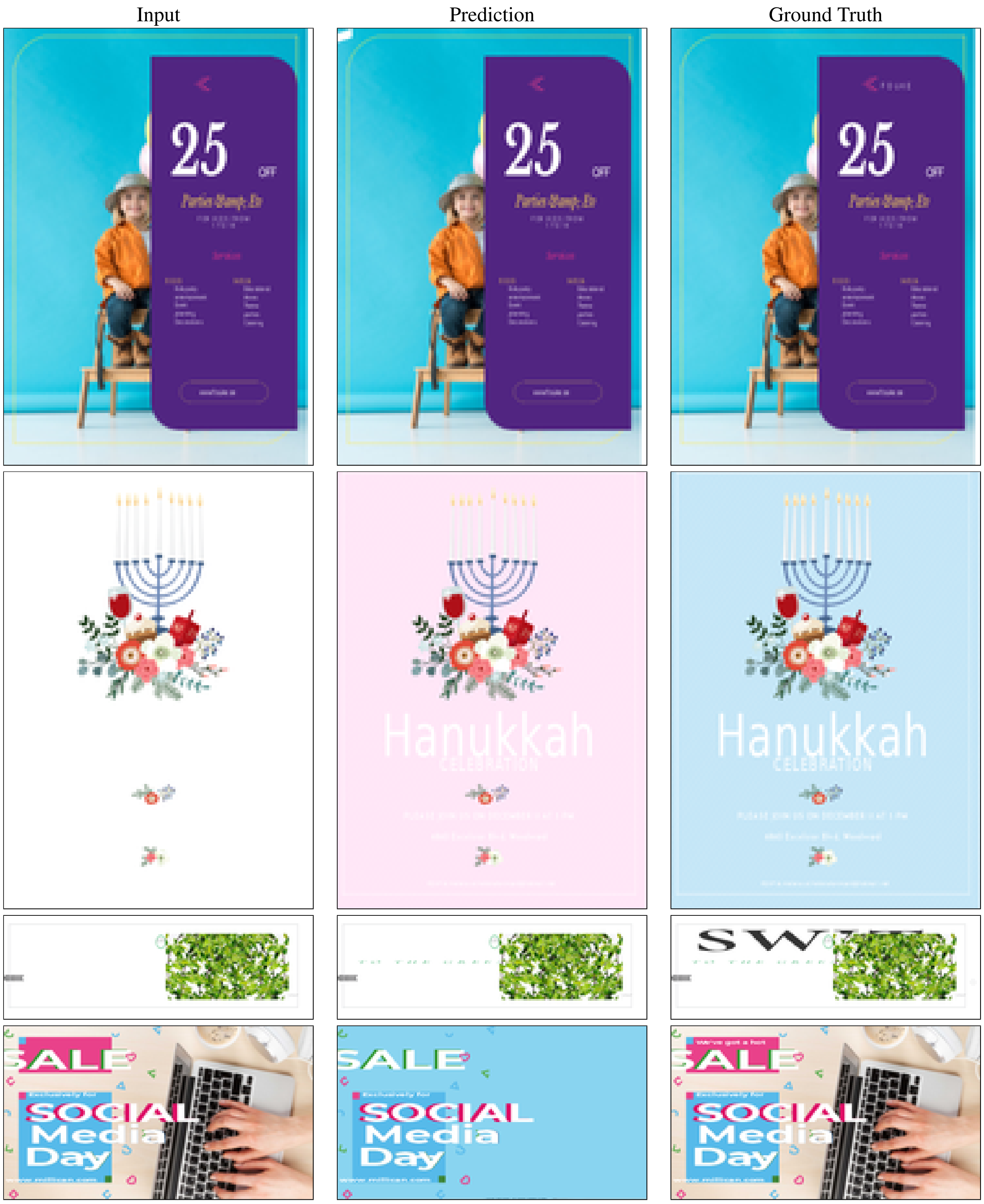


Figure 16. Randomly sampled examples in ELEM filling. Best viewed with zoom and color.