Scene Graph Expansion for Semantics-Guided Image Outpainting (Supplementary Material)

Chiao-An Yang[1], Cheng-Yo Tan[1], Wan-Cyuan Fan[1], Cheng-Fu Yang[1]
Meng-Lin Wu[2], Yu-Chiang Frank Wang[1]
[1] National Taiwan University, [2] Qualcomm Technologies, Inc.
joeyang@ntu.edu.tw, {cy.ugo.tan, christine5200312, joeyy5588}@gmail.com
menglinw@qti.qualcomm.com, ycwang@ntu.edu.tw

A. Implementation

A.1. Architecture

Encoders E_O, E_R and Classifiers C_O, C_R for SGE. Both the object embedding encoder E_O and the relationship embedding encoder E_R are word embedding codebooks. The object classifier C_O has its weight shared with E_O. That is, when classifying an output feature $\boldsymbol{f}_i^O$, the classifier will compare it to each object codevector in the codebook. The object class with the highest cosine similarity between the codevector and the feature is then returned as predicted label. The classifier C_R has its weight shared with E_R for the same reason.

Encoders E_B, E_D and Regressors R_B, R_D for G2L. Both the bounding box encoder E_B and the disparity encoder E_D are two-layer MLPs. The disparity regressor R_D is a two-layer MLP while the bounding box regressor R_B contains two distinct two-layer MLPs, R_B^1 and R_B^2 respectively. As mentioned in Sect. 3.3.2, one of the MLP R_B^1 is trained for predicting the bounding boxes for novel/masked objects and the another one R_B^2 is trained for predicting the boundary offset for existing objects.

Encoder E_I for G2L & L2I. The architecture of the image encoder E_I we deployed is detailed in Table A. The E_I consists of five 2d convolution layers with batch normalization and leaky relu activation in each layer.

Scene Graph Transformers for SGE & G2L. The architecture overview of the SGT layer is shown in Figure A. Noted that the details of our node-level and edge-level attention are depicted in Sect. 3.2.1 and Sect. 3.3.2. All the SGT layer in T_{SGE} and T_{G2L} consists of attention hidden size $d_{atten} = 512$, feed forward size $d_{ff} = 2048$, multi-head number $n_{head} = 4$, and dropout $= 0.1$. Both T_{SGE} and T_{G2L} have 4 SGT layers.

Table A. Architecture design of image encoder E_I for the stages of G2L and L2I.

Type	Argument
Conv2d	c=64, k=5, pad=2
BatchNorm2d	c=64
LeackReLU	slope=0.2
Conv2d	c=128, k=3, stride=2, pad=1
BatchNorm2d	c=128
LeackReLU	slope=0.2
Conv2d	c=128, k=3, stride=2, pad=1
BatchNorm2d	c=128
LeackReLU	slope=0.2
Conv2d	c=256, k=3, stride=2, pad=1
BatchNorm2d	c=256
LeackReLU	slope=0.2
Conv2d	c=256, k=3, pad=1
BatchNorm2d	c=256
LeackReLU	slope=0.2

Generator G_{SPADE} for L2I. Our G_{SPADE} is modified from AttSpade [2], with the ability to take input image as guidance for image outpainting work. The generator is comprised of seven SPADE Resnet blocks. Apart from the size of input channels, which is required to be adjusted due to the input image feature, the architectures and the parameters mostly follow their original setting with the same Pix2Pix models as image and object discriminators. Please refer to the official repository https://github.com/roeiherz/CanonicalSg2Im for more details of the implementation of AttSpade.

Baselines. For the experiments in Table 1 and Table 2, the Transformer, LTNet and GTwE are all set to have the same aforementioned hyper-parameters, which are the attention hidden size $d_{atten} = 512$, feed forward size $d_{ff} = 2048$, multi-head number $n_{head} = 4$, and dropout $= 0.1$. For the experiments in Table 2, the hidden size of GCN is also set

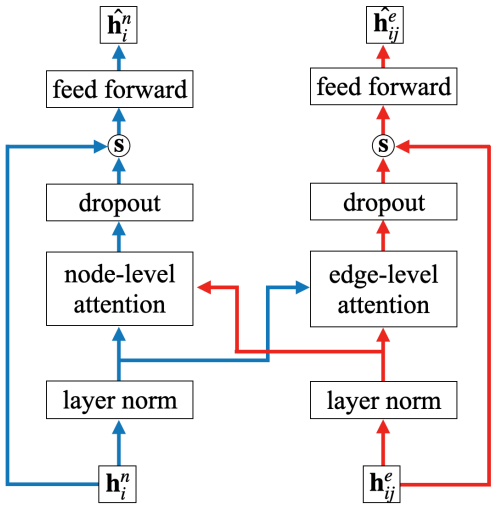


Figure A. **The block diagram for our proposed SGT layer.** The blue arrows indicate the flow of the node inputs $\boldsymbol{h}^n$, while the red ones denote the flow of the edge inputs $\boldsymbol{h}^e$. In order to exploit the observed graph structure, the node-level attention of $\boldsymbol{h}^n$ in our SGT is guided by the edge inputs $\boldsymbol{h}^e$, while the edge-level attention of $\boldsymbol{h}^e$ is guided by the node inputs $\boldsymbol{h}^n$. The symbol $\boldsymbol{s}$ indicates the summation operation.

to 512.

A.2. Datasets

COCO-stuff. The COCO dataset [6] provides images of 80 object categories with bounding box information. COCO-stuff [1] adds 91 more *stuff* categories (e.g., sky, snow, etc.), with 118K/50K images for training and validation.

VG-MSDN. The original Visual Genome dataset [4] contains more than 100K images with noisy labels, and thus we consider its subset of VG-MSDN [5], containing 150 object categories and 50 relationship categories with about 45K/10K for training/validation.

Cityscapes. The Cityscapes dataset [3] contains 2,975/500 images for training and validation. It provides about 40 object categories with bounding box information for each image.

A.3. Scene Graph Generation

For datasets with scene graph annotation, such as VG-MSDN, the labeled scene graphs are used as ground truth without any pruning. For datasets without scene graph annotation, such as COCO and CityScapes, we follow the processing technique of [2, 8] to generate ground truth scene graphs. That is, we utilize the ground truth objects & bounding boxes to construct rule-based scene graphs. This processing stage does *not* require extra data/label supervision.

A.4. Training and Inference

All our models are trained with Adam optimizer and the base learning rate γ is set to 4e-4. We train the three models (T_{SGE}, T_{G2L}, and G_{L2I}) individually rather than end-to-end, since the errors of upper-stream models can greatly deteriorate the result of down-stream ones during training, i.e. incorrect generated novel object in SGE leads to impossible image reconstruction in L2I.

A.4.1 Scene Graph Expansion

We train our SGE model for a total of 10 epochs. We deploy the learning rate schedule as the following. The initial learning rate γ_0 is set to a tenth of γ. In the warmup stage, the learning rate grows linearly up to the maximum learning rate γ. Then the learning rate will hold for a short period of steps. Finally, the learning rate decay exponentially until the minimum learning rate $2.5 \times 10^{-5}\gamma$ is hit at the same time as the end of training. The warmup stage spans a tenth of our total training steps while the hold stage spans a thousandth. It takes 4-12 hours to train on a single NVIDIA-GTX 1080 depending on the datasets and different hyper-parameters.

For data processing on scene graphs, we also introduced a few dummy tokens besides [*MASK*] (the token for *masked* objects and relationships). A dummy object [*IMAGE*] is added to prevent null input. Each object will have an relationship [*IN_IMAGE*] between it and the dummy object [*IMAGE*]. Each object will have an relationship [*SELF*] between it and itself, i.e. $\forall i \in 1:N, r_{ii}^{gt} = \textit{SELF}$. Noted that when considering the objective function for training, such target labels are not excluded from computation.

We follow the setting in [8] where images with less than 3 objects are excluded while images with more than 8 objects will have the exceeding objects ignored. That is, if there are 10 objects in the given scene graph, we will randomly keep 8 of all. This rule is also applied to the training and inference on G2L, while on L2I the maximum number of objects is set to 30.

The objective function $\mathcal{L}_{SGE}$ has to be further adjusted since the distribution of the relationship labels is highly uneven, which can have a negative impact on training. That is, when computing the objective function $\mathcal{L}_{SGE}^{r} = \mathcal{L}_{CE}(r_{ij}^{op}, r_{ij}^{gt})$, most of the target r_{ij}^{gt} is *no-relation* due to the sparsity of the annotation in the datasets. To prevent biased training, weighted cross-entropy loss is deployed instead, and the weight of label *no-relation* is set to 0.05 while the weights of other labels are still set to 1.0.

A.4.2 Scene Graph to Layout

Each ground truth image $\boldsymbol{I}^{gt}$ is first resized to 256×256 pixels and a sub-image of size 128×128 pixels is cropped

Table B. Ablation studies of our SGT for scene graph expansion on VG-MSDN. Note that $\mathcal{L}_{sym}$ exploits the converse relationship from the input node pair as described in Sect. 3.3.1.

	Obj		Rel	
	rAVG	Hit@1 / 5	rAVG	Hit@1 / 5
node-level	9.32	35.7 / 64.8	3.69	46.1 / 81.6
+ edge-level	8.68	38.7 / 68.6	3.80	48.6 / 80.7
+ $\mathcal{L}_{sym}$	8.96	38.2 / 67.8	3.65	52.0 / 82.4
Ours	**8.38**	**39.7 / 68.9**	**3.43**	**55.3 / 84.3**

out as the input image $\boldsymbol{I}^{in}$. The images then are normalized to the range $[-1, 1]$. The associated mask $\boldsymbol{M}^{in}$ indicating which part of the $\boldsymbol{I}^{in}$ is input guidance and which part is missing region will also be concatenated onto $\boldsymbol{I}^{in}$ as the input to E_I. Objects that have their bounding boxes completely out of the cropped region will be considered *masked* and so are the associated relationships.

As for the input bounding boxes $\boldsymbol{b}^{in}$, those of the *masked* objects will be set to $(b^x, b^y, b^w, b^h) = (0.5, 0.5, 0.0, 0.0)$. The bounding box of the dummy object [*IMAGE*] is set to $(0.5, 0.5, 1.0, 1.0)$. The rest of the objects will have their bounding boxes reduced to be in the area of the cropped region. All the input disparities $\boldsymbol{d}^{in}$ will be calculated after the processing on $\boldsymbol{b}^{in}$.

We train our G2L model for a total of 50 epochs. The learning rate schedule is the same as the one used for SGE. It takes about 2 days to train on a single NVIDIA-GTX 1080 depending on the datasets and different hyper-parameters.

A.4.3 Layout to Image

The image pre-processing is the same as G2L. We train our L2I model for a total of 30 epochs. No learning rate schedule is used. It takes about 7 days to train on a single NVIDIA-V100.

B. Ablation Studies

Due to the limitation of spaces, we report only a part of the scores for ablation studies in Sect. 4.2. Here we report the complete table with both rAVG and Hit@1/3 as metrics for ablation studies on SGE in Table. B. It is worth noting that, since scene graphs are often sparse, adding only edge-level attention w/o exploiting converse relationships, i.e. L_{sym}, tends to overfit the observed relationships thus with degraded results.

Additionally, we also conduct ablation studies for G2L to evaluate the effectiveness of our design. As shown in Table C, the introduction of edge-level attention is also beneficial to the layout prediction. Adding in visual features $\boldsymbol{f}^I$ from E_I greatly enhances the performance on layout expansion with existing objects. Overall, with full objective implemented, our model achieve the best performance.

Table C. **Ablation studies of our SGT for scene graph to layout on VG-MSDN in terms of mIoU.** The three mIoU scores in each row are calculated for novel objects, existing objects, and all objects, respectively.

	mIoU
node-level	11.5 / 79.2 / 61.1
+ edge-level	12.0 / 80.0 / 62.1
+ E_I	13.2 / 80.3 / 60.8
Ours	**14.5 / 81.1 / 62.4**

Table D. **Quantative result of image outpainting on VG-MSDN in terms of FID.**

	VG-MSDN FID
Boundless [7]	36.07
AttSpade [2]	23.61
Ours	**23.42**

C. Quantitative Result

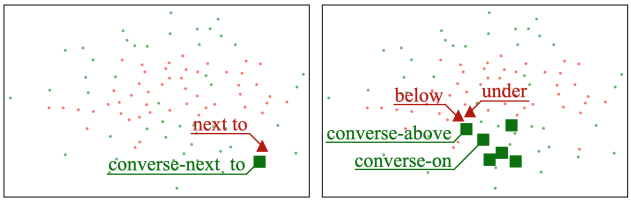
For image outpainting on VG-MSDN and COCO-stuff, we report the FID scores (the lower the better) for evaluation of the quality of outpainted images. As shown in Table D, our model surpass Boundless [7] by a great margin on VG-MSDN. It is also expected that our model is only slightly better than AttSpade due to the similarity between our L2I architecture and theirs. The main difference lies on the fact that our model is able to generate novel objects which itself does not directly enhance image quality but resulting in richer and more meaningful images. Our advantage is reflected better in Sect. D.4

D. Visualization

D.1. t-SNE on relationship features

To examine the effect of our proposed feature converter E_C for exploiting converse relationships on VG-MSDN, we project all the relationship features $E_R(y_i^R)$ and converse relationship features $E_R(\tilde{y}_i^R)$ for all $i = 1 : M$ onto a 2D plane with t-SNE, where each y_i^R is a relationship label while $\tilde{y}_i^R$ is its pseudo converse label, and M is the number of relationship labels.

As shown in Figure B(a), the feature of *next_to* and *converse-next_to* are close to each other since the converse relationship of *next_to* is itself. In Figure B(b), the features of *below* and *under* and the derived converse relationship features of their antonyms such as *converse-on* and



(a) (b)

Figure B. **t-SNE visualization of relationship features f^R.** (a) With the relationship feature of *next_to* highlighted in dark red triangle, its converse version (i.e., *converse-next_to*) is its synonym (shown in dark green square). Thus, their f^R are close to each other. (b) With *below* and *under* as synonyms (i.e., the two dark red triangles), the derived converse relationship features of their antonyms such as *converse-on*, *converse-above*, *converse-be_on*, *converse-stand_on*, *converse-on_top_of* and *converse-lay_on* would be nearby as expected.

converse-above are clustered as expected. Specifically, w/o exploiting the E_C, the cosine similarity scores between "under" and its (1) synonym "below" and (2) antonym "above" are (1) 0.36 and (2) 0.15 respectively. By enforcing the converse property into our SG transformer, the associated scores (1) increase to 0.62 and (2) decrease to -0.93, which is consistent with the improved results reported in ablation studies of Table D. These experiments show that our design does exploit the information carried on converse relationship pairs.

D.2. Three-level image outpainting

In Figure C, we give more examples on how our three-stage semantic image outpainting is achieved, i.e. from the extrapolation on scene graphs, to the extrapolation on layouts, then to the generation of outpainted images.

D.3. Failure cases

We provide a few failure cases in Figure D, which are due to over-annotated data and thus prevent the learning of proper scene graphs.

D.4. Additional Visualization

Additional image outpainting results on VG-MSDN are shown in Figure E, F and G. For example, one can see the first and fourth rows of Fig. F, in which instances of novel/additional animal categories are introduced. In the fifth row of Fig. G, our model is able to synthesize the entire road region with traffic lines which are not presented in the input.

References

[1] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *CVPR*, 2018. 2

[2] Roei Herzig, Amir Bar, Huijuan Xu, Gal Chechik, Trevor Darrell, and Amir Globerson. Learning canonical representations for scene graph to image generation. In *ECCV*, 2020. 1, 2, 3, 6, 7, 8

[3] Bholeshwar Khurana, Soumya Ranjan Dash, Abhishek Bhatia, Aniruddha Mahapatra, Hrituraj Singh, and Kuldeep Kulkarni. Semie: Semantically-aware image extrapolation. In *ICCV*, 2021. 2

[4] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2017. 2

[5] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *ICCV*, 2017. 2

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2

[7] Piotr Teterwak, Aaron Sarna, Dilip Krishnan, Aaron Maschinot, David Belanger, Ce Liu, and William T Freeman. Boundless: Generative adversarial networks for image extension. In *ICCV*, 2019. 3, 6, 7, 8

[8] Cheng-Fu Yang, Wan-Cyuan Fan, Fu-En Yang, and Yu-Chiang Frank Wang. Layouttransformer: Scene layout generation with conceptual and spatial diversity. In *CVPR*, 2021. 2

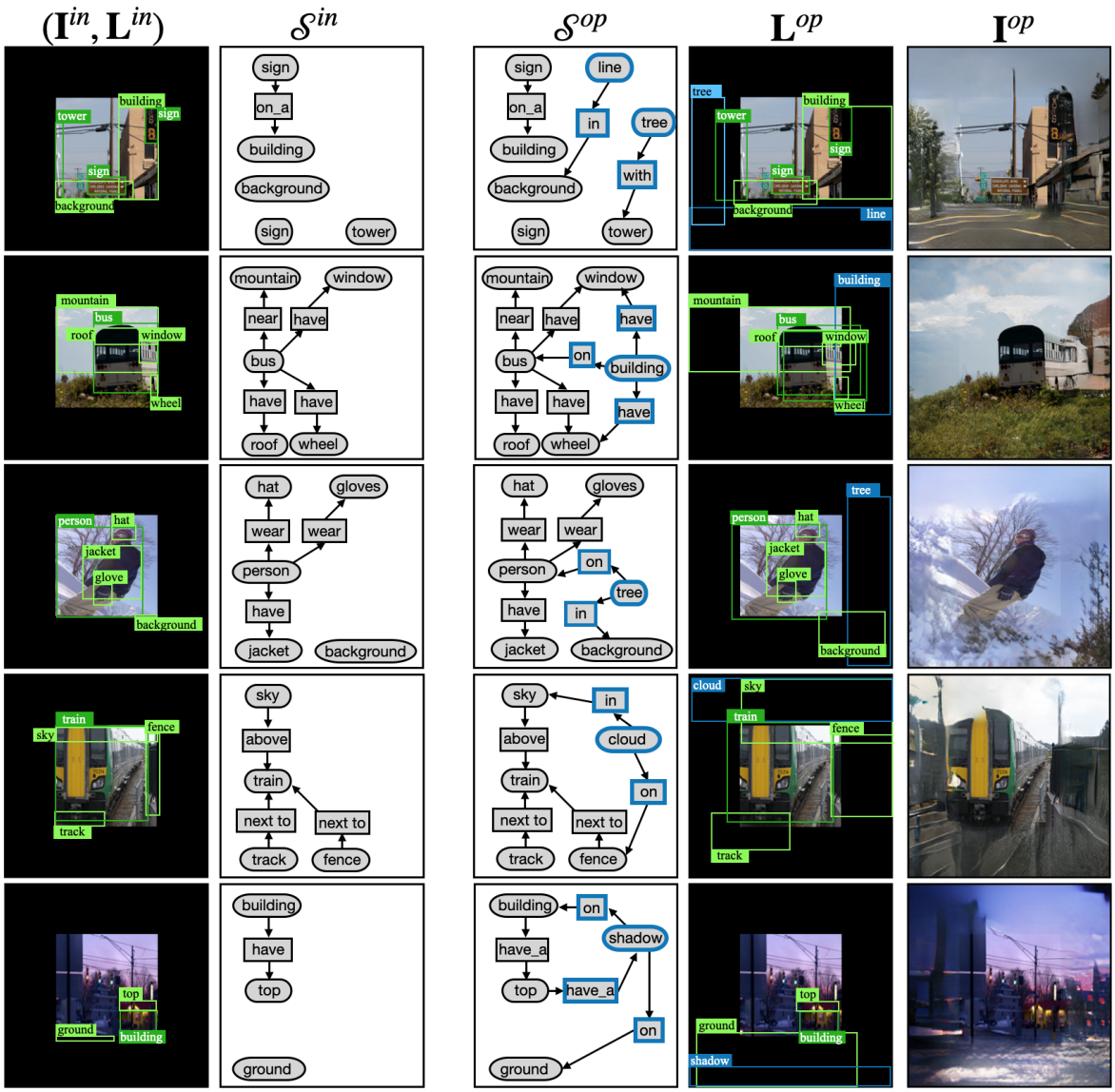


Figure C. **Visualization of our semantics-guided image outpainting on VG-MSDN.** From left to right: input image with layout $(\boldsymbol{I}^{in}, \boldsymbol{L}^{in})$, input scene graph $\mathcal{S}^{in}$, output scene graph $\mathcal{S}^{op}$, output layout $\boldsymbol{L}^{op}$ and output image $\boldsymbol{I}^{op}$. Note that only selected nodes from the input scene graphs are depicted for visualization purposes. Additionally, we note that when the SGE model is used for image outpainting, unlike the experiments in Figure 5(a) where only one novel object is generated, we do not enforce such constraint. That is, our SGE model is capable of generating multiple objects, e.g. the first example.

Figure D. **Two example failure cases.** From left to right: input image $\boldsymbol{I}^{in}$, our output image $\boldsymbol{I}^{op}$ and ground truth image $\boldsymbol{I}^{gt}$. Our model tends to generate images with noisy or repeating content (e.g. repeating "legs" or "windows" which are highlighted as blue bounding boxes) if the training images with similar visual concepts are over annotated. For example, training images of "zibra" are often annotated with multiple (more than 4) "legs", or those of "building" are typically with a large number of "windows" annotated.

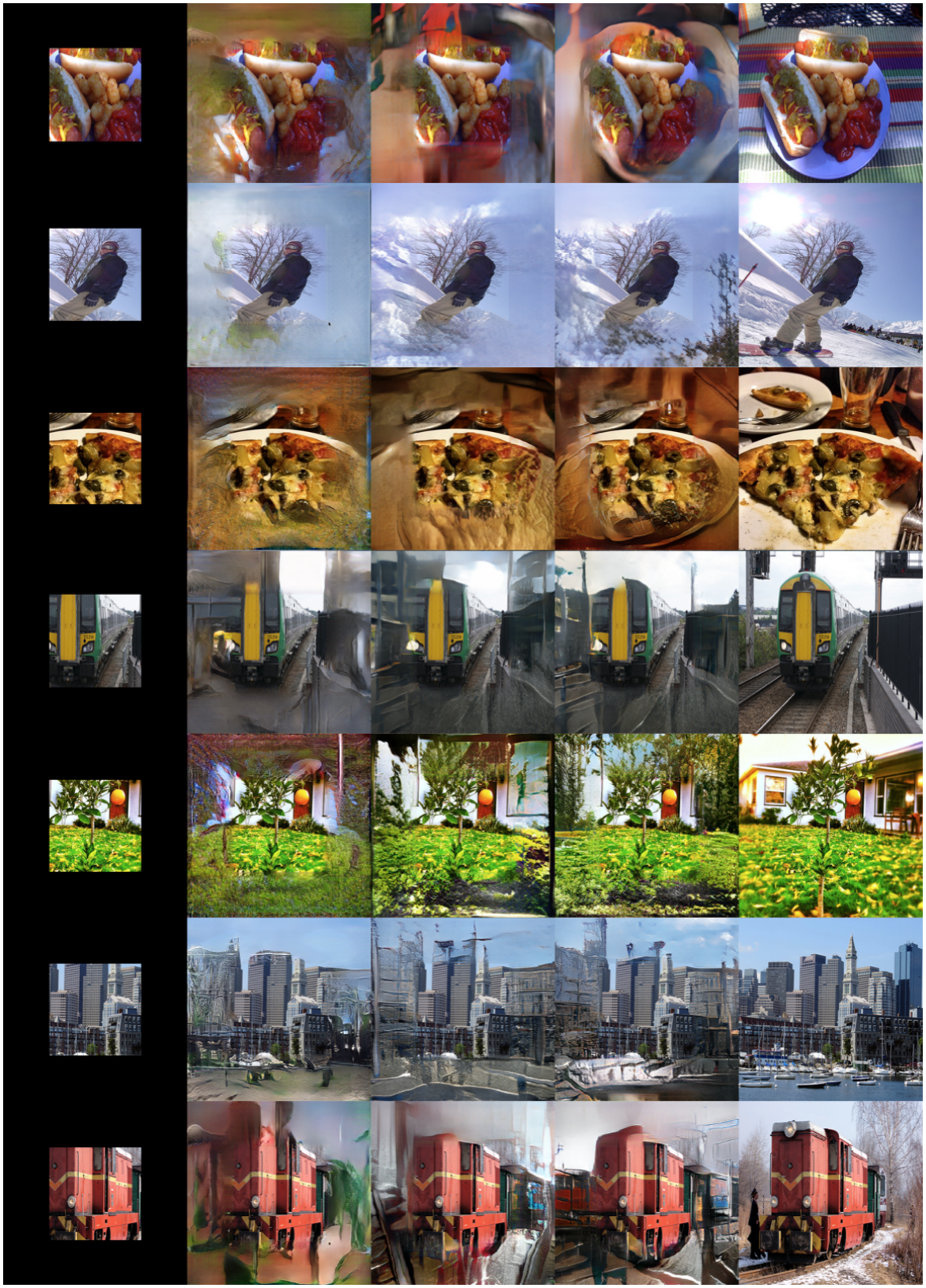


Figure E. **Example outpainting results on VG-MSDN.** From left to right: input image, output images produced by Boundless [7], AttSpade [2] & ours, and the ground truth image.

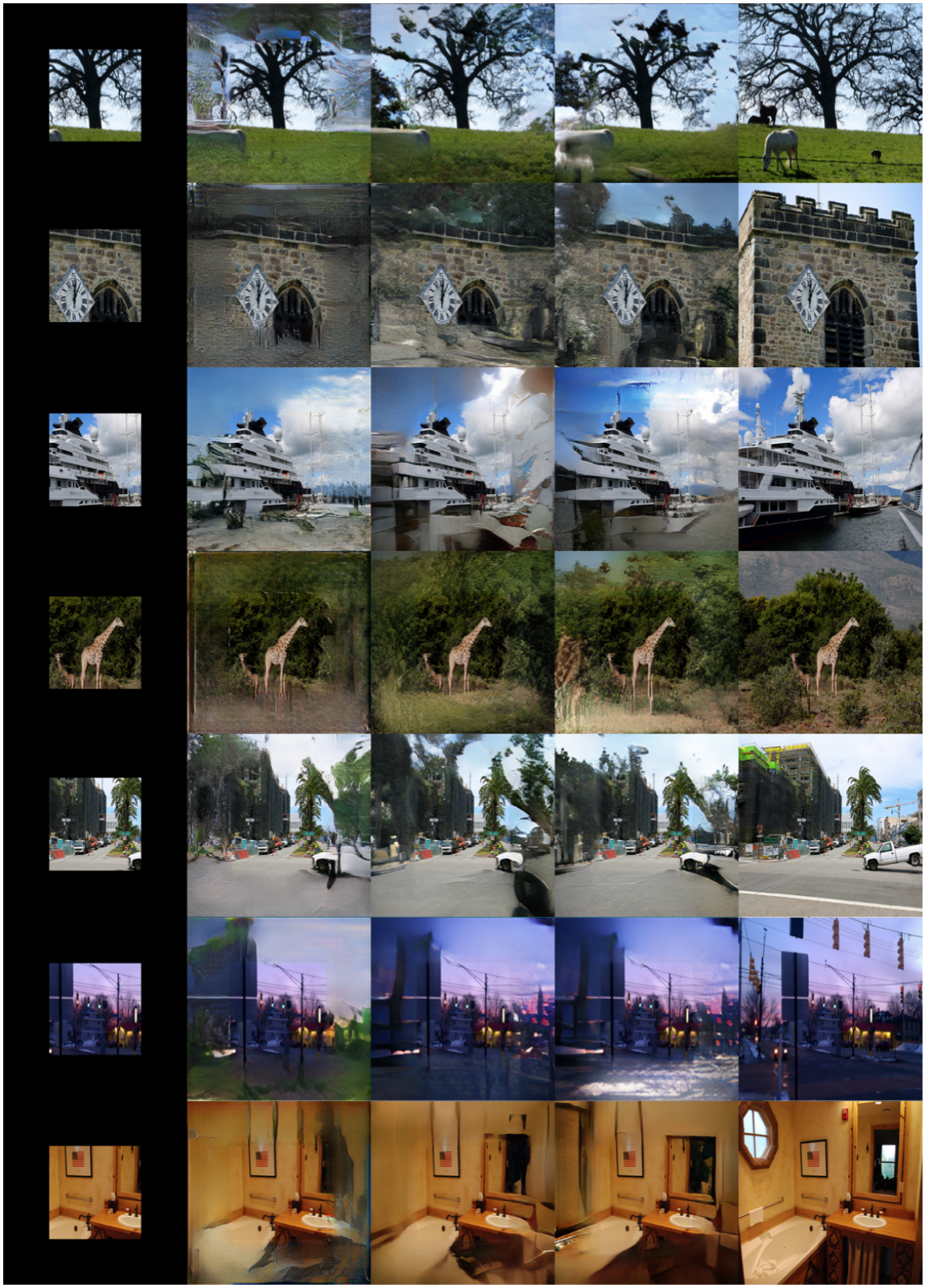


Figure F. **Example outpainting results on VG-MSDN.** From left to right: input image, output images produced by Boundless [7], AttSpade [2] & ours, and the ground truth image.

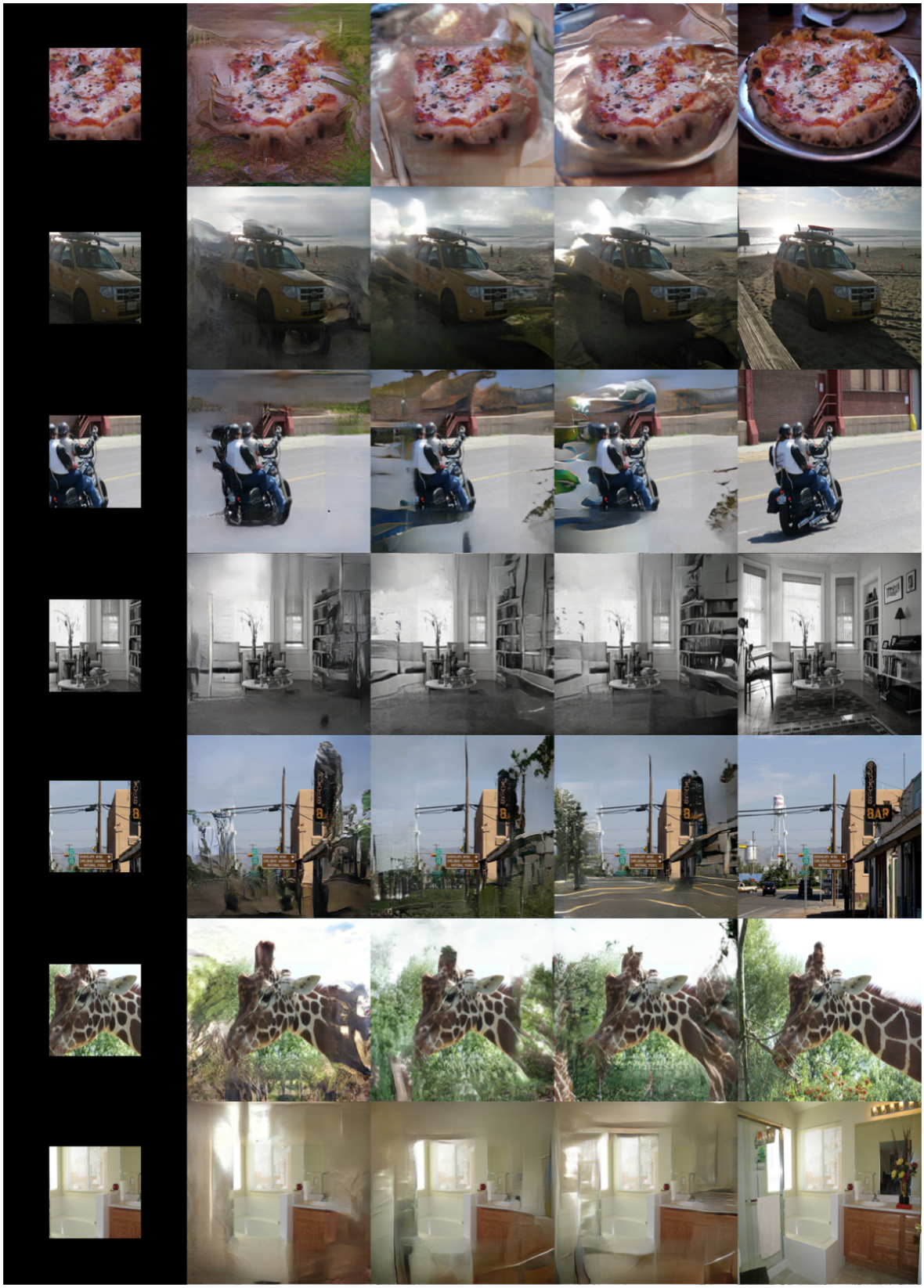


Figure G. **Example outpainting results on VG-MSDN.** From left to right: input image, output images produced by Boundless [7], AttSpade [2] & ours, and the ground truth image.