Contextual Outpainting with Object-Level Contrastive Learning
Supplementary Material

Jiacheng Li[1] Chang Chen[2] Zhiwei Xiong[1]
[1]University of Science and Technology of China
[2]Noah's Ark Lab, Huawei Technologies Co., Ltd.

The supplementary material is organized as follows:

Sec. 1 and the demo video on our project page[1] provide two application scenarios of our method, including time-lapse outpainting and creative editing with an interactive interface.

Sec. 2 provides additional visual results.

Sec. 3 provides more comparison and discussion on alternative solutions.

Sec. 4 provides additional ablation experimental results and analyses.

Sec. 5 provides the details of data preprocessing.

Sec. 6 provides the architectures of components of CTO-GAN and additional experimental details.

1. Applications

1.1. Background Interpolation and Time-lapse Outpainting

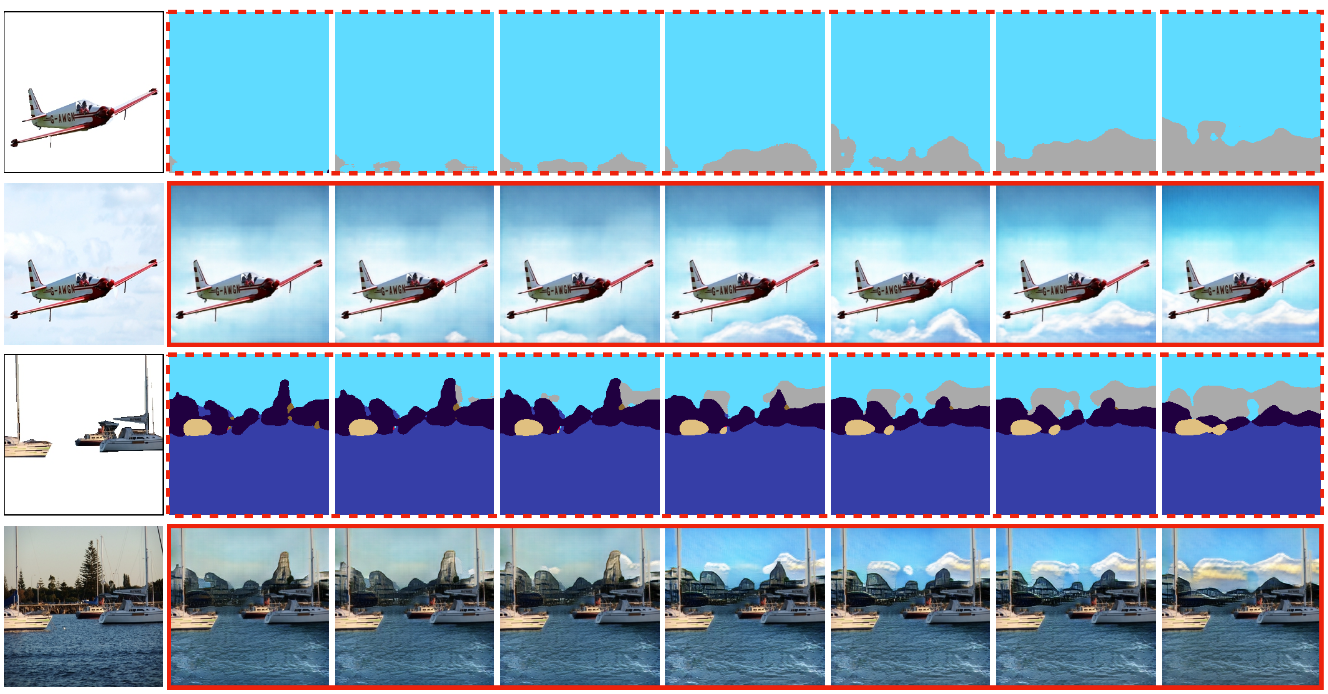


Figure 1. Interpolation results of background semantic layouts and corrresponding outpainted images.

[1]https://ddlee-cn.github.io/cto-gan

Based on VAE, our method is able to sample a series of latent vectors from a continuous latent space, making it possible to interpolate the background semantic layouts as well as contents. In Fig. 1, we show the interpolated background semantic layouts and corresponding outpainted images. As can be seen, the generated semantic layouts and outpainted images transform smoothly, indicating that our method learns a smooth and meaningful latent space for the background semantics. In our demo video, we include more examples and create animated GIFs from the interpolated results. Through interpolating latent vectors, our method is capable of synthesizing a smooth transition of background contents across time. We name this application as "time-lapse outpainting".

1.2. Creative Editing with Interactive Interface

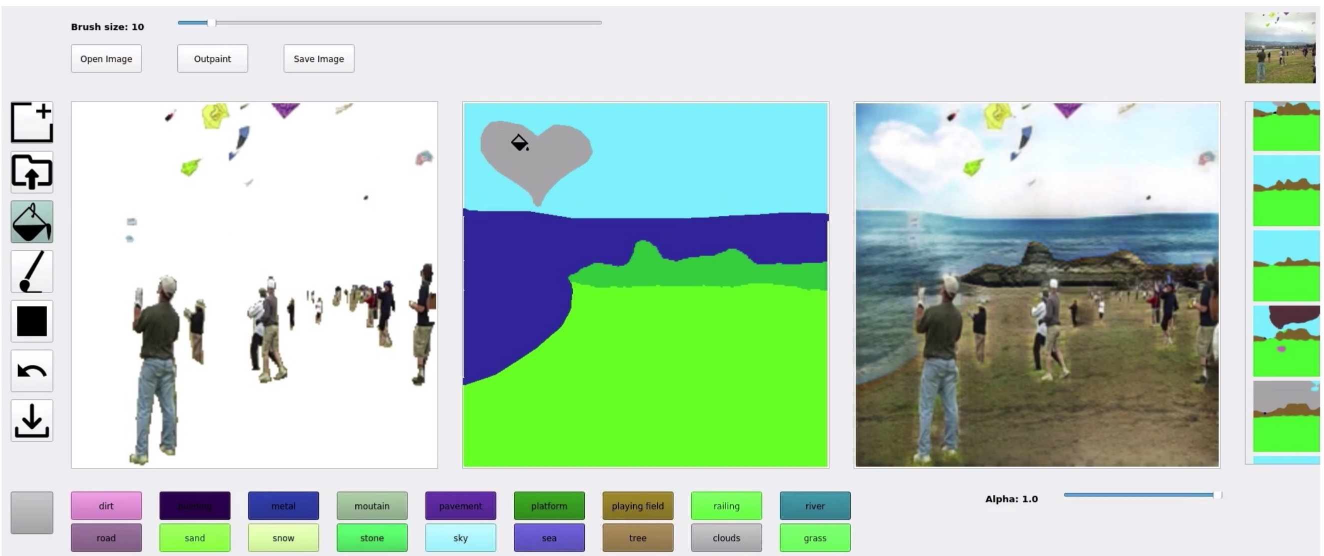


Figure 2. The interactive interface for real-time creative editing of outpainted images.

As mentioned in the paper, one of the benefits of introducing the semantic layout as a bridge is interpretability, since it provides an explicit description for the semantic reasoning result. Thanks to its bridging role, we can control the outpainted image both semantically and spatially through editing the generated semantic layouts. As shown in Fig. 2 and our demo video, we build an interactive application, which enables real-time creative editing for the outpainted images. We show how a user can choose a favorable result from the set of generated semantic layouts, and outpaint the input image with the chosen semantic layout as guidance. We also demonstrate how a user can add, move, or change the background contents of the outpainted image by editing its semantic layout on the canvas. Our implementation of the GUI builds upon MaskGAN[2] [8] and SEAN[3] [22].

[2]https://github.com/switchablenorms/CelebAMask-HQ
[3]https://github.com/ZPdesu/SEAN

2. Additional Visual Results

In Fig. 3 and Fig. 4, we show additional visual results and qualitative comparison with existing methods. As can be seen, our method generates coherent and diverse background contents, outperforming comparison methods.

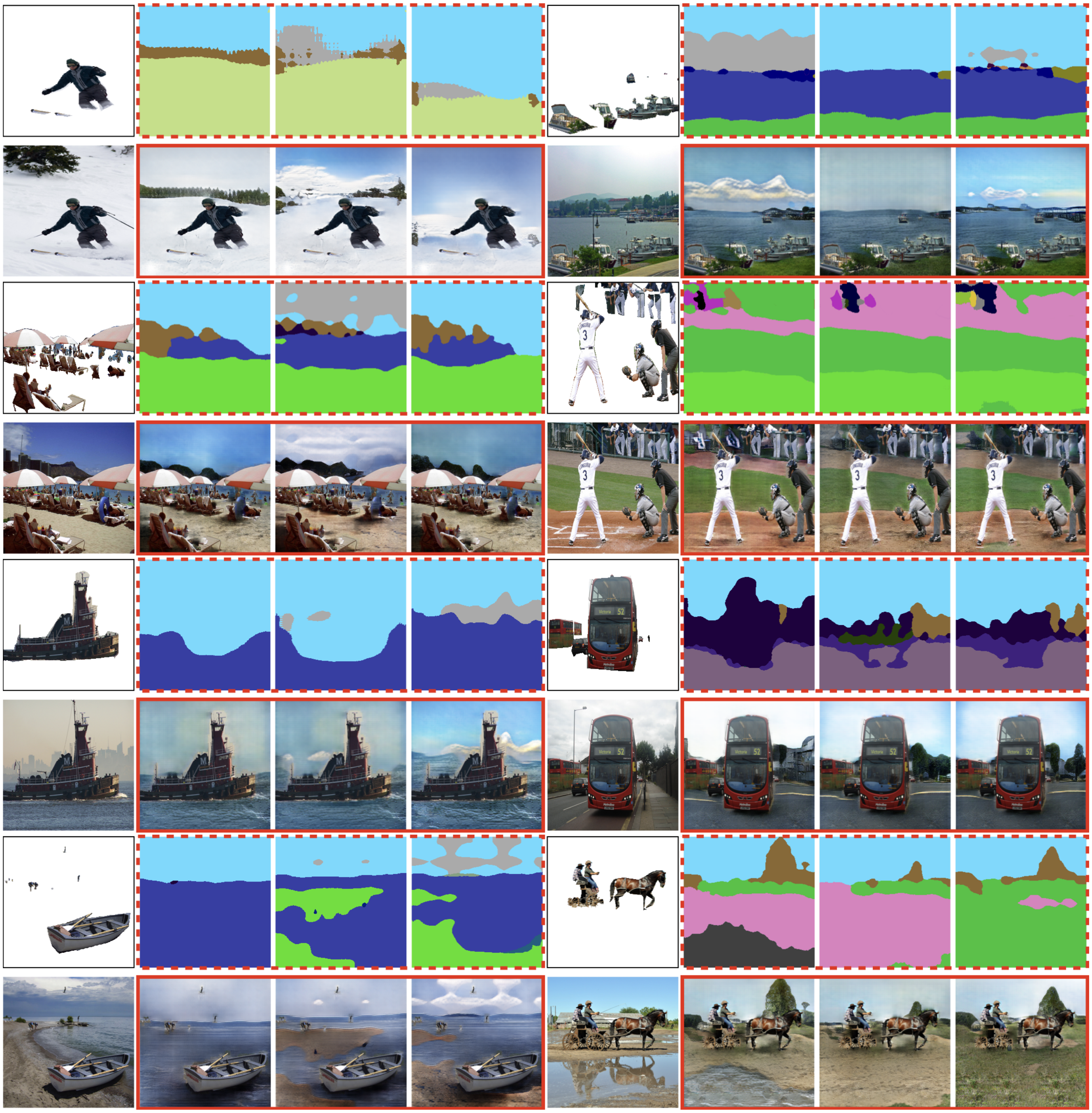


Figure 3. Additional visual results generated by our method. For each example, we show the semantic layouts (in red dashed boxes) and the outpainted images (in red boxes) produced by our method after the input image and the ground truth image, respectively.

Figure 4. Additional qualitative comparison with existing methods. For each example, from top to bottom, from left to right, the pictures are: the input image, results of GatedConv [19], Boundless [7], results of MIO [20] (in blue box), results of PIC [21] (in purple box), results of DSI [13] (in yellow box) and results of our method (in red box).

3. More Comparison with Alternative Solutions

	FID ↓	LPIPS ↓	mIoU ↑	Accu ↑
GT Foreground→pix2pix→SPADE	41.81	0.471	23.3	35.2
DeepLabV2→pix2pix→SPADE	43.69	0.520	18.9	26.8
pix2pixHD	45.97	0.473	22.7	33.6
Ours	**27.34**	**0.371**	**31.5**	**47.0**

Table 1. Comparisons with a 3-stage solution and image-to-image translation method.

Comparison with context modeling methods. Context modeling methods like [14] may play a role in a cascaded 3-stage solution, *i.e.*, "foreground recognition (image-to-layout) $\rightarrow$ context modeling (layout-to-layout) $\rightarrow$ background synthesis (layout-to-image)". However, as explored in [17, 23], contextual bias plays a key role in image recognition methods. In the semantic segmentation task, the mIoU for the foreground objects of DeepLabV2 [2] on the COCO dataset drops from 46.1 to 39.8. Consequently, the 3-stage solution limits itself because of the performance drop in recognizing foregrounds without involving their context. We validate this argument in Table 1, where pix2pix [5] (a strong baseline in [14]) is used for context modeling and SPADE [12] is used for image synthesis. Instead of predicting the pixel-level classes for the foreground objects, we relate the latent representations of foreground and background contents in a joint embedding space through contrastive learning.

Comparison with image-to-image translation methods. Further, we supplement the comparison results with pix2pixHD [16], where our method still has a clear advantage. Image-to-image translation methods often assume a pixel-to-pixel alignment between the source and target images, which is violated in the task of contextual outpainting.

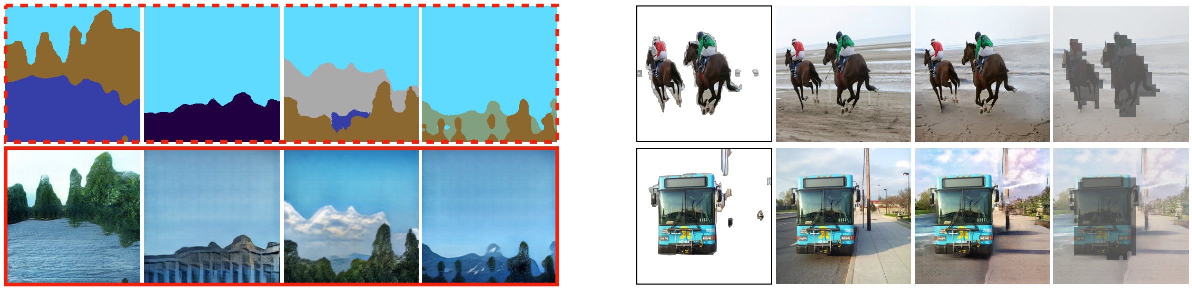
4. Additional Ablation Experiments and Analyses

	FID ↓	LPIPS ↓	mIoU ↑	Accu ↑
Ours	**27.34**	0.371	**31.5**	**47.0**
increase **K**	28.46	**0.366**	30.7	45.4
decrease **K**	29.02	0.372	30.0	44.1
MLP proj. head	29.90	0.375	30.4	46.2
increase latent size	29.11	0.389	29.6	44.7
image-level contra.	36.77	0.397	25.5	38.5
Ours w/ semantic dis.	29.10	0.388	30.0	45.2

Table 2. Additional ablation experiments on the design choices of contrastive regularization. MLP proj. head indicates adding an MLP head after pooling for learned representations. Image-level contra. denotes the strategy of applying contrastive regularization at the image level instead of the object level. Semantic dis. denotes the semantic segmentation discriminator proposed in [15].

Additional ablation experiments on the design choices of contrastive regularization. As listed in Table 2, we investigate the influences of different design choices of contrastive regularization. We find that tuning hyperparameters (increasing or decreasing the memory bank size **K**), adding additional MLP layers, and increasing latent vector size achieve comparable performance with the original design, demonstrating the robustness of the regularization effect. Furthermore, we conduct an experiment with an image-level contrastive regularization strategy, in which we concatenate the foreground and background representations together as a description for the entire image and enforce contrastive relationships across these image-level representations. The image-level strategy suffers from the noise caused by different appearances of positive samples, resulting in poor performance. This result also validates the benefit of utilizing the semantic layout as bridging information, which narrows the appearance gap at the semantic level.

Additional ablation experiment on the context-aware discriminator. Our context-aware discriminator shares similar merits with a recent work on augmenting the ability of the discriminator for image-to-image translation [15]. However, the discriminator in [15] serves as an image segmentation network, aiming at judging the alignment between the generated images and the provided condition signal. As listed in Table 2, replacing the context-aware discriminator in our method with the one in [15] hurts performance since it provides noisy feedback when the foreground objects remain untouched.



(a)

(b)

Figure 5. Additional analyses. (a) Visual results of unconditional background generation. (b) Visualization of the score map from the context-aware discriminator.

From contextual outpainting to unconditional background generation. Our method degenerates to unconditional background image generation when there is no foreground content provided. Under this scenario, our method is able to generate meaningful background contents, as shown in Fig. 5(a).

Visualization of the score map from the context-aware discriminator. We illustrate the score map predicted by the context-aware discriminator in Fig. 5(b). As expected, the context-aware discriminator learns to detect the generated background area, making it harder to be fooled by the generator and thus resulting in better visual quality.

5. Data Preprocessing

Figure 6. From left to right: the ground truth image I, the ground truth semantic layout S_{gt} (with both foreground and background annotations), the input foreground image I_{fg}, the background image I_{bg}, the inpainted background image I'_{bg}, the pseudo background-only semantic layout S_{bg}, the background-only semantic layout S^+_{bg} from the other image inside the same image group. We also show the ground truth image of S^+_{bg} for reference.

The COCO-Stuff dataset[4,5] provides pixel-level annotations (S_{gt} in Fig. 6) for both foreground and background classes. We preprocess the dataset in the following steps. As shown in Fig. 6, we perform an inpainting operation F_{inpt} to fill the foreground region with background pixels, obtaining I'_{bg}. We adopt the PatchMatch inpainting algorithm [1]. Then, we inference a pre-trained DeepLabV2 [2] model as F_{seg} to get the background semantic layout with only background classes (S_{bg}). Compared to the ground truth semantic layout S_{gt}, S_{bg} only contains the background semantics, which we set as the traning target in the semantic reasoning stage. These background semantic layouts are also used as the conditional signal for training the image generator in the content generation stage. To reorganize the images in the COCO-Stuff dataset, we simply group them according to their foreground classes, resulting in 11,296 groups. As shown in the last two items of Fig. 6, we assume the images inside the same group share similar background semantics (S_{bg} and S^+_{bg}). In the COCO-stuff dataset, the foreground (thing) and background (stuff) definitions are not always consistent across images. For indoor scenes, we find it hard to select saliency objects as the remaining foreground. Thus, we focus the outdoor scenes.

[4] https://cocodataset.org/
[5] https://github.com/nightrome/cocostuff

6. Network Architectures of CTO-GAN and Additional Experimental Details

6.1. The Semantic Reasoning Stage

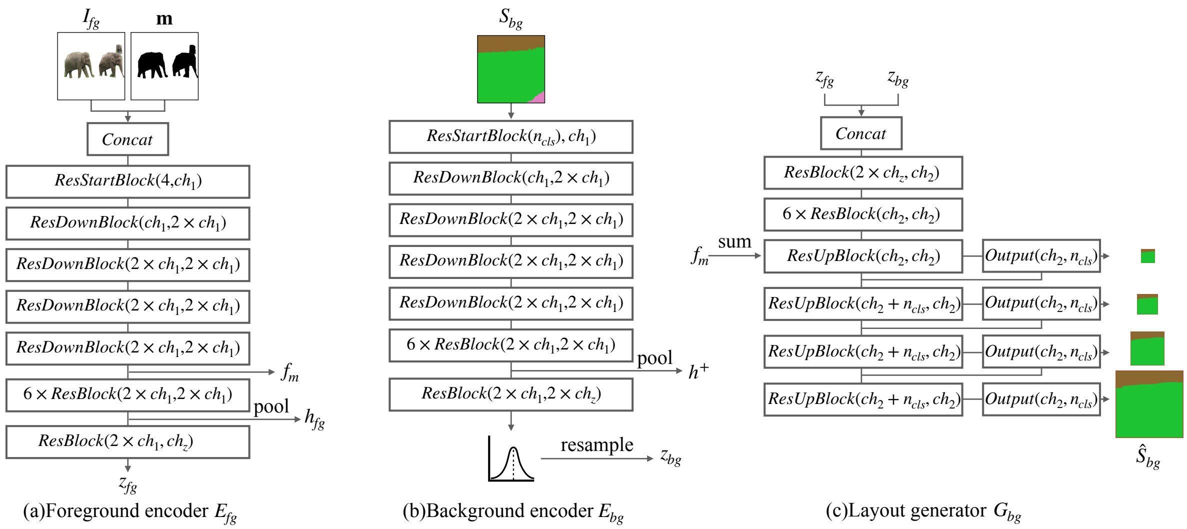


Figure 7. Detailed architectures of components in the semantic reasoning stage. We illustrate the basic blocks in the format of $\texttt{BlockName}(\texttt{input\_channel}, \texttt{output\_channel})$. $ResStartBlock$, $ResBlock$, $ResDownBlock$, $ResUpBlock$, and $Output$ are the same as in PIC [21]. ch_* denote the base channel sizes of convolution layers, and n_{cls} is the number of semantic classes.

The design of the foreground encoder E_{fg}, the background encoder E_{bg}, and the layout generator G_{bg} in the semantic reasoning stage follows PIC [21]. As shown in Fig. 7, the foreground representation h_{fg} and the background representation h^+ (or h^-) are obtained by global pooling before the latent vectors z_{fg} and z_{bg}. Following PIC, we allow a skip connection between E_{fg} and G_{bg} via an intermediate feature f_m, which is summed with the feature tensor in G_{bg}. We predict the possible semantic layout $\hat{S}_{bg}$ at 4 scales. The semantic layout prediction of the coarser scale is concatenated for learning residuals in the finer scale. We set the base channel size of E_{fg} and E_{bg} as $ch_1 = 64$ and that of G_{bg} as $ch_2 = 128$. The latent vector size ch_z is set to 128. The discriminator for the generated semantic layout in this stage is similar to the one in Fig. 8(b), but with the semantic layout as the only input. The loss function for training the semantic reasoning stage of CTO-GAN is

$$L_{SR} = L_{CMC} + \lambda_1 L_{KL} + \lambda_2 (L_{CE} + L_{focal}) + \lambda_3 L_{GAN-layout}, \quad (1)$$

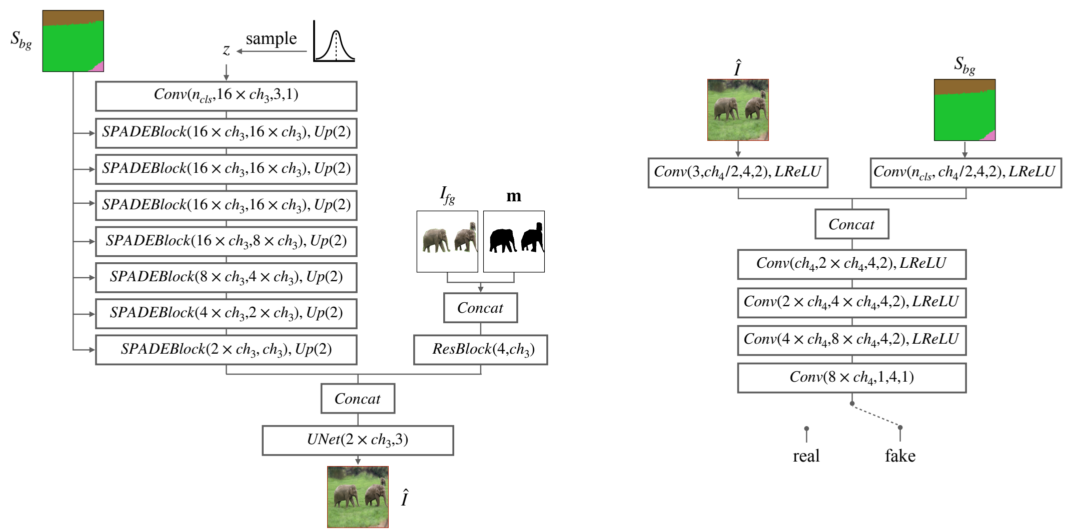
where L_{CMC} is the proposed cross-modal contrastive loss, L_{KL} the KL divergence regularization term, L_{CE} the cross-entropy loss, L_{focal} the focal loss [9], $L_{GAN-layout}$ the least square GAN loss [10] for semantic layout, and λ_* are balancing parameters. We set $\lambda_1 = 200$, $\lambda_2 = 5$, and $\lambda_3 = 1$ across all experiments.

6.2. The Content Generation Stage

As illustrated in Fig. 8(a), the content generation stage of CTO-GAN is inspired by SPADE [12]. We add a UNet generator [5] to aggregate the features of the foreground image and upsampled background features to obtain the outpainted image $\hat{I}$. The base channel size ch_3 of G_{img} is set to 64. The image discriminator D_{img} follows the multi-scale patch discriminator in pix2pixHD [16], but with the projected features of S_{bg} as conditional input, as shown in Fig. 8(b). We incorporate the discriminator at 2 scales with a base channel size ch_4 of 64. The architecture of the context-aware discriminator follows DeepLabV2 [2] with a base channel size of 16. The loss function for training the content generation stage of CTO-GAN is

$$L_{CG} = L_{Recon} + \lambda_4 L_{FM} + \lambda_5 L_{VGG} + \lambda_6 L_{GAN-det} + \lambda_7 L_{GAN-img}, \quad (2)$$

where L_{Recon} is the ℓ_1 distance, L_{FM} the distance of features from D_{img}, L_{VGG} the feature distance of the VGG network, $L_{GAN-det}$ the BCE loss of the proposed context-aware discriminator, $L_{GAN-img}$ the least square GAN loss for image, and λ_* are balancing parameters. We set $\lambda_4 = 0.2$, $\lambda_5 = 0.4$, $\lambda_6 = 0.01$, and $\lambda_7 = 0.1$ across all experiments.



(a)Image generator G_{img} (b)Image discriminator D_{img}

Figure 8. Detailed architectures of components in the content generation stage. We illustrate the convolution layer in the format of `Conv(input_channel, output_channel, kernel_size, stride)`. $SPADEBlock$ follows SPADE [12], and $UNet$ follows the "UNet" generator in Pix2pix [5]. Up denotes the nearest neighbor upsample operation. $LReLU$ denotes the Leaky ReLU activation function [18].

6.3. Experimental Details

We apply spectral normalization [11] to all the convolutional layers and sync batch normalization in the basic blocks. Following PIC [21], we regularize the learned distribution of background semantic layouts to the normal distribution with an adaptive variance according to the mask size. The τ in the CMC loss is set to 0.07 following MoCo [4]. All learnable parameters are initialized with the xavier initialization [3] and optimized by the Adam optimizer [6] with $\beta_1 = 0$ and $\beta_2 = 0.999$ at a fixed learning rate of 1×10^{-4}. The batch size is 64 for the semantic reasoning stage and 8 for the content generation stage. We train the two stages over 200K iterations in parallel.

For comparison methods, we use the official implementation of GatedConv[6], MIO[7], PIC[8], and DSI[9]. We use a third-party implementation[10] of Boundless. For MIO, we increase the input size from 128×128 to 256×256 and increase the network capacity accordingly. We retrain these methods on the COCO-Stuff dataset with default hyperparameters. We use a third-party implementation[11] of DeepLabV2 for data preprocessing and semantic coherence evaluation.

[6] https://github.com/JiahuiYu/generative_inpainting
[7] https://github.com/owenzlz/DiverseOutpaint
[8] https://github.com/lyndonzheng/Pluralistic-Inpainting
[9] https://github.com/USTC-JialunPeng/Diverse-Structure-Inpainting
[10] https://github.com/recong/Boundless-in-Pytorch
[11] https://github.com/kazuto1011/deeplab-pytorch

References

[1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B. Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009. 6

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2018. 5, 6, 7

[3] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010. 8

[4] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 8

[5] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 5, 7, 8

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 8

[7] Dilip Krishnan, Piotr Teterwak, Aaron Sarna, Aaron Maschinot, Ce Liu, David Belanger, and William T. Freeman. Boundless: Generative adversarial networks for image extension. In *ICCV*, 2019. 4

[8] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *CVPR*, 2020. 2

[9] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 7

[10] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, 2017. 7

[11] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018. 8

[12] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019. 5, 7, 8

[13] Jialun Peng, Dong Liu, Songcen Xu, and Houqiang Li. Generating diverse structure for image inpainting with hierarchical VQ-VAE. In *CVPR*, 2021. 4

[14] Xiaotian Qiao, Quanlong Zheng, Ying Cao, and Rynson W. H. Lau. Tell me where I am: Object-level scene context prediction. In *CVPR*, 2019. 5

[15] Edgar Schönfeld, Vadim Sushko, Dan Zhang, Juergen Gall, Bernt Schiele, and Anna Khoreva. You only need adversarial supervision for semantic image synthesis. In *ICLR*, 2021. 5

[16] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018. 5, 7

[17] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. In *ICLR*, 2021. 5

[18] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv:1505.00853*, 2015. 8

[19] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Free-form image inpainting with gated convolution. In *ICCV*, 2019. 4

[20] Lingzhi Zhang, Jiancong Wang, and Jianbo Shi. Multimodal image outpainting with regularized normalized diversification. In *WACV*, 2020. 4

[21] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic image completion. In *CVPR*, 2019. 4, 7, 8

[22] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. SEAN: image synthesis with semantic region-adaptive normalization. In *CVPR*, 2020. 2

[23] Zhuotun Zhu, Lingxi Xie, and Alan L. Yuille. Object recognition with and without objects. In Carles Sierra, editor, *IJCAI*, 2017. 5