SESAME: Semantic Editing of Scenes by Adding, Manipulating or Erasing Objects - Supplementary Material

Evangelos Ntavelis[1,2], Andrés Romero[1], Iason Kastanis[2], Luc Van Gool[1,3], and Radu Timofte[1]

[1] Computer Vision Lab, ETH Zurich, Switzerland
[2] Robotics and Machine Learning, CSEM SA, Switzerland
[3] PSI, ESAT, KU Leuven, Belgium

Model Architectures The detailed architectures of the SESAME generator and discriminator are depicted in Tables 1 and 2 respectively. Please note that for the discriminator we use this architecture twice, once for each scale.

To further justify the architectural choices of our generator we compare against Pix2PixHD++ [6], with SPADE layers on the decoder part. We use both our SESAME discriminator and the commonly used PatchGAN.

The performance assessment is reported in Table 3. The use of our discriminator over PatchGAN improved the results in almost all cases. However, we observe that while our generator-discriminator combination performs the best, the second combination is without any of our networks. We argue that our proposed method works better together as the large receptive field provided by the dilated convolutions in our generator synergizes well with the highly focused gradient flow coming from our discriminator.

Replacement of objects Apart from adding and removing objects, SESAME can also be used to replace an instance of an object, given that we know its class and its outline. SESAME can be utilized in this manner for dataset augmentation. We conduct experiments on replacing objects in street scenes of Cityscapes and we ablate on the usage of our SESAME discriminator against the PatchGAN. In Table 4 we measure the FID score of the image results, the SSIM of the generated regions and we also devoted a part of our user study, described in Section 4 of the main paper, to test which of the two discriminators produces the most *photo-realistic* results. Visual results can be found in Figure 1.

Visual Results We show more edited and generated images produced by our method:

- Figure 1 contains visual results of our ablation analysis on various access levels of semantics and different discriminators.
- Figure 2 contains visual results for removing objects under different configurations.

Table 1. SESAME generator architecture. We depict the number of **F**ilters, the **K**ernel size, the **S**tride and the **D**ilation factor

Layer	Normalization	Activation
ConvBlock F = 64, K = 7, S = 1, D = 1	Instance	ReLU
ConvBlock F = 128, K = 3, S = 2, D = 1	Instance	ReLU
ConvBlock F = 256, K = 3 , S = 2, D = 1	Instance	ReLU
ResBlock F = 256, K = 3, S = 1, D = 1	Instance	ReLU
ResBlock F = 256, K = 3, S = 1, D = 2	Instance	ReLU
ResBlock F = 256, K = 3, S = 1, D = 2	Instance	ReLU
ResBlock F = 256, K = 3, S = 1, D = 2	Instance	ReLU
ResBlock F = 256, K = 3, S = 1, D = 2	SPADE	LeakyReLU(0.02)
ResBlock F = 256, K = 3, S = 1, D = 2	SPADE	LeakyReLU(0.02)
ResBlock F = 256, K = 3, S = 1, D = 2	SPADE	LeakyReLU(0.02)
ResBlock F = 256, K = 3, S = 1, D = 2	SPADE	LeakyReLU(0.02)
ResBlock F = 256, K = 3, S = 1, D = 1	SPADE	LeakyReLU(0.02)
Nearest Neighbour Upsampling ×2	-	-
ResBlock F = 128, K = 3, S = 1, D = 1	SPADE	LeakyReLU(0.02)
Nearest Neighbour Upsampling ×2	-	-
ResBlock F = 64, K = 3, S = 1, D = 1	SPADE	LeakyReLU(0.02)
ConvBlock F = 3, K = 3, S = 1, D = 1	-	TanH

Table 2. SESAME discriminator architecture **per scale**, We depict the number of **F**ilters, the **K**ernel size, the **S**tride and the **D**ilation factor

Layer	Normalization	Activation
Image Stream		
ConvBlock F = 64, K = 4, S = 2, D = 1	-	LeakyReLU(0.02)
ConvBlock F = 128, K = 4, S = 2, D = 1	SpectralInstance	LeakyReLU(0.02)
ConvBlock F = 256, K = 4, S = 2, D = 1	SpectralInstance	LeakyReLU(0.02)
ConvBlock F = 512, K = 4, S = 1, D = 1	SpectralInstance	LeakyReLU(0.02)
Semantics Stream		
ConvBlock F = 64, K = 4, S = 2, D = 1	-	LeakyReLU(0.02)
ConvBlock F = 128, K = 4, S = 2, D = 1	SpectralInstance	LeakyReLU(0.02)
ConvBlock F = 256, K = 4, S = 2, D = 1	SpectralInstance	LeakyReLU(0.02)
ConvBlock F = 512, K = 4, S = 1, D = 1	SpectralInstance	LeakyReLU(0.02)
Sum Global Pooling	-	-
Common Head		
ConvBlock F = 1, K = 4, S = 1, D = 1	-	-

Table 3. We ablate on the semantics availability, the generator architecture and the discriminator architecture for adding objects on street scenes from Cityscapes w.r.t. FID score (lower is better)

	Discriminator			
	Full semantics		BBox semantics	
Generator	PatchGAN	SESAME	PatchGAN	SESAME
SPADEPix2PixHD	11.92	12.74	12.32	12.66
SESAME	11.95	11.64	11.13	**11.03**

Table 4. Object Replacement - Cityscapes. We show the performance of our SESAME model with our discriminator and the PatchGAN[3,5] discriminator as well as the percentage of user answers to the question: *Which image looks more photo-realistic?*

Discriminator	SSIM↑	FID↓	User Preference(%)
PatchGAN	0.390	10.63	45.03
SESAME	**0.433**	**9.3**	**54.97**

– Figure 3 shows results for editing ADE20k-Bedroom scenes[8,7].
– Figure 4 showcases examples of free-from semantic editing.
– On Figures 5 and 6 we can observe layout to image generation results for Cityscapes and ADE20k.

Label to Image Generation: Comparison with CC-FPSE: Similarly to SESAME, Liu *et al.* [4] developed an approach to tackle image generation conditioned on semantic layouts. They propose a generator architecture that learns to predict convolutional kernel weights conditioned on the semantic input. Moreover, they propose a feature pyramid semantics-embedding (FPSE) discriminator using an Encoder-Decoder architecture. Each upsampling layer outputs two per-patch score maps, one trying to measure the *realness* and one to gauge the *semantic matching* with the labels; the later is derived after a patch-wise inner product operation with the down-sampled semantic embeddings.

Their FPSE discriminator, while it also addresses the shortcomings of previous models, follows a different approach to our SESAME discriminator. Although they similarly aim to short-circuit the guidance of the semantic labels to the discrimination, they choose to do so by embedding the patch with a 1×1 convolution and down-sampling via average pooling the semantic layout to match the size of their image processing pipeline. As we explained in the main paper, the receptive field of a patch may contain a multitude of different semantic classes with a variety of compositions. Trivially down-sampling the semantic label can result into loss of information. Thus, we proposed a dedicated part of the discriminator to derive a meaningful representation for such an intricate semantic patch. Moreover, our model independently processes the semantic information for each scale. We experimented with their discriminator architecture along our SESAME generator but we were unable to achieve similar performance to our full method: SSIM 0.371, FID: 12.49, mIoU 64.24% and accuracy 85.93%.

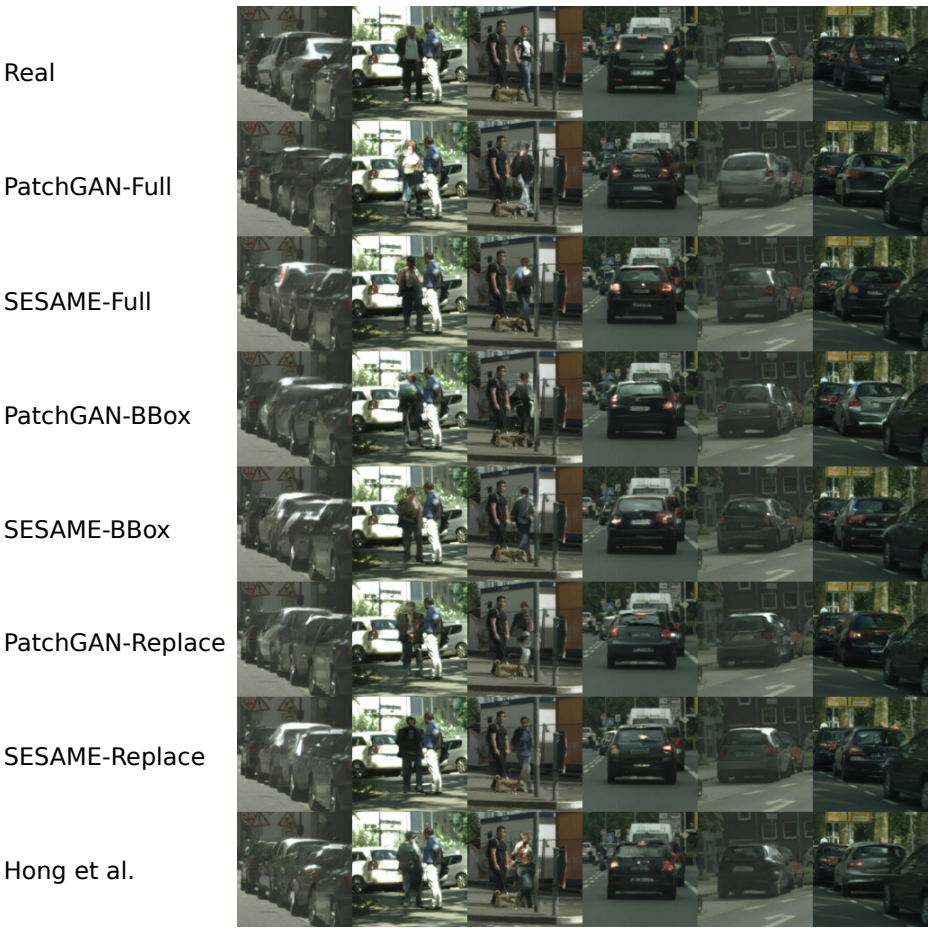

Fig. 1. Visual results for addition on Cityscapes[1]. We ablate on: (a) using the Full context, using only the labels of the rectangular areas to be edited and only replacing an object given its mask (b) generations due to training with the PatchGAN Discriminator and SESAME. Finally, we show the results produced by the method of Hong *et al.* [2], using the full semantics information

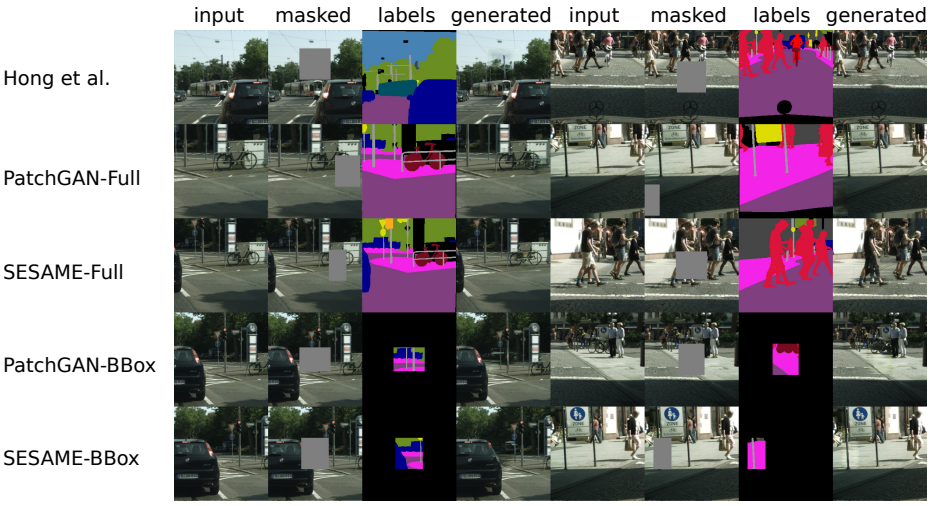

Fig. 2. Visual results for removal on Cityscapes[1]. We ablate on: (a) using the Full context and using only the labels of the rectangular areas to be edited (b) generations due to training with the PatchGAN Discriminator and SESAME. In the first row we show the results produced by the method of Hong *et al.* [2], using the full semantics information

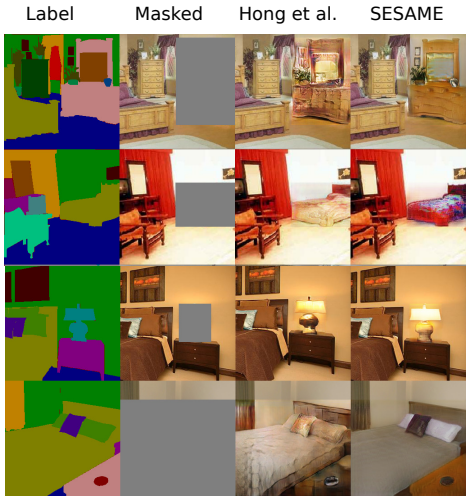

Fig. 3. Visual results for editing Bedroom scenes from ADE20K dataset. Here we are using the Full semantic information to alter the gray area

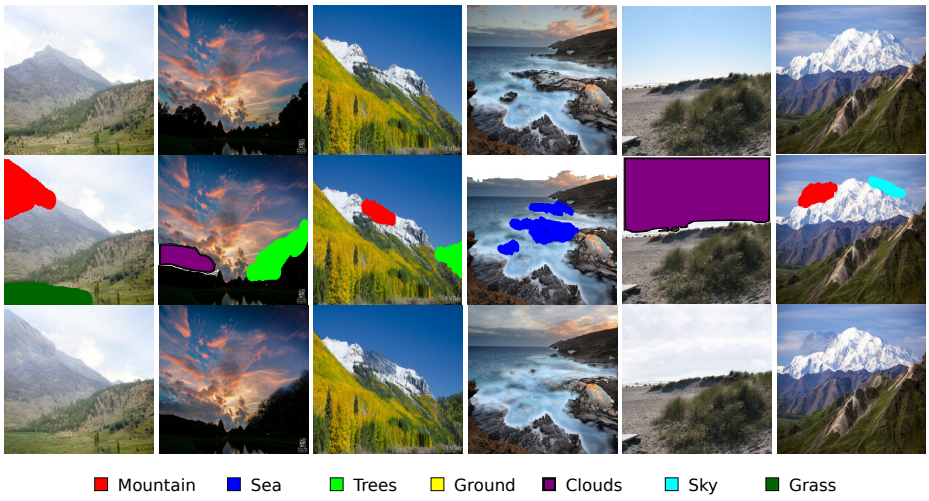

Fig. 4. Examples of free form editing using semantic brushes. Note that *snow* is a different semantic label from *mountain* and we can observe when *draw* with the mountain brush the model learned to differentiate this from snow. The model fails to correctly depict a *semantic concept* out of context or with an unexpected shape

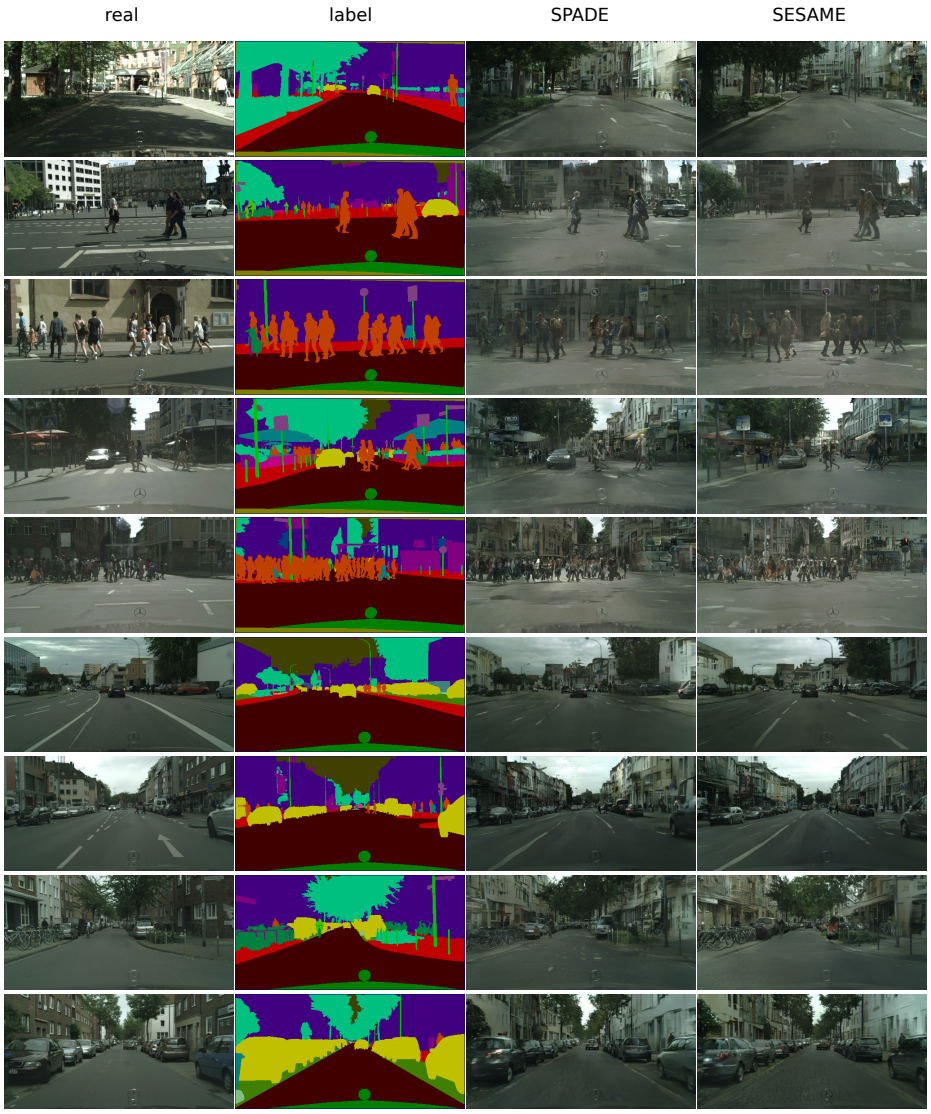

Fig. 5. Cityscapes[1]: Visual results for image generation conditioned on Semantic Labels. We showcase the results using the generator from SPADE[5] with the PatchGAN Discriminator(SPADE) and ours(SESAME)

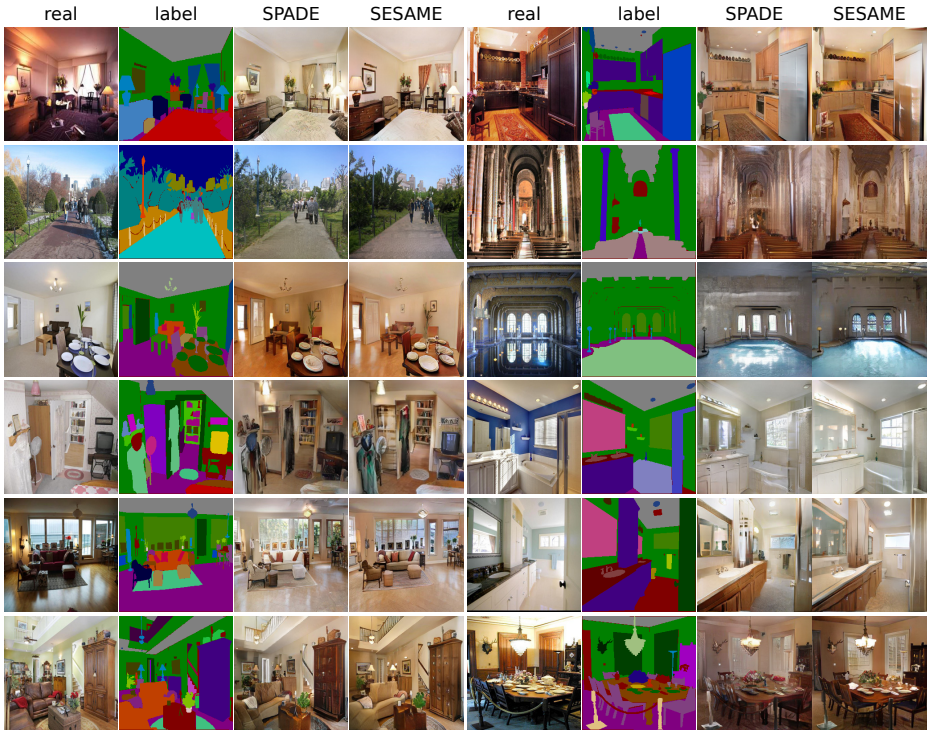

Fig. 6. Ade20k[8,7]: Visual results for image generation conditioned on Semantic Labels. We showcase the results using the generator from SPADE[5] with the PatchGAN Discriminator(SPADE) and ours(SESAME)

References

1. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
2. Hong, S., Yan, X., Huang, T.E., Lee, H.: Learning hierarchical semantic image manipulation through structured representations. In: Advances in Neural Information Processing Systems. pp. 2713–2723 (2018)
3. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
4. Liu, X., Yin, G., Shao, J., Wang, X., Li, H.: Learning to predict layout-to-image conditional convolutions for semantic image synthesis. In: Advances in Neural Information Processing Systems (2019)
5. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
6. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
7. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. arXiv preprint arXiv:1608.05442 (2016)
8. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)