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

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
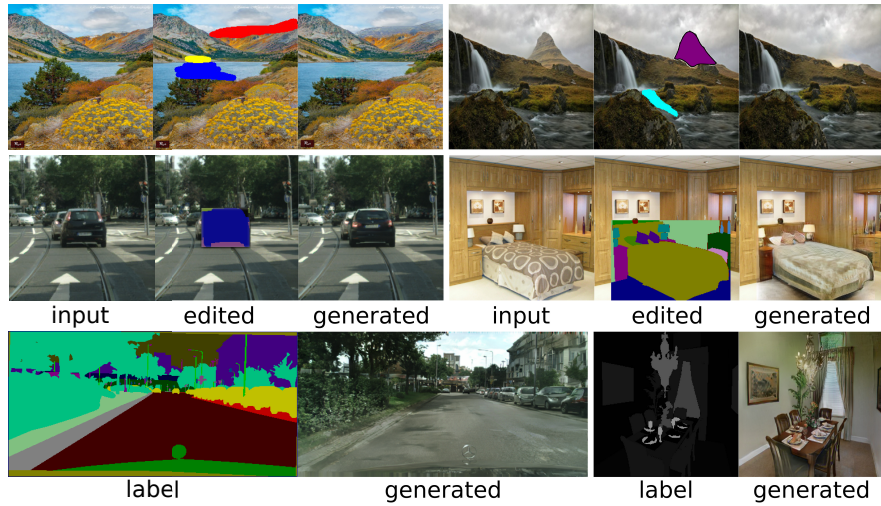[3] PSI, ESAT, KU Leuven, Belgium

Fig. 1. We assess SESAME on three tasks (a) image editing with free form semantic drawings (first row) (b) semantic layout driven semantic editing (second row) (c) layout to image generation with SESAME discriminator (third row)

Abstract. Recent advances in image generation gave rise to powerful tools for semantic image editing. However, existing approaches can either operate on a single image or require an abundance of additional information. They are not capable of handling the complete set of editing operations, that is addition, manipulation or removal of semantic concepts. To address these limitations, we propose SESAME, a novel generator-discriminator pair for **S**emantic **E**diting of **S**cenes by **A**dding, **M**anipulating or **E**rasing objects. In our setup, the user provides the semantic labels of the areas to be edited and the generator synthesizes the corresponding pixels. In contrast to previous methods that employ a discriminator that trivially concatenates semantics and image as an input, the SESAME discriminator is composed of two input streams that

independently process the image and its semantics, using the latter to manipulate the results of the former. We evaluate our model on a diverse set of datasets and report state-of-the-art performance on two tasks: (a) image manipulation and (b) image generation conditioned on semantic labels.

Keywords: Generative Adversarial Networks, Interactive Image Editing, Image Synthesis

1   Introduction

Image editing is a challenging task that has received increasing attention in the media, movies and social networks. Since the early 90s, tools like Gimp [38] and Photoshop [36] have been extensively utilized for this task. Yet, both require high level expertise and are labour intensive. Generative Adversarial Networks (GANs) [10] provide a learning-based alternative able to assist non-experts to express their creativity when retouching photographs. GANs have been able to produce results of high photo-realistic quality [19,20]. Despite their success in image synthesis, their applicability on image editing is still not fully explored. Being able to manipulate images is a crucial task for many applications such as autonomous driving [15] and industrial imaging [7], where data augmentation boosts the generalization capabilities of neural networks [1,49,9].

Image manipulation has been used in the literature to refer to various tasks. In this paper, we follow the formulation of Bau *et al.* [3], and define the task of semantic image editing as the process of adding, altering, and removing instances of certain classes or *semantic concepts* in a scene. Examples of such manipulations include but are not limited to: removing a car from a road scene, changing the size of the eyes of a person, adding clouds in the sky, etc. We use the term *semantic concepts* to refer to various class labels that can not be identified as objects, *e.g.* mountains, grass, etc.

Training neural networks for visual editing is not a trivial task. It requires a high level of understanding of the scene, the objects, and their interconnections [45]. Any region of an image added or removed should look realistic and should also fit harmoniously with the rest of the scene. In contrast to image generation, the co-existence of real and fake pixels makes the fake pixels more detectable, as the network cannot take the "easy route" of generating simple textures and shapes or even omit a whole class of objects [4]. Moreover, the lack of natural image datasets, where a scene is captured with and without an object, makes it difficult to train such models in a supervised manner.

One way to circumvent this problem is by inpainting the regions of an image we seek to edit. Following this scheme, we mask out and remove all the pixels we want to manipulate. Recent works [55,32,37,16] improve upon this approach by incorporating sketch and color inputs to further guide the generation of the missing areas and thus provide higher level control. However, inpainting can only tackle some aspects of semantic editing. To address this limitation, Hong *et al.* [12] manipulate the semantic layout of an image and subsequently, they utilize

for inpainting the image. Yet, this approach requires access to the full semantic information of the image, which is costly to acquire.

To this end, we propose SESAME, a novel semantic editing architecture based on adversarial learning, able to manipulate images based on a semantic input. In particular, our method is able to edit images with pixel-level guidance of semantic labels, permitting full control over the output. We propose using the semantics *only* for regions to be edited, which is more cost-efficient and produces better results in certain scenarios. Moreover, we introduce a new approach for semantics-conditioned discrimination, by utilizing two independent streams to process the input image and the corresponding semantics. We use the output of the semantics stream to adjust the output of the image stream. We employ visual results along with quantitative analysis and a human study to validate the performance and flexibility of the proposed approach.

2   Related Work

Generative Adversarial Networks [10] have completely revolutionized a great variety of computer vision tasks such as image generation [20,19,30], super resolution [48,27], image attribute manipulation [28,40] and image editing [12,3].

Initially, GANs were only capable of generating samples drawn from a random distribution [10], but soon multiple models emerged able to perform **conditional image synthesis** [29,33]. These approaches condition the generation on various types of information. For example, [29,31,57,5] synthesize images characterized by a single label. In a different setting, [39,58,59,52] employ a text to image pipeline to create an image based on a high-level description. Recently, many methods utilize information of a scene graph [18,2] and sketches with color [42] to represent where objects should be positioned on the output image. A more fine-grained approach aims to translate semantic maps, which carry pixel-wise information, to realistic looking images [14,47,35,22]. For all the aforementioned models, the user can control the output image by altering the conditional information. Nonetheless, they are not suitable for manipulating an existing image, as they do not consider an image as an input.

In **user-guided semantic image editing** the user is able to semantically edit an image by adding, manipulating, or removing semantic concepts [3]. Both GANPaint [3] and SinGAN [43] are able to perform such operations: GAN-Paint [3] by manipulating the neuron activations and SinGAN [43] by learning the internal batch statistics of an image. However, both are trained on a single image and require retraining in order to be applied to another, while our model is able to handle the manipulation of multiple images without retraining.

Another simple type of editing is inpainting [13,56,25], where the user masks a region of the image for removal and the network fills it accordingly to the image context. In its classic form the user does not have control over the generated pixel. To address this, other research works guide the generation of the missing areas using edges [55,32] and/or color [37,16] information.

Recently, semantic aware approaches for inpainting address object addition and removal. Shetty *et al.* [44] proposes a two-stage architecture to facilitate removal operations, with an auxiliary network predicting the objects' masks during training; at inference users provide them. Note that their model cannot handle object generation. Works in the object synthesis task are utilizing semantic layout information, a fine-grained guidance over the manipulation of an image. Yet, a subset of them is limited by generating objects from a single class [34,51] or placing prior fixed objects on the semantics plane [23]. Hong *et al.* [12] are able to handle both addition and removal, but require full semantic information of the scene to produce even the smallest change to an image. In contrast, our method requires only the semantics of the region to be edited.

The core of the majority of the aforementioned works rely on adjusting the generator for the task of image editing. Most recent models use a PatchGAN variant [14] which is able to discriminate on the high frequencies of the image. This is a desired attribute as conventional losses like *Mean Squared Error* and *Mean Absolute Error* can only convey information about the lower frequencies to the generator. PatchGAN can also be used for conditional generation of images on semantic maps, similar to our case study. Previous works targeting a similar problem concatenate the semantic information to the image and use it as an input to the discriminator. However, conventional conditional generation literature suggests that concatenation is not the optimal approach for conditional discrimination [39,33,31]. To address this, Liu *et al.* [26] 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. Rather than incorporating a semantics loss, we use semantics to guide the image discrimination. Our model incorporates conditional information by processing it separately from the image input. In a later stage of the network, the two processed streams are merged to produce the final output of the discriminator.

3   SESAME

In this work we describe a deep learning pipeline for semantically editing images, using conditional Generative Adversarial Networks (cGANs). Given an image I_{real} and a semantic guideline of the regions that should be altered by the network, denoted by M_{sem}, we want to produce a realistic output I_{out}. The real pixels values corresponding to M_{sem} are removed from the input image. The generated pixels in their place should be both true to the semantics dictated by the mask and coherent with the rest of the pixels of I_{real}. In order to achieve this, our network is trained end-to-end in an adversarial manner. The generator is a Encoder-Decoder architecture, with dilated convolutions [53] and SPADE [35] layers and the discriminator is a two stream patch discriminator.
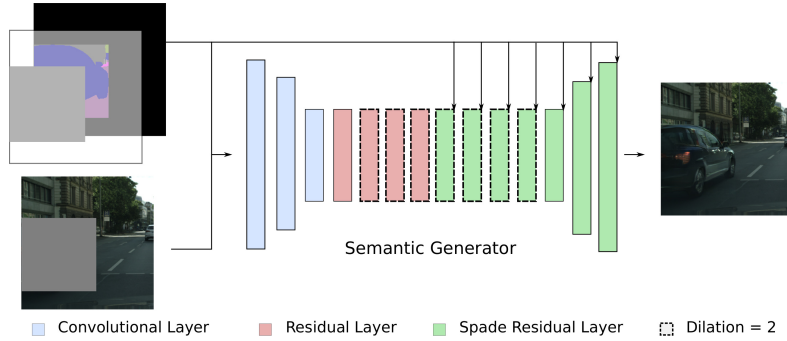
Fig. 2. The SESAME Generator aims to generate the pixels designated by the semantic mask so they are both (1) true to their label and (2) fit naturally to the rest of the picture. It is an encoder-decoder architecture with dilated convolutions to increase the receptive field as well as SPADE layers in the decoder to guide in-class generation

SESAME Generator. Semantically editing a scene is an Image to Image translation problem. We want to transform an image where we substituted some of the RGB pixels with an one-hot semantics vector. From the generator's output, only the pixels on the masked out regions are retained, while the rest are retrieved from the original image:

$$I_{gen} = G(I_m, M, M_{sem}), \tag{1}$$

$$I_{out} = I_{gen} \cdot M + I_{real} \cdot (1 - M). \tag{2}$$

This architecture has two goals: generated pixels should 1) be coherent with their real neighboring ones as well as 2) be true to the semantic input. To achieve these goals we adapt our generator from the network proposed by Johnson *et al.* [17] to fill the gaps: two downsampling layers, a semantic core made of multiple residual layers and two upsampling ones.

We conceptually divide our architecture into an encoder and a decoder part. The first extracts the contextual information of the pixels we want to synthesize. The seconds combines the semantic information using Spatially Adaptive De-Normalization [35] blocks to every layer. As the area to be edited can span over a large region, we would like the receptive field of our network to be relatively large. Thus, we use dilated convolutions in the last and first layers of the encoder and the decoder, respectively. A scheme of our SESAME generator can be seen in the Fig. 2, and for further details refer to the supplementary materials.

SESAME Discriminator. Layout to image editing can be seen as a sub-task of label to image translation. Inspired by the Pix2Pix [14], more recent approaches [47,35] employ a variation of the PatchGAN discriminator. The Markovian discriminator, as it is also called, was a paradigm shift that made
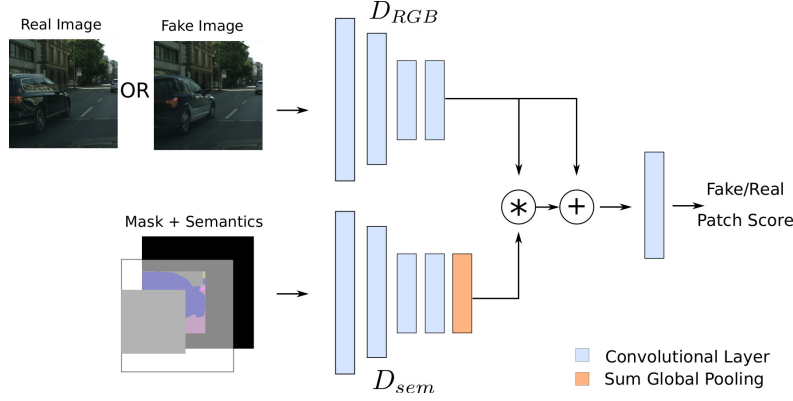
Fig. 3. The SESAME discriminator, in contrast to the commonly used PatchGAN, is handling the RGB Image and its Semantics independently. Before the last convolutional layer the two streams, D_{RGB} and D_{Sem}, are merged. The semantics stream is reduced via a *Sum Global Pooling* operation to a 2D matrix of spatial dimensions equal to the number of output patches. The feature vector of D_{RGB} at each path is scaled by D_{Sem} and a residual is added to product

the discriminator focus on the higher frequencies by limiting the attention of the discriminator into local patches, producing a different fake/real prediction value for each of them. The subsequent methods added a multiscale discrimination approach, the Feature Matching-Loss [47] and the use of Spectral Normalization [30] instead of Instance Normalization [35], which stabilized training and further improved the quality of the generated samples. However, the way that the conditional information was provided to the discriminator remained unchanged.

Label to image generation is a sub-task of conditional image generation. In this more general category of methods, the discriminator has evolved from the cGAN's input concatenation [29], to concatenating the class information with a hidden layer [39], and lastly, to take the form of the projection discriminator [31]. In the latter approach, the inner product of the embedding of the conditional information and a feature extracted from the hidden layers of the discriminator are summed with the output of the discriminator to produce the final prediction. Each step of the conditional discriminator evolution improved the results over the naive concatenation at its input [31]. On all aforementioned methods the discriminator produces, nonetheless, a scalar output for the whole image.

We aim to design a discriminator for label to image generation that combines the aforementioned attributes. On the one hand, it should preserve the ability of PatchGAN to discriminate on high-frequencies. On the other hand, we want to enforce the semantic information guidance on the discriminator's decision. If the pixels of the whole image shared semantic class, the projection discriminator would be easily extended to PatchGAN. In contrast, our case is characterized

by fine-grained per pixel semantics: each output patch encompasses a variety of classes and different compositions of them.

Our proposed SESAME discriminator is comprised by two independent streams that handle the RGB and Semantic Labels inputs. As Fig. 3 depicts, the two streams have identical architectures. Before the information is merged a *Sum Global Pooling* operation is applied to the output of the Semantic Stream. The output of the semantic stream is used to scale each output coming from the RGB stream. The resulted feature map is passed as input to a last 3×3 convolutional layer, which produces the final output. The process can be written as follows:

$$D(I, Sem) = Conv_{3\times3}(\, D_{RGB}(I_{out}) \cdot (1 + \sum_{channels} D_{sem}(Sem))\,), \qquad (3)$$

where the D_{RGB} is the output of the RGB stream and D_{sem} of the semantic stream before the *Global Sum Pooling*. We also integrate the changes made to PatchGAN by Pix2PixHD [47] and SPADE [35]. We use a multiscale discrimination scheme with squared patches and two different edge-sizes of 70 and 140 pixels, in order to provide also discrimination at a coarser level and Spectral Normalization. The input to the semantic stream is the same for both fake and real images discrimination, so we only need to calculate D_{sem} once. Moreover, it makes sense to apply the Feature Matching Loss only to the Feature Maps produced by the RGB stream.

Training Losses. We train the Generator in an adversarial manner using the following losses: Perceptual Loss [17], Feature Matching Loss [41], and Hinge Loss [24,46,30] as the Adversarial Loss. Early experiments with Style Loss [17] did not improve the results. Accordingly:

$$L_G = \lambda_{percept} \cdot L_{perc} + \lambda_{feat} \cdot L_{FM} - E_{z \sim p(z)}[D_k(I_{out}, M, M_{sem}))], \qquad (4)$$

Where each λ represents the relative importance of each loss component.

For the discriminator at each scale, the Hinge Loss takes the following form:

$$L_{D_k} = E_{z \sim q_{data}(x)}[min(0, -1 + D_k(I_{real}, M, M_{sem}))]+ \\ E_{z \sim p(z)}[min(1, -1 - D_k(I_{real}, M, M_{sem}))], \quad (5)$$

which is then combined to form the full discrimination loss,

$$L_D = L_{D_1} + L_{D_2}. \qquad (6)$$

4 Experiments

In Section 3 we described how the SESAME Generator can be used to semantic edit images for addition, manipulation, and removal, and how we designed the SESAME discriminator to tackle both image editing and generation. To elucidate the merits of our approach, we conducted a series of different experiments:

(a)   (b)   (c)   (d)   (a)   (b)   (c)   (d)

Fig. 4. Visual results of addition on Cityscapes: (a) input image (b) edited semantics (c) SESAME (d)Hong *et al.* [12]. Note that Hong *et al.* [12] require the whole semantics while we use only the semantics of the box

- In order to quantify the performance of our network we follow the data preparation and evaluation steps of Hong *et al.* [12], for generating and removing objects based on a given semantic layout.
- We train our model to permit free form semantic input from users to manipulate scenes and show qualitative results.
- We train our SESAME discriminator along with SPADE Generator for Label to Image Generation.

Implementation Details. For training we are using the Two Time-Scale Update Rule [11] to determine the scale between the learning rate of the generator and the discriminators, with $lr_{gen} = 0.0001$ and $lr_{disc} = 0.0004$. We train for 200 epochs. After 100 we start to linearly decay the learning rates to 0. For our generator losses we multiplied the Feature Matching Loss and Perceptual loss by a factor of 10 before adding them to the adversarial loss. We use the Adam optimizer [21] with coefficient values of $b_1 = 0$ and $b_2 = 0.999$, similar to [35].

Datasets. In line with the literature we conduct experiments on:

- **Cityscapes [8].** The dataset contains 3,000 street-level view images of 50 different cities in Europe for the training set and 500 images for the validation set. The images are accompanied by fine-grained information of the per-pixel semantics and instance segmentation with original resolution of the images is 2048×1024 pixels. For addition and removal, we downsample to 1024×512 pixels before patches of 256×256 pixels are extracted. Following Hong *et al.* [12], we choose 10 of the 30 available semantics classes as foreground objects, *e.g. pedestians, cars, bicycles, etc.*. For generation we resize the image to 512×256 pixels.
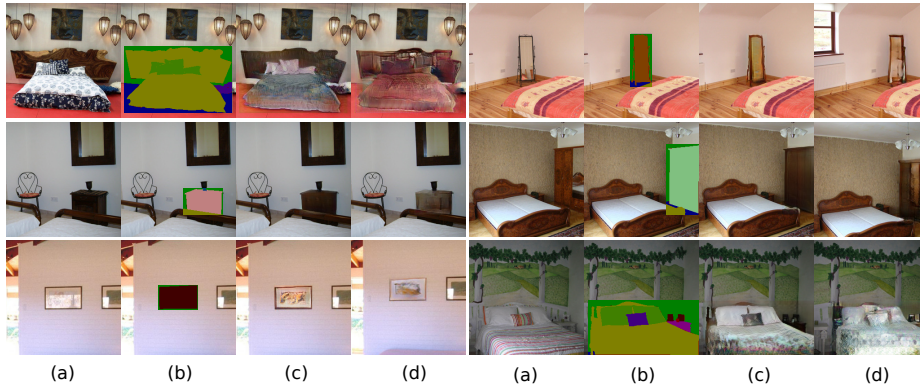
Fig. 5. Visual results of addition on ADE20k: (a) input image (b) edited semantics (c) SESAME (d) Hong *et al*. [12]. In this setting we use the full semantic information to guide the editing

- **ADE20K [60,61]** ADE20K has over 20,000 images together with their detailed semantics for 150 different semantic classes. In addition, 2,000 more images are offered for validation. The whole dataset is used for the generation task. For manipulation, following Hong *et al*. [12], we experiment on a subset of the ADE20K dataset comprised of bedroom scenes. Similarly to Cityscapes, 31 objects are chosen as foreground objects. In total we consider 49 semantic categories for training and evaluation.
- **Flickr-Landscapes Datasets [35]** Similar to SPADE [35], we first scrapped 200,000 images from flickr with only landscape constraint. As our main purpose is to show image editing over significant areas within a landscape, we use a DeepLab-v2 [6] network trained on COCO-Stuff in order to extract images that contain at least 80% pixels of clouds, mountains, water, grass, etc. After post-processing, our curated dataset consists of 7367 training and 500 validation images with their corresponding segmentation for 17 different semantic classes.

Data Preprocessing. Free-form semantic editing is not trivial to achieve. The model can easily overfit on mask shapes used during training. In order to train our free-form semantic editing experiments, we randomly draw a box mask in conjunction with random strokes [55] with 70% chance, otherwise we drop all the pixels belonging to a semantic class of the training image.

For layout driven editing, we extend the data pre-processing scheme introduced by Hong *et al*. [12]. A rectangular area is removed from the input image and we try to inpaint it using the semantic labels. To train the addition operation, they extract the boundary boxes based on the instances of the foreground classes. For the removal subtask, they randomly choose and remove blocks to train the network to inpaint background classes. While this makes sense for a

Fig. 6. Free-form image manipulation. The user can select a semantic brush and paint over the image to adjust as they see fit

dataset like ADE20k where the foreground objects can be found anywhere in the pictures, in Cityscapes the foreground objects placement follow certain distributions [23]. Thus, we only extract a randomly chosen rectangular area if it contains at least a pixel of *ground, road, sidewalk and parking.*

Quantitative Results. For measuring the performance of our network, we combine the evaluation approach of previous methods [12,35]. To assess the perceptual quality of our synthesis we use the Frechet Inception Distance (FID) [11]. We compare the mean Intersection over Union (mIoU), and the pixel-accuracy loss between the ground truth semantic layout and the inferred one. We infer the semantic labels of the generated images using pretrained semantic segmentation networks [53,54,60,61]. In order to maintain consistency, we chose the same object for validating all our experiments. Additionally, in our comparison with Hong *et al.* [12], we compute the Structural Similarity Index (SSIM) [50] between each $\langle I_{real}, I_{out} \rangle$ pair, taking into account only the generated pixels. Naturally, as in the editing task only a small percentage of image pixels are changed we expect better results than those in the Generation experiments, but also better methods yield larger performance gains when tackling the latter.

Our Baselines. For our semantic image editing baseline we are using the work of Hong *et al.* [12]. They introduced a hierarchical model to tackle the task of

Table 1. Addition Results for Cityscapes and ADE20k dataset. We ablate on the Generator and Discriminator architecture as well as the semantic availability. For the SSIM, accuracy and mIoU higher is better, while for FID, lower is better.

Generator	Disc	Labels	Cityscapes				ADE20k			
			SSIM	accu	mIoU	FID	SSIM	accu	mIoU	FID
Hong *et al.* [12]	PatchGAN	Full	0.377	83.8	60.7	12.11	0.205	92.2	34.6	28.48
Hong *et al.* [12]	PatchGAN	BBox	0.303	85.2	58.8	13.80	0.183	92.7	35.3	28.36
Hong *et al.* [12]	SESAME	Full	0.396	86.0	64.0	11.76	0.192	91.3	34.1	28.55
SESAME	PatchGAN	BBox	0.375	86.0	64.5	11.13	0.193	92.2	35.7	28.16
SESAME	SESAME	BBox	**0.410**	**86.0**	**65.3**	**11.03**	**0.209**	**93.3**	**37.1**	**26.66**

Table 2. Removal results for Cityscapes and ADE20k datasets. For the SSIM, accuracy and mIoU higher is better, while for FID, lower is better.

Method	Cityscapes				ADE20k			
	SSIM↑	accu↑	mIoU↑	FID↓	SSIM↑	accu↑	mIoU↑	FID↓
Hong *et al.* [12]	0.584	83.9%	65.3%	10.34	0.456	91.7%	40.0%	24.98
SESAME	**0.797**	**85.0%**	**67.6%**	**7.43**	**0.491**	**92.3%**	**41.6%**	**23.30**

image editing. On the first stage, they inpaint the semantic classes of an image with a missing region. Then they combine the predicted output with the ground truth and after concatenating the real image with the missing pixels, they use their second stage model to fill the image. Similar to their work, we focus on the *mask to image generation* task and compare our model against their image generator trained on the ground truth labels. Their approach consists of an encoder and a decoder. The encoder has two input streams where the image and the semantics are processed separately and are then *fused* based on the mask of the object location. The result of the fusion is then passed to an image decoder which produces the end result. The generator is trained in conjunction with a PatchGAN discriminator. We use different architectures and largely decrease the number of parameters for the generator and have a larger discriminator as shown in Table 4. However, during inference time only the generator is used. Reduced number of parameters for the generator is clearly beneficial during execution. For Image Generation we compare against SPADE [35] and CC-FPSE [26].

Addition and Removal of Objects To demonstrate the ability of our network to perform well both on the addition and the removal part we compare on both tasks separately. The computed metrics for these cases can be found in Tables 1 and 2, respectively.

In the visual results we can observe that objects look sharper and their features are more distinctive. Furthermore, as Figures 4 and 5 illustrate, our method generates different patterns for different clothes, and cars in which the windows are not mixed with the rest of the car. Besides our better numerical results, our user study further illustrates the superiority of our approach. In

Table 3. Comparison in number of parameters

| | Parameters in millions | |
Method	Generator	Discriminator
Hong *et al.* [12]	190m	5.6m
SESAME	20.5m	11.1m

Table 4. Layout to image generation results. For mIoU and accu, higher is better, while for FID, lower is better

| | | Cityscapes | | | ADE20k | | |
Generator	Discriminator	mIoU	accu	FID	mIoU	accu	FID
Pix2PixHD	PatchGAN	58.3	81.4	95.0	20.3	69.2	81.8
Pix2PixHD	SESAME	59.6	81.1	55.4	49.0	85.5	36.8
SPADE	PatchGAN	62.3	81.9	71.8	38.5	79.9	33.9
CC	FPSE	65.5	82.3	54.3	43.7	82.9	**31.7**
SESAME	SESAME	**66.0**	**82.5**	**54.2**	**49.0**	**85.5**	31.9

the case of removal, artifacts of the *BBox* are commonly left in picture by the method of Hong *et al.* [12], whilst in our case this effect is difficult to notice.

Labels to Image Generation The SESAME Discriminator is designed to tackle the shortcomings of the naive concatenation of an image and its semantics label when generating images. We measure the performance on Labels to Image Generation against SPADE [35] using the same generator and against CC-FPSE of Liu *et al.* [26] . Please refer to SM for the differences in our approaches. The results for Cityscapes and ADE20k datasets can be found on Table 4 and Fig. 7.

Free-Form Semantic Image Editing The user selects a brush of a semantic class and paints over the image. The pixels that are painted over are removed from the image and SESAME is filling the gaps based on the painted semantic guidance. Examples of hand painted masks and corresponding results can be seen on Fig. 6. Additionally, the context is very important for the label we want to add: a patch of grass cannot be drawn in the middle of the sky. More results can be found in the supplementary materials.

Ablation Study SESAME incorporates a Generation/Discrimination pair able to edit a scene by only considering the Semantics of regions in the image that the user seeks to edit. In order to showcase the benefits of our approach we ablate the performance of our architecture by varying (a) the generator architecture, (b) the discriminator architecture for both image manipulation and generation and (c) the available semantics only for manipulation, by utilizing either the *Full* semantic layout or the semantics of the rectangular region we want to edit, which we refer to as *BBox* Semantics.

Fig. 7. Label to Image Generation results. For each triplet of images we are showing the semantic layout input (left), generation using PatchGAN (center) and SESAME (right) discriminator on top of SPADE generator

As we observe in Table 1 and Table 4 in almost all cases using the SESAME discriminator improves the performance compared to the commonly used Patch-GAN. Our generator is producing better visual quality results(mIoU, FID) but is lagging in fidelity(SSIM) when compared with the one from Hong *et al.* [12]. The partial *BBox* semantics improve the performance in the case of Cityscapes dataset but full semantics work better for ADE20k. We observe the same effect when using full semantics for SESAME.

In another series of experiments, we substituted our Semantics merging operation. Instead of applying *Sum Global Pooling*, we experiment with 1) concatenating the two streams of information and 2) calculating their element-wise product resulted in lower FID score compared to the proposed approach, 11.96 and 12.02 respectively.

User Study We employed Amazon Mechanical Turk[4] for the two experiments of our user study. For each of them we took 100 samples from our validation set and asked 20 Turkers: *Which among the images looks more photorealistic?*

The first experiment presented the Turkers with three options: our method with access to only the *BBox* information and both ours and Hong *et al.* [12]

---

[4] https://www.mturk.com

Table 5. User Study Results: *Which image is the most photorealistic?* The first study invited the users to choose between Hong *et al.* [12]with full semantics information and ours with full and bbox semantics, respectively. The second study invited to choose between the results produced by our SESAME and the PatchGAN discriminator, for different availabilities of semantics

User Study I		User Study II		
Setting	Preference [%]	Discriminator:	SESAME	PatchGAN
Hong *et al.* [12]	22.50	FullContext	**56.67**	43.33
SESAME w Full	35.83	BBoxContext	**61.04**	38.96
SESAME w BBox	**41.67**			

model using the *Full* Semantics. As shown in Table 5 the results of our SESAME approach were clearly preferred by the users over the results of our baseline. Moreover, in agreement with our quantitative analysis, the proposed scaling scheme in our discriminator benefits from less irrelevant semantic information. Another group of settings compared the results when the PatchGAN is used instead of our SESAME discriminator. The results consistently show that the independent processing of the semantic information leads to better perceptual quality of the results; they are picked more often by the human subjects.

5   Conclusion

In this work, we introduce SESAME a novel method for semantic image editing covering the complete spectrum of adding, manipulating, and erasing operations. Our generator is capable of manipulating an image by only conditioning on the semantics of the regions the user seeks to edit, namely without requiring the information about the full layout. Our discriminator processes the semantic and image information in separate streams and overcomes the limitations of the concatenating approach inherent in PatchGAN. SESAME produces state-of-the-art results on the tasks of (a) semantic image manipulation and (b) layout to image generation and permits the user to edit an image by intuitively painting over it. As a future research direction, we plan to extend this work on image generation conditioned on other types of information, *e.g.* scene graphs could also benefit from our two-stream discriminator. We refer to supplementary material for more details and to OpenSESAME for the code and the pretrained models.

References

1. Antoniou, A., Storkey, A., Edwards, H.: Data augmentation generative adversarial networks. arXiv preprint arXiv:1711.04340 (2017)
2. Ashual, O., Wolf, L.: Specifying object attributes and relations in interactive scene generation. In: Proceedings of the International Conference Computer Vision (ICCV) (2019)
3. Bau, D., Strobelt, H., Peebles, W., Wulff, J., Zhou, B., Zhu, J., Torralba, A.: Semantic photo manipulation with a generative image prior. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) **38**(4) (2019)
4. Bau, D., Zhu, J.Y., Wulff, J., Peebles, W., Strobelt, H., Zhou, B., Torralba, A.: Seeing what a gan cannot generate. In: Proceedings of the International Conference Computer Vision (ICCV) (2019)
5. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: Proceedings of the International Conference on Learning Representations (ICLR) (2019)
6. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence **40**(4), 834–848 (2017)
7. Cognex: Visionpro vidi: Deep learning-based software for industrial image analysis. https://www.cognex.com/products/machine-vision/vision-software/visionpro-vidi, accessed: 2019-03-05
8. 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)
9. Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., Greenspan, H.: Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification. Neurocomputing **321**, 321–331 (Dec 2018)
10. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems. pp. 2672–2680 (2014)
11. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. pp. 6626–6637 (2017)
12. 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)
13. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. ACM Transactions on Graphics (ToG) **36**(4), 1–14 (2017)
14. 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)
15. Janai, J., Güney, F., Behl, A., Geiger, A.: Computer vision for autonomous vehicles: Problems, datasets and state of the art. arXiv preprint arXiv:1704.05519 (2017)
16. Jo, Y., Park, J.: Sc-fegan: Face editing generative adversarial network with user's sketch and color. In: Proceedings of the International Conference Computer Vision (ICCV) (2019)

17. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Proceedings of the European Conference on Computer Vision (ECCV) (2016)
18. Johnson, J., Gupta, A., Fei-Fei, L.: Image generation from scene graphs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
19. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: Proceedings of the International Conference on Learning Representations (ICLR) (2018)
20. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
21. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980 (2014)
22. Lee, C.H., Liu, Z., Wu, L., Luo, P.: Maskgan: Towards diverse and interactive facial image manipulation. arXiv preprint arXiv:1907.11922 (2019)
23. Lee, D., Liu, S., Gu, J., Liu, M.Y., Yang, M.H., Kautz, J.: Context-aware synthesis and placement of object instances. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems. pp. 10393–10403 (2018)
24. Lim, J.H., Ye, J.C.: Geometric gan. arXiv preprint arXiv:1705.02894 (2017)
25. Liu, G., Reda, F.A., Shih, K.J., Wang, T.C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 85–100 (2018)
26. 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)
27. Lugmayr, A., Danelljan, M., Timofte, R., Fritsche, M., Gu, S., Purohit, K., Kandula, P., Suin, M., Rajagopalan, A., Joon, N.H., et al.: Aim 2019 challenge on real-world image super-resolution: Methods and results. In: Proceedings of the International Conference Computer Vision (ICCV), Advances in Image Manipulation Workshop (2019)
28. Mao, Q., Lee, H.Y., Tseng, H.Y., Ma, S., Yang, M.H.: Mode seeking generative adversarial networks for diverse image synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1429–1437 (2019)
29. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
30. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: Proceedings of the International Conference on Learning Representations (ICLR) (2018)
31. Miyato, T., Koyama, M.: cGANs with projection discriminator. In: Proceedings of the International Conference on Learning Representations (ICLR) (2018)
32. Nazeri, K., Ng, E., Joseph, T., Qureshi, F.Z., Ebrahimi, M.: Edgeconnect: Generative image inpainting with adversarial edge learning. arXiv preprint arXiv:1901.00212 (2019)
33. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier gans. arXiv preprint arXiv:1610.09585 (2016)
34. Ouyang, X., Cheng, Y., Jiang, Y., Li, C.L., Zhou, P.: Pedestrian-synthesis-gan: Generating pedestrian data in real scene and beyond. arXiv preprint arXiv:1804.02047 (2018)

35. 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)
36. Photoshop: version 21.1.0. Adobe Inc., San Jose, California, U.S. (2020)
37. Portenier, T., Hu, Q., Szabó, A., Bigdeli, S.A., Favaro, P., Zwicker, M.: Faceshop: Deep sketch-based face image editing. ACM Trans. Graph. **37**(4) (2018)
38. Program, G.I.M.: version 2.10.18. The GIMP Development Team (2018)
39. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text-to-image synthesis. In: Proceedings of the International Conference on Machine Learning (ICML) (2016)
40. Romero, A., Arbeláez, P., Van Gool, L., Timofte, R.: Smit: Stochastic multi-label image-to-image translation. In: Proceedings of the International Conference Computer Vision (ICCV), Workshops (2019)
41. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., Chen, X.: Improved techniques for training gans. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems. pp. 2234–2242 (2016)
42. Sangkloy, P., Lu, J., Fang, C., Yu, F., Hays, J.: Scribbler: Controlling deep image synthesis with sketch and color. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
43. Shaham, T.R., Dekel, T., Michaeli, T.: Singan: Learning a generative model from a single natural image. In: Proceedings of the International Conference Computer Vision (ICCV) (2019)
44. Shetty, R., Fritz, M., Schiele, B.: Adversarial scene editing: Automatic object removal from weak supervision. In: Bengio, S., Wallach, H., Larochelle, H., Graumann, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems. pp. 7716–7726. Curran Associates, Montréal, Canada (2018)
45. Shetty, R., Schiele, B., Fritz, M.: Not using the car to see the sidewalk: Quantifying and controlling the effects of context in classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2019)
46. Tran, D., Ranganath, R., Blei, D.: Hierarchical implicit models and likelihood-free variational inference. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. pp. 5523–5533 (2017)
47. 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)
48. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Change Loy, C.: Esrgan: Enhanced super-resolution generative adversarial networks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 0–0 (2018)
49. Wang, Y.X., Girshick, R., Hebert, M., Hariharan, B.: Low-shot learning from imaginary data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2018)
50. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., et al.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
51. Wu, S., Lin, S., Wu, W., Azzam, M., Wong, H.S.: Semi-supervised pedestrian instance synthesis and detection with mutual reinforcement. In: Proceedings of the International Conference Computer Vision (ICCV) (2019)

52. Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., He, X.: Attngan: Fine-grained text to image generation with attentional generative adversarial networks. arXiv preprint arXiv:1711.10485 (2017)
53. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: Proceedings of the International Conference on Learning Representations (ICLR) (2016)
54. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
55. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. arXiv preprint arXiv:1806.03589 (2018)
56. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5505–5514 (2018)
57. Zhang, H., Goodfellow, I.J., Metaxas, D.N., Odena, A.: Self-attention generative adversarial networks. arXiv preprint arXiv:1805.08318 (2018)
58. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.: Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: Proceedings of the International Conference Computer Vision (ICCV) (2017)
59. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N.: Stackgan++: Realistic image synthesis with stacked generative adversarial networks. arXiv preprint arXiv:1710.10916 (2017)
60. 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)
61. 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)