Region-Semantics Preserving Image Synthesis

Kang-Jun Liu, Tsu-Jui Fu, and Shan-Hung Wu

National Tsing Hua University, Taiwan, R.O.C.
{kjliu,zrfu}@datalab.cs.nthu.edu.tw, shwu@cs.nthu.edu.tw

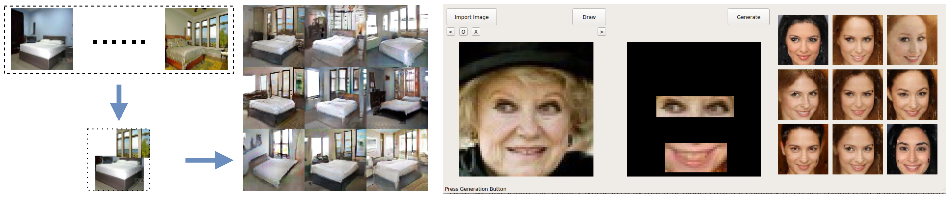
Abstract. We study the problem of *region-semantics preserving* (RSP) image synthesis. Given a reference image and a region specification R, our goal is to train a model that is able to generate realistic and diverse images, each preserving the same semantics as that of the reference image within the region R. This problem is challenging because the model needs to (1) understand and preserve the *marginal semantics* of the reference region; i.e., the semantics excluding that of any subregion; and (2) maintain the compatibility of any synthesized region with the marginal semantics of the reference region. In this paper, we propose a novel model, called the *fast region-semantics preserver* (Fast-RSPer), for the RSP image synthesis problem. The Fast-RSPer uses a pre-trained GAN generator and a pre-trained deep feature extractor to generate images without undergoing a dedicated training phase. This makes it particularly useful for the interactive applications. We conduct extensive experiments using the real-world datasets and the results show that Fast-PSPer can synthesize realistic, diverse RSP images efficiently.

1 Introduction

Image synthesis is a long-standing goal in computer vision, graphics, and machine learning. Recent advances in image modeling with neural networks [26,18,7,24,14,8] have made it feasible to generate photorealistic and diverse/creative images. While such unconditional models are fascinating, many practical applications of image synthesis require a model to be conditioned on prior information, motivating further studies on *controlling* the synthesis.

One common way to control the model is to use a *reference image*; that is, given an image as the input, the model is expected to generate images that resemble the input image. This strategy advances the state-of-the-art in density estimation [2], compression [30], in-painting [23,12], super-resolution [17] and image-to-image translation [13,35]. However, the reference image may be a too stringent control that limits the *diversity* of synthesized images. In many other applications, such as interactive photo editing [1] and stimuli generation in psycho-physical experiments [6], diversity is a key to success. There is a need for a new form of control that guides the model in a *soft* manner.

In this paper, we study the problem of *region-semantics preserving* (RSP) image synthesis, as shown in Figure 1(a). Specifically, given one or more reference images and the region specification R, our goal is to train a model that is able to



(a) Input/output

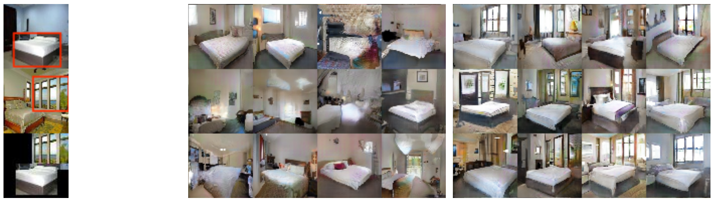
(b) Interactive editor

Fig. 1. Region semantic preserving (RSP) image synthesis problem. (a) A user provides reference regions R, which can be copied from different sources, then get synthesized, complete images preserving the semantics behind the reference regions. (b) An interactive editor based on real-time RSP image synthesis.

generate realistic and diverse images, each showing the same semantics as that of the reference image within the region R. We call the pixels enclosed by R in the reference image and synthesized image the *reference region* and *synthesized region*, respectively. Depending on the application, R could be specified in either a coarse grain (coordinates of a bounding box) or fine grain (pixel identifiers) manner. For example, one can ask the model to synthesis images with the semantic “happy mouth plus female eyes” by giving the reference regions shown in Figure 1(b). Note that the mouth or eyes in a synthesized region need *not* be identical to that in the reference region since only the *semantics* is preserved. One can also obtain many bedroom images with the same partial layout by giving some reference bedroom regions, as shown in Figure 1(a). Note that in an image, the semantics of a region R may be closely related to that of other regions residing either inside or outside R. So, to preserve the semantics of the reference region, the model will generate images that are compatible with the semantics everywhere. In other words, although being regional, the reference region is able to guide *all* pixels in a synthesized image, and we can expect “bedroom-only” decorators to appear in Figure 1(a) and “female faces” in Figure 1(b).

The RSP image synthesis is a challenging problem because when generating an image, the model needs to (1) understand and preserve the *marginal semantics* of the reference region. By “marginal” we mean that the semantics excludes the meaning of any other region (residing in or out R). For example, in Figure 1(a), the marginal semantics of R is “a bedroom with partial layout” despite that any subregion of R could have its own meaning (e.g., a bed, scene outside the window, etc.); (2) maintain the compatibility of any generated region with the marginal semantics of the reference region. To the best of our knowledge, the Ladder-VAEs [29,22] are the only off-the-shelf models that can be used to synthesize RSP images. Another solution could be a modified generative adversarial network [7] (GAN). However, in practice, these approaches usually produce images with sub-optimal quality and/or are very slow to train and run.

Here, we propose a novel model, called the *fast region-semantics preserver* (Fast-RSPer), that can produce high-quality RSP images and are fast to run. The Fast-RSPer is based on GANs but unlike most GANs where a generator



	R	iGAN	Fast-RSPer
Pixel-/Latent-Feature-Loss		0.128 / 507.99	0.172 / 428.64

Fig. 2. Pixel- vs. semantics-preserving. iGAN [34], whose goal is to align pixels of synthesized images with those in R, gives lower pixels-losses but higher feature losses at a deep CNN layer. As the pixels in R become more diverse, the semantics is harder to preserve by aligning pixels.

and a discriminator are trained jointly, it uses a pre-trained generator and feeds the output (images) into a pre-trained deep feature extractor. Given a reference image and R, the Fast-RSPer synthesis an image by finding (using the gradient descent) an input variable $\boldsymbol{z}$ for the generator such that, at a deep layer where neurons capture the semantics of the reference R, the feature extractor maps the synthesized region to features similar to those of the reference region. Since both the generator and feature extractor are pre-trained, the Fast-RSPer has no dedicated training phase and can generate images efficiently. Furthermore, it has been shown [20] that a properly trained generator can map $\boldsymbol{z}$ to a high quality image at a high resolution.

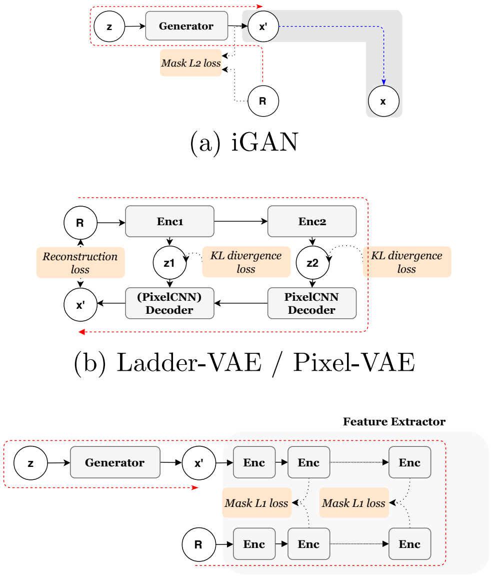
However, in practice there are still many technical problems to solve to synthesize a high-quality RSP image. The first problem is how to align the semantics of R with that of the synthesized region while maintaining its compatibility with other regions. Fast-RSPer uses both the *macro*- and *micro*-alignment techniques to achieve these goals. Second, the gradient descent algorithm may find $\boldsymbol{z}$'s (for different images) close to each other, resulting in the lack of diversity among multiple synthesized images. Fast-RSPer adds *gradient noises* to create visual diversity. Furthermore, a pre-trained generator may fail to generate satisfactory images in some cases. We investigate the cause and discuss pitfalls to avoid when using a pre-train generator. Following summarizes our contributions:

- We propose the the problem of region-semantics preserving (RSP) image synthesis that allows a user to easily guide the synthesis using a reference image in a soft manner.
- We present the Fast-RSPer model that generates realistic, diverse RSP images in near realtime.
- We discuss practical tricks to use and pitfalls to avoid to synthesize high-quality RSP images.
- We conduct experiments using the real-word datasets and the results show that the Fast-PSPer can synthesize photorealistic, diverse RSP images efficiently, enabling new interactive applications.

2 Related Work

Studies have explored different ways other than using a reference image to control the image synthesis. Mirza et al. [19] train a GAN by feeding the class labels and show that the generator can learn to produce images conditioned on a given label. Reed et al. [25] extend the GAN architecture to synthesis images from text. Güçlütürk et al. [9] use deep neural networks for inverting face sketches to synthesize face images. Studies [4,31,33] propose models that synthesize images based on domain-specific attributes such as the orientation with respect to the camera. The goals of the above studies are orthogonal to ours—to present an easy-to-use, powerful, soft control over the reference image.

The style transfer [15,5] and image analogies [11] can be seen as ways to synthesize images using a soft control. The former expects a *style image* to come along with the reference image (serving as the content image); while the latter takes *a pair of images* as additional input and synthesis images following the example analogy between the images in the pair. These soft controls (i.e., the style image and the image pair) are very different from the reference region R proposed in this paper, and their applications normally don't emphasize on diversity of the synthesize results.



(a) iGAN

(b) Ladder-VAE / Pixel-VAE

(c) Fast-RSPer

Fig. 3. Model architecture of (a) Ladder-VAE and Pixel-VAE; (b) iGAN; (c) Fast-RSPer. The $\boldsymbol{x}$ is a reference image, R is the reference region, $\boldsymbol{x}'$ is a synthesized image, and $\boldsymbol{z}$ is a latent variable. Red dashed lines denote the computation flows for generating an image while the blue line denotes post-processing.

iGAN. This idea of using a pre-trained generator to speed up image synthesis is not new and has appeared in iGAN [34], whose model architecture is shown in Figure 3(a). However, the work studied a different problem, where there are two types of input: one the "base image" and another the "changes" to the base image, and the goal is to synthesis images that preserve the semantics of the base image while having the changes applied. The optimization procedure in [34] finds a $\boldsymbol{z}$ so that the generated image $G(\boldsymbol{z})$ satisfies two constraints: (1) the semantics of the base image is preserved and (2) the changes are applied to $G(\boldsymbol{z})$. For the constraint (2), the changes are applied *at the pixel level* (see the data term in Eqs. (4) and (5) in [34]), which is different from our region-semantic-preserving image synthesis problem where only the latent marginal semantic should be preserved. In iGAN, the $\boldsymbol{z}$ can only output images that helps an *auxiliary algorithm* measure and apply the same amount of geometric and color changes to the original base image to produce the final result. The images synthesized by iGAN and Fast-RSPer has different characteristics, as shown in Figure 2. When pixels in R (e.g., window scene) are diverse, images synthesized by iGAN may end up preserving the *average* of pixels in R, destroying the semantics.

Ladder-VAEs/Pixel-VAE. To overcome the challenges of the RSP image synthesis discussed in Section 1, an image model needs to be able to extract and manipulate the semantics of *any* region in an image. One way to achieve this is to use the Ladder-VAE [29], whose architecture is shown in Figure 3(b). Assuming that the encoding and decoding of an image follow two multi-layer distributions recursively parametrized by a series of shared latent variables $\boldsymbol{z}^{(i)}$'s , the Ladder-VAE learns the $\boldsymbol{z}^{(i)}$'s from a training dataset and synthesizes images following the multi-layer decoding distribution parametrized by the learned $\boldsymbol{z}^{(i)}$'s. To preserve the marginal semantics of the reference region, one can encode the reference image using the Ladder-VAE encoder, identify the dimensions of $\boldsymbol{z}^{(i)}$ parametrizing the reference region at a deep layer i, and then *fix* the decoding distribution along those dimensions in the Ladder-VAE decoder when synthesizing images. To create diversity while maintaining the compatibility of any region in a synthesized image with the reference region, one can add *sampling variances* to decoding distributions parametrized the rest dimensions of $\boldsymbol{z}^{(i)}$ and all dimensions of $\boldsymbol{z}^{(j)}$, $j \neq i$. However, the VAE variants are known to produce blurry images [14]. To solve this problem, one can use the Pixel-VAE [22] that replaces the decoder of Ladder-VAE with an autoregressive decoder based on the Pixel-CNN [23]. But this creates another problem that the decoder needs to generate pixels one-by-one (due to the autoregressive nature) and becomes very slow. Furthermore, the autoregressive decoder may occasionally generate “bad” pixels (due to the sampling for diversity) that degrades the quality of following pixels and creates local artifacts in the synthesized images.

3 Fast RSP Image Synthesis

In this section, we present a fast, end-to-end algorithm [7] for synthesizing RSP images based on GANs. As compared to (non-autoregressive) VAEs, GANs are able to produce sharp images. In a GAN model, the generator maps a random noise vector $\boldsymbol{z}$ to an image $\boldsymbol{x}'$ and tries to trick the discriminator into believing that $\boldsymbol{x}'$ is a real image coming from the training dataset. To synthesize RSP images, a GAN-based generator needs to be capable of extracting and manipulating the semantics of any region in an image (see the challenges discussed in Section 1).

One naive way to do so is to extend the VAE-GAN [16], by replacing the generator with a Ladder-VAE. This allows us to control the multi-layer decoding distributions using the sampling techniques discussed in Section 2. So, the extended VAE-GAN model is able to output crisp RSP images without resorting to the autoregressive alternatives, which are slow at generating images. However, the model suffers from a practical drawback that it requires a dedicated training phase *for each* given reference image and R. The training cost prevents the model from being useful in interactive scenarios, such as the interactive photo editing [1].

Here, we propose a novel model, called the *fast region-semantics preserver* (Fast-RSPer), that uses a pre-trained GAN generator G and a pre-trained deep feature extractor $\Phi = \{\Phi^{(1)}, \Phi^{(2)}, \cdots, \Phi^{(L)}\}$ (e.g., a deep CNN), where $\Phi^{(L)}$ is the deepest layer, to synthesize RSP images. We feed the output of the generator to the feature extractor, as shown in Figure 3(d). Given a reference image $\boldsymbol{x}$ and a region specification R, the Fast-RSPer synthesizes an image by solving the input $\boldsymbol{z}^*$ for G first, where

$$\boldsymbol{z}^* = \arg\min_{\boldsymbol{z}} \Sigma_{i=l}^{L} \|\Phi^{(i)}(G(\boldsymbol{z})) - \Phi^{(i)}(\boldsymbol{x})\|_{mask}^{(i)}, \tag{1}$$

l is a sufficiently deep layer in Φ, and $\|\cdot\|_{mask}^{(i)}$ is a masked one-norm taking into account only the dimensions/features/activations of neurons whose receptive fields overlaps with R at layer i. The model then feeds $\boldsymbol{z}^*$ into G to synthesize an RSP image $G(\boldsymbol{z}^*)$. Intuitively, the Fast-RSPer generates an image by finding an input variable $\boldsymbol{z}^*$ for the generator G such that, at deep layers where neurons are able to capture the semantics of the reference region, the feature extractor Φ maps the synthesized region to features similar to those of the reference region. Note that the marginal semantics of the reference region is preserved in $G(\boldsymbol{z}^*)$ by explicitly solving Eq. (1). On the other hand, the compatibility of other regions in $G(\boldsymbol{z}^*)$ with the semantics of the reference region is maintained *implicitly* by the GAN generator—if the generator produced incompatible regions, it wouldn't have fooled the discriminator during the GAN training process.

Merits. The Fast-RSPer is an unsupervised model and does not requires human labels to train. As compared with native GAN extensions discussed above, the Fast-RSPer has an advantage that there is no dedicated training phase since both the generator G and feature extractor Φ are pre-trained. As compared with iGAN, our model preserves the semantics of R and is *end-to-end*—no auxiliary algorithm is needed to output the final results. Furthermore, the Fast-RSPer can generate an RSP image more efficiently than the autoregressive Pixel-VAE because Eq. (1) can be solved efficiently using the gradient descent and G generates all pixels at once.

3.1 Technical Contributions

Next, we discuss some engineering "tricks" we used to deliver good results. We also discuss some pitfalls to avoid when using a pre-trained generator.

Semantics alignment. We align the semantics of synthesized images with R at both the macro- and micro-scales. *Macro-alignment*: Given a reference image and R, one needs to decide the value of l in Eq. (1) and make sure that the features extracted by $\|\cdot\|_{mask}^{(i)}$, $i \geq l$, capture the marginal semantics of the reference region. There are several ways to choose an appropriate l. For example, we can use the visualization techniques [32] to understand what each neuron "sees" in Φ and then pick the layer l that can capture sufficiently complex concepts. We can also choose different l's for different R's based on the visualization techniques. However, these approaches require human intervention and may not be acceptable to the interactive applications. Instead, we use a simple strategy that selects l based on the *size of receptive fields*—the layer l is the shallowest layer in Φ where the receptive field of a neuron is larger than the minimal bounding box of R. The size of receptive fields can be calculated automatically, and it turns out that this strategy works well in our experiments. *Micro-alignment*: When finding $\boldsymbol{z}$, one can follow the style transfer [15,5] to align the convolution output (before it is applied to an activation function) of $G(\boldsymbol{z})$ and R at a deep layer. However, we find that aligning the *RELU output* of $G(\boldsymbol{z})$ and R instead can significantly improve the quality of the synthesized images. This observation may be valuable to future studies on semantic preserving.

Diversity. The vanilla Fast-RSPer has a problem that different synthesized images based on the same R may lack diversity, because the gradient descent algorithm may find the same $\boldsymbol{z}^*$ (or multiple $\boldsymbol{z}^*$'s close to each other) for different synthesized images. One way to solve this problem is to transform Eq. (1) to an energy-based model [21] and sample iteratively from such a model using a sampling algorithm (e.g., an approximate Metropolis-adjusted Langevin sampling). Here, we propose a simple alternative by taking advantage of the fact that *only the semantics* of the reference region needs to be preserved. During the gradient descent, the Fast-RSPer adds random noises to the gradients at different iterations. This leads to diverse $\boldsymbol{z}^*$'s and in turn diverse synthesized images $G(\boldsymbol{z}^*)$'s having different features at layers shallower than l. Since the layer l in Eq. (1) is a deep layer, the synthesized images $G(\boldsymbol{z}^*)$'s can have different features at shallow layers, which encode textures or colors or shapes, and can look very different from each other.

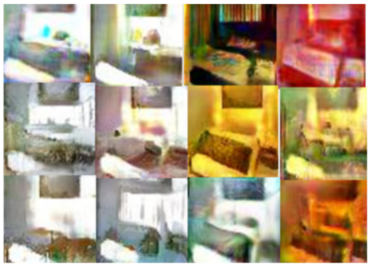


Fig. 4. Images synthesized by batch-normalized generator may have degraded quality due to the bias of moments in the current input batch.

Generator Selection. We have tried pairing up Fast-RAPer with many types of pre-trained generators but found that some generators failed to deliver good results. After investigation, we find that the root cause is due to the batch-normalization. Most existing GANs employ the batch-normalization *without* learning moving moments. They only normalize the latent outputs based on the moments of individual training batches rather than the moments of entire training data. If this is the case, we suggest that one should turn *off* the batch-normalization in G, because it introduces dependency between batch inputs during training and seriously degrades the quality of images generated by

G, as shown in Figure 4. To ensure the best quality, we use pre-trained generators *without* batch normalization layers in our experiments.

4 Experiments

In this section, we evaluate the performance of Fast-RSPer by comparing it with existing models. We use the CelebA[1] and LSUN Bedroom[2] datasets for training the models and choosing the reference images/regions for problems of RSP image synthesis. We implement the models using Tensorflow.[3] Following summarizes the models we implemented:

Pixel-VAE. To the best of our knowledge, the only off-the-shelf models that can be used for RSP image synthesis is the Ladder-VAE [29] and its variant Pixel-VAE [22]. We implement both the models but find that the Ladder-VAE consistently yields blurry images that are *not* comparable to other approaches in terms of image quality. Therefore, we omit its results in order to save space. The Pixel-VAE model consists of a Ladder-VAE and a Pixel-CNN [23]. When generating an image, the Ladder-VAE part ensures the global coherence of the image while the Pixel-CNN part enhances the details at the pixel level. We implement Pixel-VAE by following the architecture described in the original paper [22]. Note that in this architecture, the encoder and decoder of the Ladder-VAE part have two layers parametrized by the latent variables $\boldsymbol{z}^{(1)}$ and $\boldsymbol{z}^{(2)}$ respectively (see Figure 3(a)), and its is sufficient for $\boldsymbol{z}^{(2)}$ to control the semantics of *any* region in an image since the second-layer encoder and decoder are fully-connected layers. The number of weights to learn is about 67M.

Extended VAE-GAN. We also implement the naive extension of VAE-GAN [16] discussed in Section 3. We follow the architecture used in [16] and replace the VAE part with the Ladder-VAE described above. Unfortunately, despite applying many GAN-training techniques [27] and careful engineering, we are not able to train the VAE-GAN extension successfully. It turns out this is due to the limitation of VAE-GAN—it can be trained to generate face images but not bedrooms . We believe this is because that a face have relatively simpler parametrization than a bedroom. And the distribution assumption of the VAE-GAN (in the VAE part) limits the model complexity and stops the model from learning a complex parametrization. We therefore omit the results of this model.

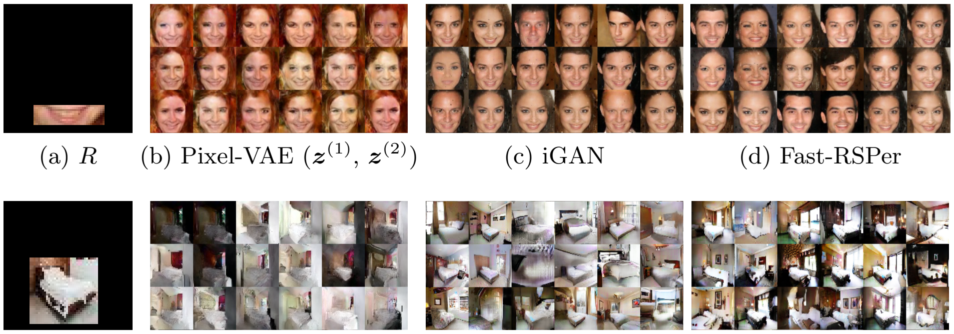
Fast-RSPer. We use a pre-trained VGG-19 [28] as the feature extractor. Note that in VGG-19, the convolution, pooling, and activation layers are grouped into *blocks*. We set l at the *start* of a block instead of in the middle because this leads to better results empirically. For the CelebA and LSUN Bedroom datasets, we pre-train two generators following the BEGAN [3] and WGAN-GP [10], respectively. As discussed in Section 3, we do *not* employ normalization layers in both generators. The numbers of model weights to learn are about 23M for CelebA dataset and 29M for Bedroom dataset, respectively.

[1] http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

[2] http://lsun.cs.princeton.edu/

[3] https://www.tensorflow.org/

iGAN. Although the iGAN[34] was not proposed to solve the RSP image synthesis problem, we implement it as a baseline to study the difference between pixel- and semantics-preserving. Like Fast-RSPer, iGAN needs a pre-trained generator. So, for fairness, we let it use the same generators as in Fast-RSPer (i.e., BEGAN on CelebA dataset and WGAN-GP on LSUN Bedroom dataset). Following the settings reported in the iGAN paper [34], we use L2 pixel loss for alignment and update the $\boldsymbol{z}$ variable without gradient noise. And we use the same optimizer and maximum step as in Fast-RSPer during training. The numbers of model weights for different datasets are also roughly the same as Fast-RSPer.



(a) R (b) Pixel-VAE ($\boldsymbol{z}^{(1)}$, $\boldsymbol{z}^{(2)}$) (c) iGAN (d) Fast-RSPer

(e) R (f) Pixel-VAE ($\boldsymbol{z}^{(1)}$, $\boldsymbol{z}^{(2)}$) (g) iGAN (h) Fast-RSPer

Fig. 5. Synthesized images given simple reference regions containing only a single object. (a) A laughing mouth as a reference region. (b-d) Images synthesized by Pixel-VAE, iGAN, and Fast-RSPer trained on the CelebA dataset. (e) A bed as a reference region. (f-h) Images synthesized by Pixel-VAE, iGAN, and Fast-RSPer trained on the LSUN Bedroom dataset.

4.1 Results Given Single-Object Regions

We first study the images synthesized by different models given a simple reference region consisting of one object. We choose a "laughing mouth" as the reference region for models trained on CelebA and a "bed" as the reference region for models trained on LSUN Bedroom, as sown in Figures 5(a)(e). In Pixel-VAE, we fix the decoding distributions parametrized by both $\boldsymbol{z}^{(1)}$ and $\boldsymbol{z}^{(2)}$ along the dimensions whose receptive field covers R in order to make it preserve the semantics of the reference region. The synthesized images are shown in Figures 5(b)(c)(e)(f).

As we can see, the iGAN and Fast-RSPer give relatively better results than Pixel-VAE. The iGAN performs well because the pixels in the reference region are simple thus aligning pixels amounts to aligning the semantics. On the other hand, the images synthesized by Pixel-VAE are less satisfactory. First, they contains local artifacts. This is because the images are synthesized by the Pixel-CNN part, which is an autoregressive model. A "bad" pixel sampled by Pixel-CNN creates a negative impact on the following pixels and finally leads to a local artifact. The iGAN and Fast-RSPer avoid this problem by generating all pixels of an image at once.[4] Moreover, the images produced by Pixel-VAE have less diversity—most images share the same global tone. Since $\boldsymbol{z}^{(2)}$ is the output of fully connected layer, its receptive field covers the entire face region. So when $\boldsymbol{z}^{(2)}$ was fixed, the output would have low diversity, even $\boldsymbol{z}^{(1)}$ can be sampled. Fast-RSPer avoids this problem by using a GAN without feature parametrization. We can also see that, on LSUN Bedroom, Pixel-VAE is less capable of preserving the semantics of the reference region than Fast-RSPer (Figures 5(e)(f)). A bedroom scene is usually more complex than a face, implying a wider feature space to learn at each layer. Images in the LSUN Bedroom dataset usually look very different from each other, and it is hard to learn a shared parametrization of such complex feature spaces for the encoder and decoder.

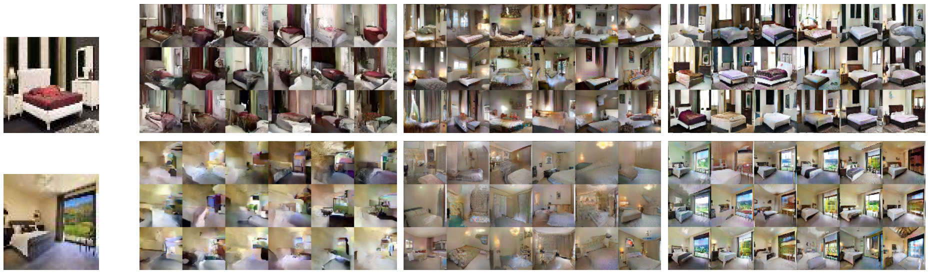
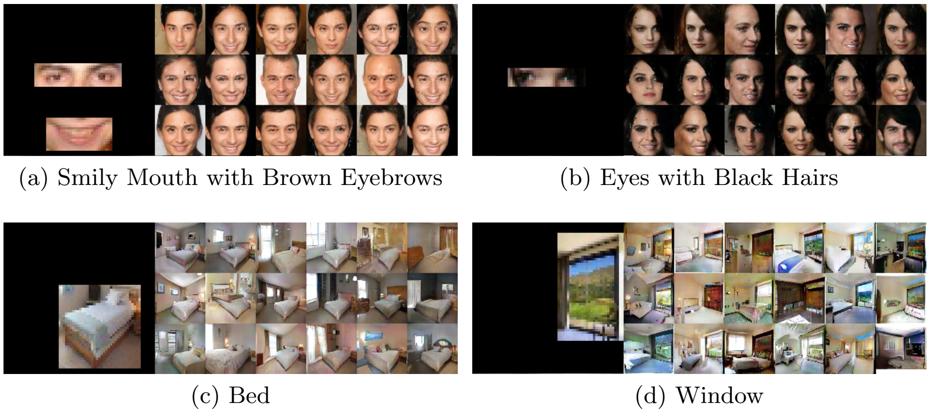

(a) R (b) Pixel-VAE ($\boldsymbol{z}^{(2)}$) (c) iGAN (d) Fast-RSPer

Fig. 6. Synthesized images given a complex reference region containing many objects. (a) A bedroom scene as a reference region. (b-d) Images synthesized by PixelVAE, iGAN, and Fast-RSPer respectively.

4.2 Results Given Complex Regions

Here, we investigate the images synthesized by different models given a complex reference region representing the entire bedroom scene, as shown in Figure 6(a). In a complex region, subregions may have their own semantics. For example, a bedroom image may contain beds, desks, lamps, or pictures on the wall. The goal of RSP image synthesis is to preserve the *marginal* semantics of the reference region; that is, the semantics excluding that of any subregion. In this case,

[4] One may notice that the images synthesized by iGAN and Fast-RSPer contains some small "holes" on the CelebA dataset. This is due to the pre-trained generator BEGAN, not the synthesis models themselves, as evidenced by Figure 8(a).



(a) Smily Mouth with Brown Eyebrows

(b) Eyes with Black Hairs

(c) Bed

(d) Window

Fig. 7. Images synthesized by Fast-RSPer given different reference regions. In each image block, the left (large) image shows R in which the marginal semantics needs to be preserved.

the marginal semantics should be the layout of a bedroom. The synthesized images are shown in Figures 6(b)(c)(d). As we can see, only Fast-RSPer generates satisfactory images this time.

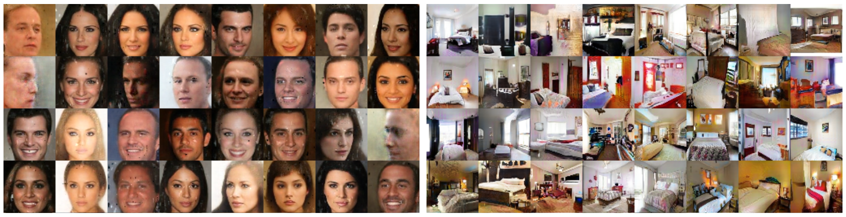
The iGAN, which is based on pixel-alignment, cannot well preserve the semantics of R because the pixels in R are diverse. Aligning pixels of synthesized images with those in R results in a group of images whose average resembles R but lack semantics individually.

In Pixel-VAE, we fix only the decoding distribution parametrized by $\boldsymbol{z}^{(2)}$ (not the one by $\boldsymbol{z}^{(1)}$) along the dimensions whose receptive field covers R in order to make it preserve the marginal semantics of the reference region. As we can see from Figure 6(b), the layout is less preserved by the images generated by Pixel-VAE than by Fast-RSPer. This is because the shared parametrization is harder to train on a complex dataset, as discussed in the previous subsection.

4.3 Semantics Preserving vs. Mode Collapse

Next, we look into the capability of semantics preserving of Fast-RSPer. We can see from Figures 7(a-d) that Fast-RSPer can successfully synthesize images given various reference regions with different facial/bedroom semantics.

Note that, in some cases, the images synthesized by Fast-RSPer render less variety as compared to those that could have been generated by a well-trained, unconstrained GAN generator. One may conjecture that a poor-trained GAN generator with significant mode collapse is used. However, this is not the case. Figure 8 shows the images synthesized by our pre-trained generator, which include a variety of modes. The reason behind the reduced variety is because of the constraints imposed by the *semantics* of the reference region. When comparing Fast-RSPer with Pixel-VAE (which is also conditioned) in Figures 5 and 6, our method clearly gives more diversity.



(a) CelebA (b) LSUN Bedroom

Fig. 8. Images synthesized by our pre-trained generators do not have the problem of mode collapse, showing that the reduced variety among synthesized images is due to the semantics constraints. Note that the CelebA images have small "holes" because BEGAN [3] clips color ranges of pixels. The WGAN-GP [10], which generates bedroom images, does not have the same problem.

4.4 Quantitative Comparison

Table 1. Quantitative comparison results. Average image synthesis time is reported on a machine with an Intel Core-i7 6900K CPU, 128GB RAM, and an Nvidia GeForce GTX 1070 GPU.

	CelebA			**LSUN Bedroom**			
	Semantics	**Quality**	**Diversity**	**Semantics**	**Quality**	**Diversity**	**Avg. Time**
Pixel-VAE	328 wins	416 wins	310 wins	201 wins	525 wins	291 wins	43 secs
iGAN	451 wins	347 wins	482 wins	508 wins	227 wins	408 wins	**0.3** sec
Fast-RSPer	**560** wins	**576** wins	**547** wins	**630** wins	**587** wins	**640** wins	0.4 sec

To quantitatively compare the level of semantics preserving, quality and diversity of images synthesized by Pixel-VAE, iGAN and Fast-RSPer, we conduct human evaluations. We invite 103 external users from the Internet. Each user is asked to decide the winners in terms of quality, diversity and level of semantics preserving for 13 experiments. In each experiment, we display 36 images synthesized by Pixel-VAE, iGAN and Fast-RSPer respectively given the same reference region. The number of wins of each model is shown in Table 1. It is clear that people think Fast-RSPer 1) gives better-quality images on both the CelebA and LSUN Bedroom datasets; and 2) preserves more semantics on the LSUN Bedroom dataset while 3) giving more diversity. Table 1 also includes the average time required to generate an image. The time applies to both the datasets since they contain images of the same size (64 × 64 pixels). Both the iGAN and Fast-RSPer can synthesize images in near realtime. The average generating time of iGAN is 0.1 second faster since it doesn't need to feed forward the VGG. In summary, Fast-RSPer is able to efficiently generate RSP images of higher quality and diversity. This is important when one wants to synthesize a large amount of images for, e.g., data augmentation.

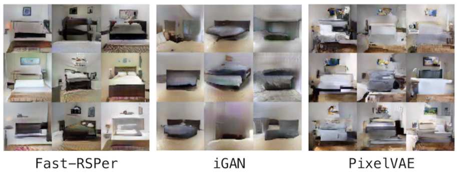
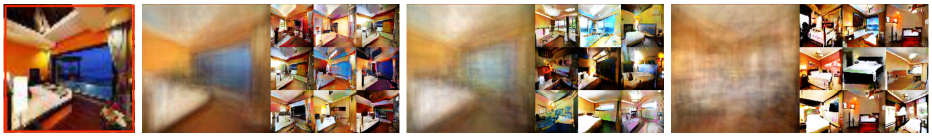


Fig. 9. Synthesized Images

We have also used two standard metrics, the Inception Score (IS) and structural similarity (SSIM), to evaluate the quality and diversity of synthesized images. The higher the IS (and the lower the SSIM) the better. We conduct experiments where there are 25 reference images and we generate 64 images for each reference image using PixelVAE, iGAN, and our Fast-RSPer. We average the IS and SSIM scores of the synthesized images and give the results in Table 2. We also show some randomly sampled synthesized images in Figure 9 As we can see, neither IS nor SSIM can truly indicate the quality and diversity in our problem. Although giving the highest quality through human eyes, Fast-RSPer does not top the Inception Scores because it can generate some objects that are out of the classes which the Inception model was trained for. Also, the low SSIM values given by PixelVAE are not from diversity but from distortion and artifacts. This is why we employ human evaluation instead.

	Inception		SSIM	
	CelebA	Bedroom	CelebA	Bedroom
PixelVAE	1.39	2.17	0.92	**0.20**
iGAN	**1.56**	**2.61**	1.01	0.66
Ours	1.51	2.23	**0.89**	0.28

Table 2. Inception Scores and SSIM.



(a) R (b) Mid l and Gradient Noises (c) High Gradient Noises (d) High l

Fig. 10. Images synthesized by Fast-RSPer given different l's and levels of gradient noises. In each image block, the large image at left is the average of synthesized images.

4.5 Effect of l and Gradient Noises

We study the impact of l in Eq. (1) and the noises added to the gradients at each iteration when solving $\boldsymbol{z}^*$ using the gradient descent. The layer l controls the minimal receptive fields of neurons in the feature extractor used to capture the semantics of the reference regions. The larger the l, the higher-level concepts are captured and preserved by Fast-RSPer. On the other hand, the level of gradient noises controls how diverse the features in the shallow layers of the feature extractor when generating an image. The larger the gradient noises, the more visual diversity we can expect *given the same preserved semantics*. To see how l and the level of gradient noises affect a synthesized image, we show images synthesized by Fast-RSPer given enlarged l and noise level in Figure 10. In Fig 10, we use Fig 10(b) as the baseline and change *one hyperparameter at a time* in Fig 10(c) and Fig 10(d). Fig 10(c) and Fig 10(d) show the impact of increasing gradient noises and l, respectively. As we can see, with a high noise level (Figure 10(c)), the diversity of synthesized images increases, but in average the image does not change too much. The diversity is added over the same semantics. A larger l (Figure 10(d)) also leads to more diversity. But such diversity comes from less preserved semantics. In this case, the only preserved semantics is the layout of the roof, excluding the layout of furnitures in the room which was preserved in Figure 10(b). In practice, these two parameters offer a flexible way for users to fine-control the diversity and what semantics to preserve.

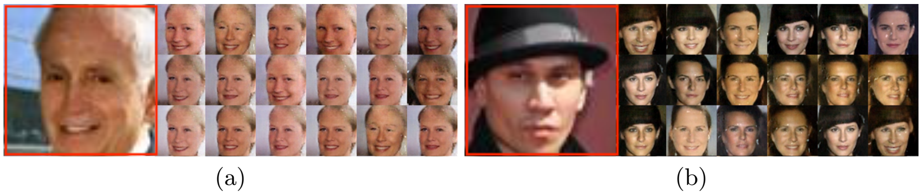


Fig. 11. Cross-domain RSP image synthesis, where the source domain is "male," the target domain is "female," and the region semantics is "face." Given a source domain A, a target domain B, and the reference region R in the source domain, the goal is to synthesize an image $\boldsymbol{y}'$ *in the target domain* that keeps the high-level semantics of R. In Fast-RSPer, we replace the single-domain feature extractor with a cross-domain feature extractor and use an ordinary GAN generator for domain B. This way, an image generated by the generator ($\boldsymbol{x}'$ in Figure 3(c)) can lie in domain B while preserving the high-level semantics of $\boldsymbol{x}$ in domain A. Note that the cross-domain feature extractor can be trained in an unsupervised manner using an autoencoder that takes both example images from domains A and B as the input.

5 Conclusion

We study the problem of region-semantics preserving (RSP) image synthesis that allows a user to easily guide the synthesis using a reference image in a soft manner. We then propose the Fast-RSPer model based on a pre-trained GAN generator and a pre-trained deep feature extractor. The Fast-RSPer is able to generate images without undergoing a dedicated training phase, making it particularly useful for the interactive applications. We conduct extensive experiments using the real-world datasets and the results show that Fast-PSPer can synthesize realistic, diverse RSP images efficiently.

References

1. Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D., Cohen, M.: Interactive digital photomontage. In: ACM Transactions on Graphics (TOG). vol. 23, pp. 294–302. ACM (2004) 1, 3
2. Ballé, J., Laparra, V., Simoncelli, E.P.: Density modeling of images using a generalized normalization transformation. arXiv preprint arXiv:1511.06281 (2015) 1
3. Berthelot, D., Schumm, T., Metz, L.: Began: Boundary equilibrium generative adversarial networks. arXiv preprint arXiv:1703.10717 (2017) 4, 8
4. Dosovitskiy, A., Tobias Springenberg, J., Brox, T.: Learning to generate chairs with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1538–1546 (2015) 2
5. Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576 (2015) 2, 3.1
6. Gatys, L.A., Ecker, A.S., Bethge, M.: Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks. arXiv preprint arXiv:1505.07376 **12** (2015) 1
7. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014) 1, 1, 3
8. Gregor, K., Danihelka, I., Graves, A., Rezende, D., Wierstra, D.: Draw: A recurrent neural network for image generation. In: International Conference on Machine Learning. pp. 1462–1471 (2015) 1
9. Güçlütürk, Y., Güçlü, U., van Lier, R., van Gerven, M.A.: Convolutional sketch inversion. In: European Conference on Computer Vision. pp. 810–824. Springer (2016) 2
10. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of wasserstein gans. arXiv preprint arXiv:1704.00028 (2017) 4, 8
11. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H.: Image analogies. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. pp. 327–340. ACM (2001) 2
12. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. ACM Transactions on Graphics (TOG) **36**(4), 107 (2017) 1
13. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. arXiv preprint arXiv:1611.07004 (2016) 1
14. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013) 1, 2
15. Kyprianidis, J.E., Collomosse, J., Wang, T., Isenberg, T.: State of the "art": A taxonomy of artistic stylization techniques for images and video. IEEE transactions on visualization and computer graphics **19**(5), 866–885 (2013) 2, 3.1
16. Larsen, A.B.L., Sønderby, S.K., Larochelle, H., Winther, O.: Autoencoding beyond pixels using a learned similarity metric. In: International Conference on Machine Learning. pp. 1558–1566 (2016) 3, 4
17. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. arXiv preprint arXiv:1609.04802 (2016) 1
18. Lee, H., Grosse, R., Ranganath, R., Ng, A.Y.: Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the 26th annual international conference on machine learning. pp. 609–616. ACM (2009) 1

19. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014) 2
20. Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T., Clune, J.: Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In: Advances in Neural Information Processing Systems. pp. 3387–3395 (2016) 1
21. Nguyen, A., Yosinski, J., Bengio, Y., Dosovitskiy, A., Clune, J.: Plug & play generative networks: Conditional iterative generation of images in latent space. arXiv preprint arXiv:1612.00005 (2016) 3.1
22. van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al.: Conditional image generation with pixelcnn decoders. In: Advances in Neural Information Processing Systems. pp. 4790–4798 (2016) 1, 2, 4
23. Oord, A.v.d., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. In: International Conference on Machine Learning. pp. 1747–1756 (2016) 1, 2, 4
24. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015) 1
25. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. In: International Conference on Machine Learning. pp. 1060–1069 (2016) 2
26. Salakhutdinov, R., Hinton, G.: Deep boltzmann machines. In: Artificial Intelligence and Statistics. pp. 448–455 (2009) 1
27. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in Neural Information Processing Systems. pp. 2234–2242 (2016) 4
28. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) 4
29. Sønderby, C.K., Raiko, T., Maaløe, L., Sønderby, S.K., Winther, O.: Ladder variational autoencoders. In: Advances in Neural Information Processing Systems. pp. 3738–3746 (2016) 1, 2, 4
30. Toderici, G., Vincent, D., Johnston, N., Hwang, S.J., Minnen, D., Shor, J., Covell, M.: Full resolution image compression with recurrent neural networks. arXiv preprint arXiv:1608.05148 (2016) 1
31. Yan, X., Yang, J., Sohn, K., Lee, H.: Attribute2image: Conditional image generation from visual attributes. In: European Conference on Computer Vision. pp. 776–791. Springer (2016) 2
32. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European conference on computer vision. pp. 818–833. Springer (2014) 3.1
33. Zhou, T., Tulsiani, S., Sun, W., Malik, J., Efros, A.A.: View synthesis by appearance flow. In: European Conference on Computer Vision. pp. 286–301. Springer (2016) 2
34. Zhu, J.Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. In: European Conference on Computer Vision. pp. 597–613. Springer (2016) 2, 2, 4
35. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv preprint arXiv:1703.10593 (2017) 1