DeepSEE: Deep Disentangled Semantic Explorative Extreme Super-Resolution *Supplementary Material*

Marcel C. Bühler[0000−0001−8104−9313], Andrés Romero[0000−0002−7118−5175], and Radu Timofte[0000−0002−1478−0402]

Computer Vision Lab, ETH Zürich, Switzerland
{buehlmar,roandres,timofter}@ethz.ch

1 Architecture

In this section, we describe the *DeepSEE* architecture in more detail. For a clearer understanding, we provide tabular illustrations for the encoder and generator networks for 8× magnification (Tables 1, 2, 3 and 4) and we zoom into the residual blocks where we inject semantic and style information (Fig. 1).

1.1 Style Encoder

The style encoder extracts a style matrix from a low- or high-resolution image. Each row in the style matrix represents a disentangled representation of the appearance of one semantic region. One important step to achieve such a disentanglement is *regional average pooling*. *Regional average pooling* leverages segmentation masks to extract spatial-dependent features.

In our implementation, it works as follows: The style encoder computes features maps from an image. We average those features along the spatial dimensions in order to obtain a style matrix $S \in [-1,1]^{N \times d}$, where N is the number of semantic regions and d is the size of the style vector for one region.

Concretely, let $F \in [-1,1]^{H_{\text{fm}} \times W_{\text{fm}} \times d}$ be the encoder features extracted from an image, where H_{fm} and W_{fm} are the output height and width of the shared encoder E_{shared}. For each semantic region, we average the features of that region along the spatial dimensions. First, we scale the predicted binary segmentation mask to the size of the feature map: $M^{\text{resized}} \in \{0,1\}^{H_{\text{fm}} \times W_{\text{fm}} \times N}$. Second, we define the set of all spatial locations belonging to region R_n, $n = 1, ..., N$ and then compute the entries $k = 1, ..., d$ in the style matrix S from the features F:

$$R_n = \{(x,y) | M^{\text{resized}}_{x,y,n} = 1 \text{ and } x = 1, .., H_{fm} \text{ and } y = 1, ..., W_{fm}\} \quad (1)$$

$$S_{n,k} = \frac{1}{H_{fm} W_{fm}} \Sigma_{(x,y) \in R_n} F_{x,y,k} \quad (2)$$

Table 1. LR Style Encoder Architecture. The low-resolution style encoder consists of a series of convolutional layers with kernel size 3×3 and instance norm layers (IN) [1]. We upsample the feature maps for a better style disentanglement (as described in Section 1.1). The shared style encoder (Table 3) processes the output further and maps it to a shared latent space.

Block / Layer	Stride	#Params	Output Shape
Conv+IN	1	864	$16 \times 16 \times 32$
Conv+IN	1	18,432	$16 \times 16 \times 64$
Conv+IN	1	73,728	$16 \times 16 \times 128$
Upsampling	-	0	$32 \times 32 \times 128$
Conv+IN	1	294,912	$32 \times 32 \times 256$
		387,936	

Table 2. HR Style Encoder Architecture. For the high-resolution encoder, we downsample twice and then re-upsample again to create a bottleneck. All convolutions have kernel size 3×3 and they are followed by instance normalization [1]. Again, the shared encoder (Table 3) processes the output further.

Block / Layer	Stride	#Params	Output Shape
Conv+IN	1	864	$128 \times 128 \times 32$
Conv+IN	2	18,432	$64 \times 64 \times 64$
Conv+IN	2	73,728	$32 \times 32 \times 128$
Upsampling	-	0	$64 \times 64 \times 128$
Conv+IN	1	294,912	$64 \times 64 \times 256$
		387,936	

1.2 Generator

In this section, we describe the main building block of the generator, the *ResBlock*, and explain how *DeepSEE* injects semantics and style.

Residual Blocks. Starting from a low-resolution image, the generator repeatedly doubles the resolution via nearest-neighbour upscaling and processes the result in residual blocks (ResBlocks). As a guidance, the ResBlocks inject semantic and style information via spatially adaptive normalization (SPADE) [2] and *semantic region adaptive normalization* [3].

Fig. 1 shows a ResBlock and zooms into the normalization block (NormBlock). The ResBlock follows the design by [3] and processes the input feature maps with two normalization blocks NormBlocks in a main path and one NormBlock in a residual connection. It also adds noise from a standard normal distribution, which improved the FID score in [3].

The NormBlock (Fig. 1) learns two sets of modulation parameters. The first set is responsible for semantic consistency. The second set applies the disen-

Table 3. Shared Style Encoder Architecture. The shared style encoder processes the output of both the low-resolution and the high-resolution encoders and maps their outputs to a latent space. It uses a predicted semantic mask to extract styles for each semantic region in the regional average pooling layer. For a detailed description of regional average pooling, please refer to Section 1.1.

Block / Layer	Stride	#Params	Output Shape LR	Output Shape HR
Conv+IN	1	294,912	$32 \times 32 \times 128$	$64 \times 64 \times 128$
Tanh	-	0	$32 \times 32 \times 128$	$64 \times 64 \times 128$
Regional avg pool	-	0	19×128	19×128
		294,912		

Table 4. Generator Architecture for $8\times$ Magnification. We list the main layers and building blocks with the number of parameters and output shapes. All residual blocks inject semantics [2] and style [3], except for the first residual block, which only injects semantics. Please refer to Fig. 1 for a detailed view of the ResBlock.

Block / Layer	Semantics	Style	#Params	Output Shape
Conv	-	-	14,336	$16 \times 16 \times 512$
ResBlock	✓	-	7,126,528	$16 \times 16 \times 512$
Upsampling	-	-	0	$32 \times 32 \times 512$
ResBlock	✓	✓	9,488,636	$32 \times 32 \times 512$
ResBlock	✓	✓	9,488,636	$32 \times 32 \times 512$
Upsampling	-	-	0	$64 \times 64 \times 512$
ResBlock	✓	✓	9,488,636	$64 \times 64 \times 512$
Upsampling	-	-	0	$128 \times 128 \times 512$
ResBlock	✓	✓	9,488,636	$128 \times 128 \times 512$
Conv	-	-	13,827	$128 \times 128 \times 3$
Tanh	-	-	0	$128 \times 128 \times 3$
			45,109,235	

tangled style information from a style matrix. As a reminder, the style matrix contains a style vector for each semantic region. For computing the modulation parameters, the style matrix expands and colors the semantic mask, yielding a *styled map*. Similar to previous work [2, 3], we compute a scale and offset for each feature pixel. As a last step, we combine the two sets of modulation parameters via a weighted average and apply them to the input feature map. In our ablation study (in the main paper), we compare the model performance for variants where we remove semantics, style or both from the NormBlock.

Upsampling and Number of Layers. The generator consists of residual blocks, which refine the output of deterministic upscaling layers. The number of upscaling layers depends on the magnification factor. Each upscaling layer increase the image dimensions by the factor 2. For upscaling $4\times$, our model

consists of 2 upscaling layers. A model with upscaling factor 2^i applies i upscaling layers. For example, our extreme super-resolution model with upscaling factor $32 = 2^5$ consists of 5 upscaling layers.

1.3 Discriminator

Our Patch-GAN discriminator [4] follows the one in [2]. It consists of two networks, one that operates on the high-resolution image and one on the half scale. The networks take the concatenation of an image together with its segmentation mask as input. Both networks predict the realism of overlapping image patches and the loss function computes the average Hinge loss of their outputs.

Architecturally, four convolutional and spectral instance normalization [5] layers process the input, with leaky ReLU [6–8] as the activation function. In the first three layers, we use a stride of two in order to reduce the feature map size. The last convolution applies a stride of one. We use the same number of discriminator layers for all our experiments.

1.4 Semantic Segmentation Network

We train a state-of-the-art DeepLab semantic segmentation network [9, 10] to predict a high-resolution segmentation map $M \in \{0,1\}^{H_{hr} \times W_{hr} \times N}$ from a low-resolution image $x_{lr} \in \mathbb{R}^{H_{lr} \times W_{lr} \times 3}$. Our segmentation model first upscales the low-resolution image to the high-resolution and then processes it in the same way as a regular high-resolution image.

We train the segmentation model on the CelebAMask-HQ [11, 12] dataset for 150 epochs, with a polynomially decaying learning rate of 0.001 and stochastic gradient descent with momentum 0.9 and weight decay 0.0005.

Table 5 lists performance metrics with different backbones. All of the models are trained with balanced class weights. For our experiments, we choose the best scoring model with DRN backbone [13, 14]. Fig. 2 shows a random sample of qualitative results.

2 Training and Testing

In this section, we list formulas for the loss function, training details, hyper-parameters and describe the test dataset splits. Unless otherwise stated, we use the same hyper-parameters for all our experiments.

2.1 Loss Function

Our loss function follows [2]. We use a discriminator network D to compute adversarial losses for the generator and the style encoder $\mathcal{L}_{adv}$ and for the discriminator $\mathcal{L}_{adv_D}$. For the generator and the style encoder, we extend the loss with a discriminator feature matching $\mathcal{L}_{feat}$ and a perceptual loss $\mathcal{L}_{vgg}$ from

a VGG-19 network [15, 16]. In the following, we define our loss terms for real high-resolution images $x \sim p_{data}$ and the corresponding model output $\hat{x} \sim G_{\Theta}$.

We use a discriminator D to compute an adversarial hinge loss, which is defined as:

$$\mathcal{L}_{adv} = -\mathop{\mathbb{E}}_{\hat{x}}[D(\hat{x})] \tag{3}$$

$$\mathcal{L}_{adv_D} = -\mathop{\mathbb{E}}_{x}[\min(0, D(x) - 1] - \mathop{\mathbb{E}}_{\hat{x}}[\min(0, -D(\hat{x}) - 1] \tag{4}$$

The feature matching loss $\mathcal{L}_{feat}$ is computed from the L1 distance between the discriminator features for the real and the fake image. Let $F_D^{(i)}(\cdot)$ be the discriminator features for the layer i. Then, we compute the feature matching loss for the intermediate discriminator layers:

$$\mathcal{L}_{feat} = \mathop{\mathbb{E}}_{(x,\hat{x})}\left[\sum_{i=2}^{4} \|F_{D^{(i)}}(x) - F_{D^{(i)}}(\hat{x})\|_1\right] \tag{5}$$

We compute our perceptual loss $\mathcal{L}_{vgg}$ from the features of a VGG-19 network [15, 16]. Let $F_{vgg^{(i)}}(\cdot)$ be the features after ReLU activation for blocks $i = 1, \ldots, 5$ in the VGG-19 network and let $\mathbf{w} = [\frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, 1]^\top$ a vector with fixed weights. Then, we compute the perceptual loss as:

$$\mathcal{L}_{vgg} = \mathop{\mathbb{E}}_{(x,\hat{x})}\left[\sum_{i=1}^{5} \mathbf{w}_i \|F_{vgg^{(i)}}(x) - F_{vgg^{(i)}}(\hat{x})\|_1\right] \tag{6}$$

Our full loss function for the generator $\mathcal{L}_G$ and style encoder $\mathcal{L}_E$, and for the discriminator $\mathcal{L}_D$ are defined in Equations 7 and 8:

$$\mathcal{L}_G = \mathcal{L}_E = \mathcal{L}_{adv} + \lambda_{feat}\mathcal{L}_{feat} + \lambda_{vgg}\mathcal{L}_{vgg} \tag{7}$$

$$\mathcal{L}_D = \mathcal{L}_{adv_D} \tag{8}$$

We set the loss weights to $\lambda_{feat} = \lambda_{vgg} = 10$ for all our experiments.

Please note that our model uses two discriminators, one operates on the full scale of the output image and one on the half scale. The final loss is computed as their average.

2.2 Training Details

Optimizer. In all our super-resolution experiments, we use the Adam [17] optimizer. We set the parameters for the exponentially moving average as beta1 = 0.0 and beta2 = 0.9 for the squared gradient.

Learning Rate. We chose the learning rate 0.0001 for the generator and 0.0004 for the discriminator. The high-resolution and the shared style encoders have the same learning rate as the generator. The low-resolution style encoder is trained with a smaller learning rate of 0.000025.

The reason for the smaller learning rate in the low-resolution style encoder is that *DeepSEE* needs to handle both low-resolution and high-resolution style inputs. The high-resolution style encoder can extract rich style information. The low-resolution style encoder, however, does not receive such high-frequency information. Therefore, it needs to predict them, which we consider a considerably harder task compared to the high-resolution encoder. Therefore, we set a lower learning rate for the low-resolution style encoder. We train both encoders alternately, by feeding a low-resolution or a high-resolution image in 50% of the iterations.

Initialization. We initialize all our networks using xavier [18] with a variance of 0.02.

Training Time. On CelebA [19], a large dataset with $162,770$ training samples, we train for 8 epochs, linearly decaying the learning rate in the last 3 epochs. On CelebAMask-HQ [11, 12], whose train split consists of $24,183$ samples, we train for 75 epochs in total, and linearly decay the learning rate in the last 25 epochs. The training time is between 3-5 days on a TITAN Xp GPU on both datasets for $8\times$ magnification. For the extreme upscaling factors ($32\times$), training takes between 7-8 days on 2 Tesla V100 GPUs.

Batch Size. We vary batch size according to the input image size and upscaling factors. For experiments starting at 16×16 low-resolution images and magnification factor $8\times$, we set the batch size to 4. For the experiments that upscale 32×32 images, we reduce the batch size to 1. We use batch size 2 for all experiments with extreme upscaling factors ($32\times$).

Noise Injection. We inject uniformly distributed noise into the style matrix to increase the model's robustness towards exploration of the style space, as described in the main paper. The scale of the noise depends on the model variant. For the *independent* model, we choose $\delta = 0.2$. The *guided* model is trained with $\delta = 0.05$.

2.3 Testing

Datasets. We compute the following test scores for all datasets on the official test splits [19, 11, 12]: PSNR, SSIM [20], LPIPS [21] and FID [22]. For CelebA, the test dataset consists of $19,962$ samples. The CelebAMask-HQ test set contains $2,824$ samples. For CelebA, the predicted segmentation masks are of size

Table 5. Results for Semantic Segmentation on Low-resolution Images. The segmentation network predicts 128×128 segmentation masks from 16×16 images. We report accuracy (Acc), mean Intersection-over-Union (mIoU) and frequency-weighted Intersection-over-Union (fwIoU).

Backbone	Acc ↑	mIoU ↑	fwIoU ↑
Xception [25]	0.821	0.457	0.714
Resnet [26]	0.848	0.457	0.747
DRN [13, 14]	**0.877**	**0.547**	**0.794**

128×128, and for CelebAMask-HQ, we predict high-resolution segmentation masks of size 512×512.

Our *guided* model, as well as some of the related work [23, 24] require an additional guiding image from the same person. For those experiments, we omit people who only appear once in the dataset.

Pre-processing. For our models, we calculate the low-resolution input via bicubic interpolation. For the related work, we follow their pre-processing and use their respective downsampling method as input to their models.

The CelebA [19] images have height 218 and width 178 pixels. We center crop the CelebA images to 178×178 pixels and resize them to 128×128 for the high-resolution ground truth.

For CelebAMask-HQ [11, 12], we resize the images to the desired high-resolution (512×512 for the extreme super-resolution and 256×256 for comparing to [23, 24]).

3 Additional Results

3.1 Exploration of the Solution Space

As an explorative super-resolution model, *DeepSEE* allows to sample in the manifold of possible solution for a given low-resolution image. This can be very useful in a broad range of scenarios. For example, some applications might be more interested in high fidelity, while others need images of high perceptual quality. To illustrate the dynamic nature of *DeepSEE*, we take 12 random images and sample 1,000 different potential solutions for each of them by adding scaled Gaussian noise to the style matrix. Fig. 5 shows some visual examples. The plots in Fig. 6 and Fig. 7 show the resulting scores for fidelity (SSIM [20], y-axis) and perceptual quality (LPIPS [21], x-axis).

In theory, one could sample an infinite number of times to increase the chances of producing the desired result. However, in most cases, a relatively small sample of 10 already contains sufficient variability. To illustrate the impact of sampling upscaled variants on the fidelity and perception metrics, we

sample 1, 10, 100 and 1,000 random solutions and mark the best-scoring sample in terms of SSIM (Fig. 7) and LPIPS (Fig. 6). We conclude that *DeepSEE* can obtain high scoring solutions for both objectives: high fidelity and perceptual quality.

3.2 Semantic Manipulations

We provide additional semantic manipulations on images with upscaling factor 32×. Fig. 8 demonstrates multiple manipulations for the same sample. In the group, we first close the eyes and reopen them again. For the last two versions, we slightly open the mouth and remove the glasses. For the second example, we play with the mouth and hair. It is interesting to note that despite the large added hair region (light blue in the last two columns), the model produces relatively little hair. This is most likely due to the low-resolution prior, which does not have hair in that region. In the last example, we close the mouth in multiple steps for a smooth transition.

In Fig. 9, we replace the teeth region with lips in the top row and in the second row, we remove the eyebrows. In the third row, we remove the annotations for the large hair region. One might expect the model to produce a bold head, but instead we get a head with smooth hair. The reason for this is again the conditioning on the low-resolution input. The low-resolution image indicates the presence of a dark region, which is in contrast to a bright bold head. Hence, the model decides to create some smooth hair, despite the missing annotation in the semantic input. In the last row, we remove the annotations for eye glasses. Remarkably, *DeepSEE* interprets the dark pixels from the low-resolution input as strong makeup.

In summary, *DeepSEE* allows a broad range of semantic manipulations, as long as they are roughly consistent with the low-resolution input.

3.3 Influence of Gradual Changes in the Style Matrix

We provide additional results for the *independent* model, where we walk in the latent style space. Fig. 3 and Fig. 4 show results for the *independent* model. We predict the style matrix from a low-resolution image (column two) and produce multiple gradually varying solutions. The middle image (column five) is generated via the unmodified predicted style from the low-resolution image. The images to the left (columns three and four) are generated after subtracting a δ from the original style matrix. Similarly, we generate the images on the right (columns six and seven) by adding a δ to the style matrix. As a result, we observe that larger values in the style matrix (columns six and seven) tend to yield more contrast and darker colors, in particular for lips, eyes and eyebrows. Contrary, smaller values in the style matrix (columns three and four) produce a rather bleached out image.

3.4 Disentangled Manipulations

In above section (Section 3.3), we modified the full style matrix. Now, in Fig. 10, we only manipulate some specific rows in the style matrix. This limits the changes to the corresponding semantic region. For example, the first two rows show the same output face, but with changes in the hair style. The changes in row three and four are limited to the skin area. Please zoom in to see differences in skin texture. Similarly, the rows five to seven manipulate a single semantic region only (lips, eyebrows and eyes).

3.5 Visualizations of Extreme Super-resolution

Fig. 14, Fig. 15 and Fig. 16 visualize extreme super-resolution examples in high resolution. For Fig. 14, some high-frequency components slightly differ (*e.g.* the exact trimming of the beard, or the wrinkles on the forehead), yet our model captures the main essence and identity from the low-resolution image. Given such extreme upscaling factors, it is not surprising that we do not always observe such consistent results out of the box. In the second example (Fig. 15 on the right), the upscaled image shows a young woman with a smooth skin texture. In reality, however, the ground-truth image is a middle-aged woman with wrinkles. Wrinkles are a typical example of a high-frequency component that is not clear in a low-resolution image. Most images in CelebAMask-HQ [12, 11] show young people with smooth skin. Given the low-resolution version of a middle-aged woman, the style encoder incorrectly inherits the bias of the dataset and predicts the style code of a young woman. This case highlights the benefits of an explorative super-resolution model. With *DeepSEE*, a user is now able to manipulate the style for the skin and generate a solution that matches the ground-truth. Fig. 16 shows the example from the main paper in higher resolution.

3.6 Additional Comparisons with Related Work

We extend our quantitative comparison to related work in Tables 6 and 7. Qualitatively, we add more examples for GFRNet [23] and GWAInet [24] in Fig. 11 and Fig. 12. Finally, we provide a visual comparison to methods based on facial landmarks [27, 28] in Fig. 13. Comparing to Kim *et al.*, our models achieve better results for perceptual metrics. Qualitatively, their outputs struggle with texture and shape of the more difficult areas, like hair or teeth. This outcome is not surprising because their face alignment ignores out-of-face regions, and without guidance, it is much more difficult to learn concepts that only cover a small area, as in the mouth region. In contrast, our method benefits from knowledge about semantic regions, which allows for guidance beyond facial landmarks.

3.7 Results on Other Datasets

We provide more qualitative results on two additional datasets. The first dataset, Flickr-Faces-HQ (FFHQ) [29], is a collection of high-resolution face images. The second dataset contains outdoor scenes from ADE20K [30, 31].

Table 6. Extended Quantitative Evaluation on CelebA. We copy the values from Table 1 from the main paper and add PSNR. The scores are computed on the full test set after center cropping and resizing to 128×128 and the upscaling factor is $8\times$, starting at 16×16.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
Bicubic	20.67	0.5917	0.5625	159.60
FSRNet [28] (MSE)	20.03	0.5647	0.2885	54.480
FSRNet [28] (GAN)	19.75	0.5403	0.2304	55.616
Kim *et al.* [27]	**23.29**	**0.6634**	0.1175	11.408
ours (independent)	21.85	0.6631	**0.1063**	13.841
ours (guided)	21.73	0.6628	0.1072	**11.253**

Table 7. Extended Quantitative Evaluation on CelebAMask-HQ. We copy the values from Table 1 from the main paper and add PSNR. We compare to models that require a guiding image from the same person. The 32×32 face images are upscaled $8\times$.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
Bicubic	23.13	0.6635	0.5443	125.148
GFRNet [23]	24.69	0.6726	0.3472	55.22
GWAInet [24]	**24.81**	0.6834	0.1832	28.79
ours (*independent*)	23.54	0.6770	0.1691	22.97
ours (*guided*)	23.94	**0.6887**	**0.1519**	**22.02**

For FFHQ, we downsample the images via bicubic interpolation. Then, our segmentation network predicts high-resolution segmentation masks from low-resolution images. For inference, we upscale the images with our *independent* model trained on CelebAMask-HQ [11]. Fig. 17 shows the results ($32\times$ upscaling).

The focus of our paper is on face images. However, *DeepSEE* could potentially be applied to other domains, such as outdoor scenes. We show some initial results on a subset containing outdoor scenes [31] from ADE20K [30]. We use the pre-trained segmentation network from [31] to predict the semantic regions from low-resolution images and retrain our style encoder and generator. Fig. 19 shows examples for $4\times$ upscaling.

3.8 Ablation Study

Fig. 18 shows visual examples from our ablation study. Please look at the zoomed in areas to see subtle differences.

References

1. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)
2. 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. (2019) 2337–2346
3. Zhu, P., Abdal, R., Qin, Y., Wonka, P.: Sean: Image synthesis with semantic region-adaptive normalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 5104–5113
4. 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. (2017) 1125–1134
5. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: International Conference on Learning Representations (ICLR). (2018)
6. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10). (2010) 807–814
7. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115** (2015) 211–252
8. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: Proc. icml. Volume 30. (2013) 3
9. 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** (2017) 834–848
10. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Rethinking atrous convolution for semantic image segmentation liang-chieh. IEEE Transactions on Pattern Analysis and Machine Intelligence (2018)
11. Lee, C.H., Liu, Z., Wu, L., Luo, P.: Maskgan: Towards diverse and interactive facial image manipulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 5549–5558
12. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. In: International Conference on Learning Representations (ICLR). (2018)
13. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: International Conference on Learning Representations (ICLR). (2016)
14. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. In: Computer Vision and Pattern Recognition (CVPR). (2017)
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (ICLR). (2015)
16. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision, Springer (2016) 694–711
17. Kingma, D.P., Ba, J.L.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR). (2015)

18. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. (2010) 249–256
19. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV). (2015)
20. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13** (2004) 600–612
21. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR. (2018)
22. 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: Advances in neural information processing systems. (2017) 6626–6637
23. Li, X., Liu, M., Ye, Y., Zuo, W., Lin, L., Yang, R.: Learning warped guidance for blind face restoration. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 272–289
24. Dogan, B., Gu, S., Timofte, R.: Exemplar guided face image super-resolution without facial landmarks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2019)
25. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 1251–1258
26. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
27. Kim, D., Kim, M., Kwon, G., Kim, D.S.: Progressive face super-resolution via attention to facial landmark. In: Proceedings of the 30th British Machine Vision Conference (BMVC). (2019)
28. Chen, Y., Tai, Y., Liu, X., Shen, C., Yang, J.: Fsrnet: End-to-end learning face super-resolution with facial priors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 2492–2501
29. 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. (2019) 4401–4410
30. 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. (2017)
31. Wang, X., Yu, K., Dong, C., Change Loy, C.: Recovering realistic texture in image super-resolution by deep spatial feature transform. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 606–615

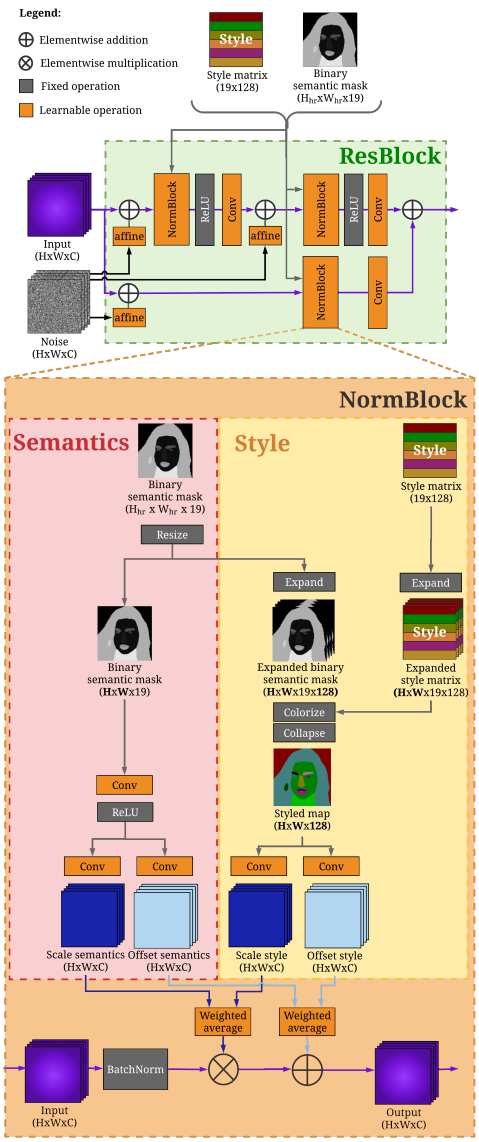

Fig. 1. ResBlock Architecture. The ResBlock consists of a series of normalization blocks (NormBlock) with a residual connection. The NormBlock injects semantics [2] and style [3]. For semantics, the NormBlock computes scale and offset modulation parameters from the binary semantic mask. For style, we first merge the regional style with the semantic mask yielding the *styled map*, from which we then compute a set of modulation parameters. A learned weighted average combines the two sets of modulation parameters, which are finally applied to the normalized input. In our ablation study (main paper), we investigate the model performance when omitting semantics or style.

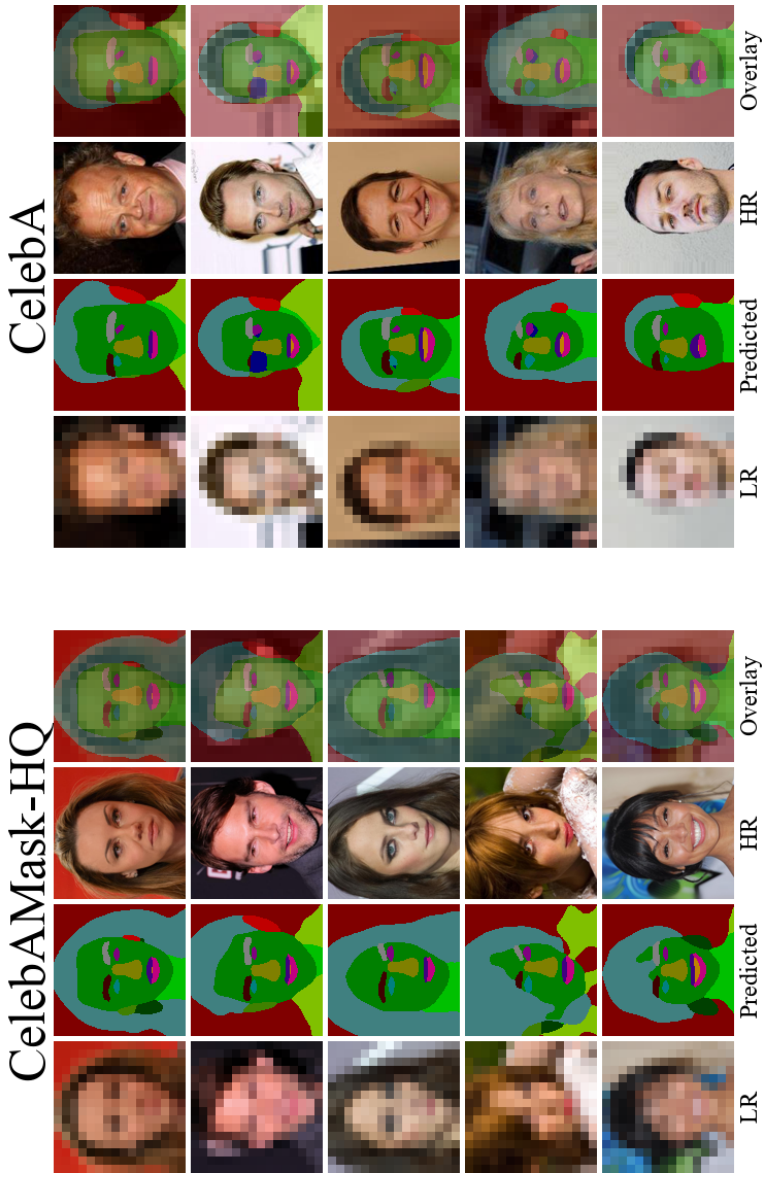

Fig. 2. Semantic Segmentation Results. We predict a high-resolution semantic mask (resolution 512×512 for CelebAMask-HQ and 128×128 for CelebA) from a low-resolution image (16×16). Columns three and five show the high-resolution image and column four and eight an overlay for a better comparison. The images are random samples from the test set.

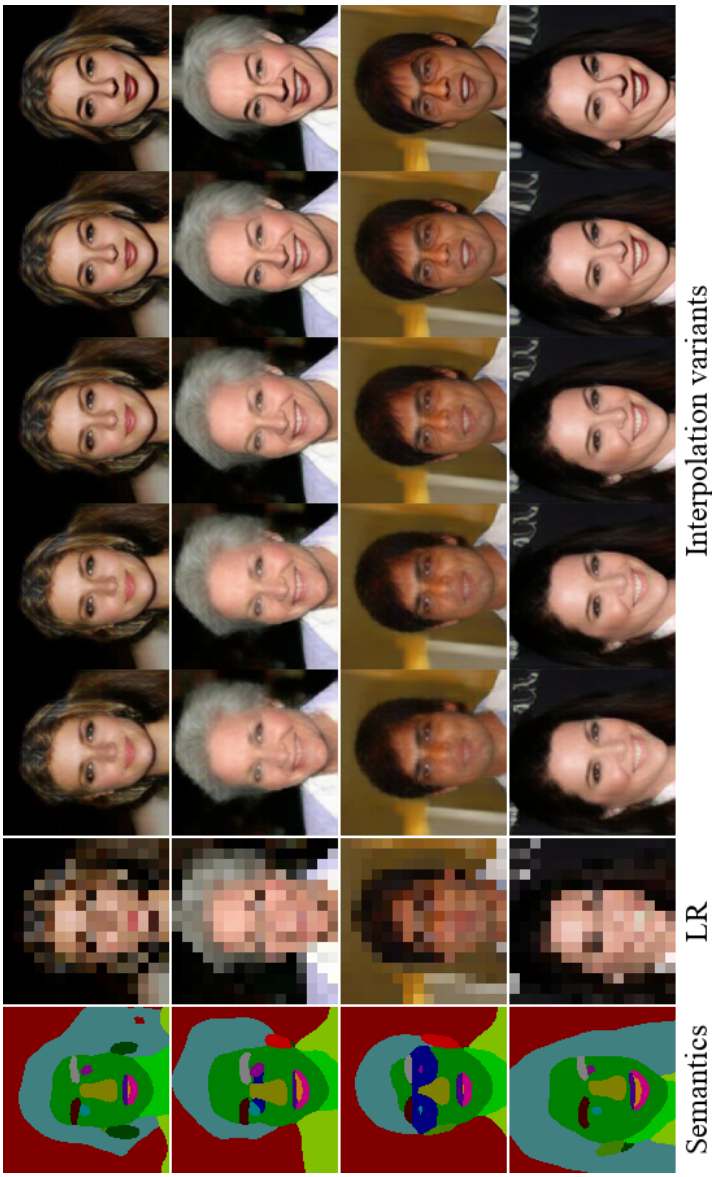

Fig. 3. Exploring Solutions by Walking the Latent Style Space (1). We predict the style from the low-resolution image (LR) and generate five smoothly varying solutions. The middle image (column five) uses the predicted style matrix without any changes. For the images on the either side, we subtract (left), respectively add (right) a linearly interpolated δ to the style matrix. We apply the same $\delta = 0.15$ to all regions.

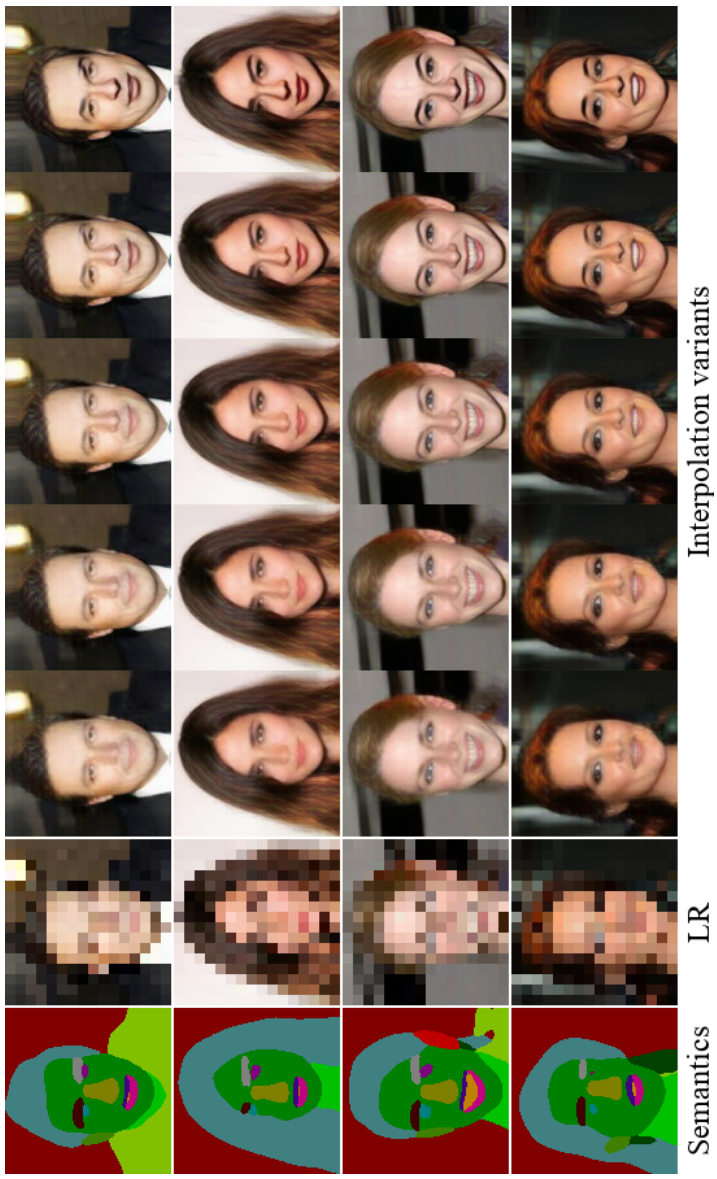

Fig. 4. Exploring Solutions by Walking the Latent Style Space (2). This visualization extends Fig. 3 with more examples. Note, how large regions, like hair, are very consistent, and small regions, such as lips and eyebrows, vary a lot more. The reason is that the low-resolution input image provide the generator with a strong for larger regions. In contrast, the generator has more freedom for small, uncertain regions and relies more on the style matrix to provide appearance information. This behaviour is wanted because it allows to preserve identities, but leaves the possibility to explore uncertain areas.

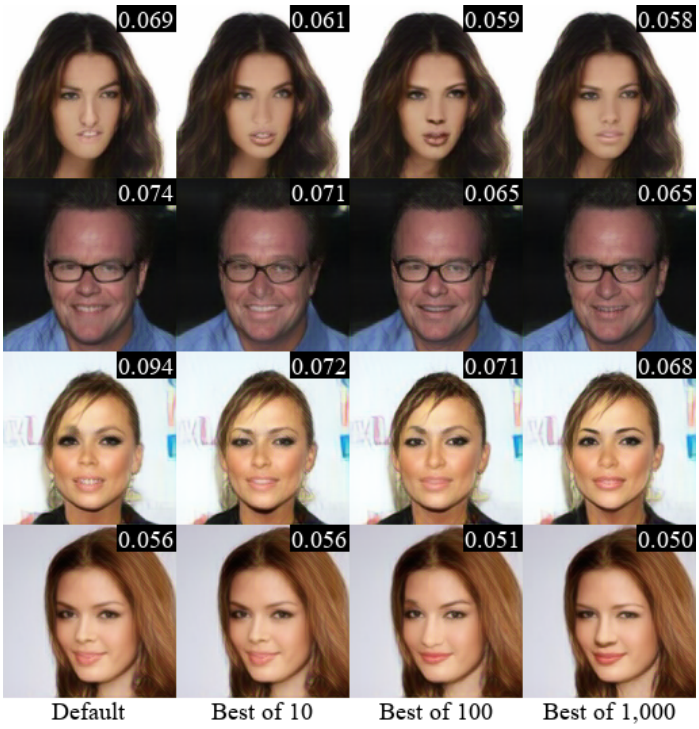

Fig. 5. Sampling in The Solution Space. We choose random test images and sample a large number of outcomes from our model by adding noise to the style matrix. We now pick 10, 100 and 1,000 random solutions and display the best-scoring image—in terms of LPIPS distance to the ground truth [21]. The score is highlighted in the top right of each image (lower is better). In contrast to our method, most related works are not able to produce more than a single solution for a given input.

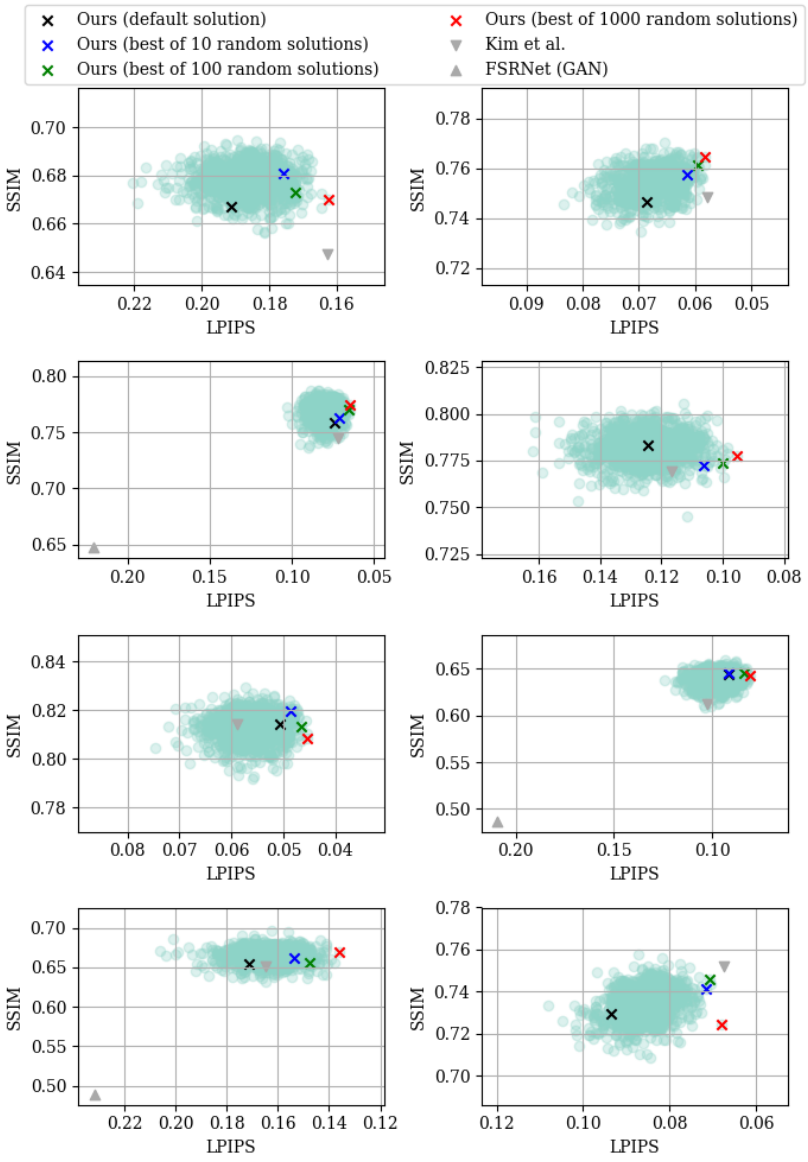

Fig. 6. Explorative Oracle Selection for Perceptual Quality. We choose 12 random test images (one image per plot) and sample many possible outcomes from our model (illustrated as point clouds). We now pick 1, 10, 100 and 1,000 random solutions and mark best-scoring image (in terms of LPIPS [21]) with a cross. Fig. 7 shows a similar plot for the image with highest SSIM [20]. While Kim et al. [27] and FSRNet [28] predict a single deterministic solution, *DeepSEE* can generate an infinite number of solutions and enables the user to pick the most desirable outcome.

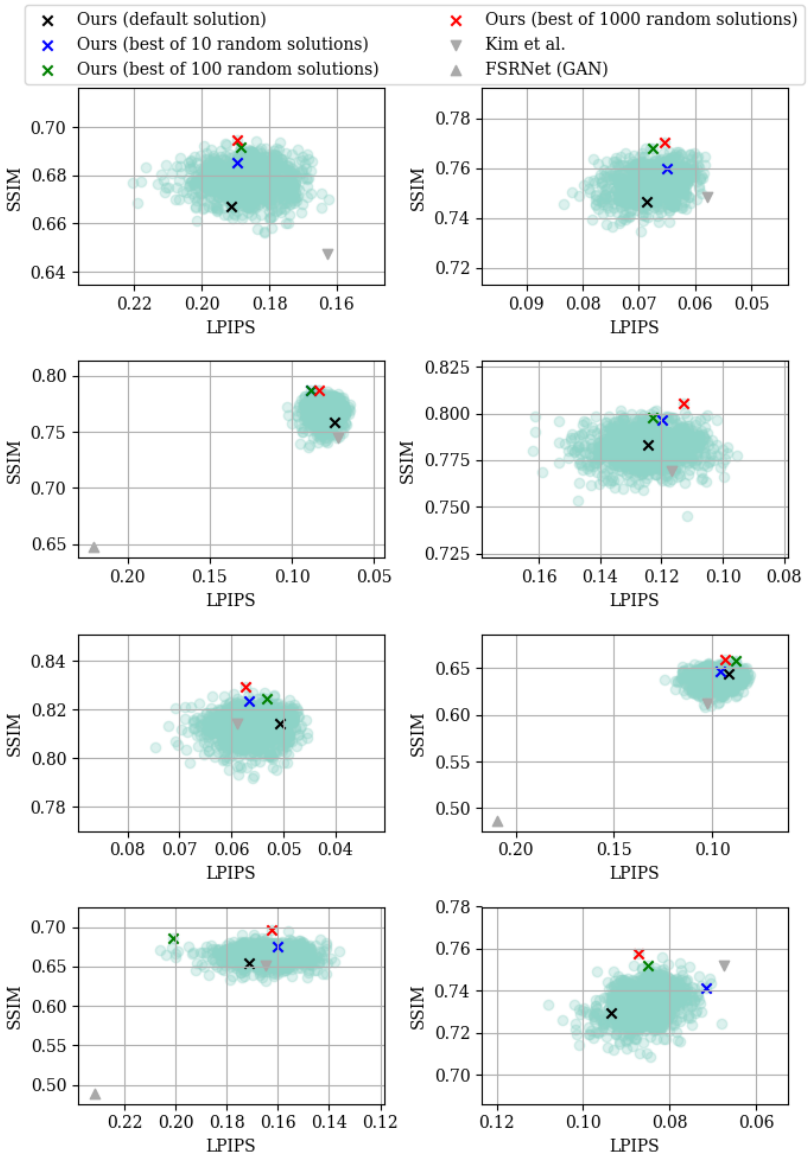

Fig. 7. Explorative Oracle Selection for Image Fidelity. We apply the same process as in Fig. 6, but sample with respect to highest SSIM [20]. In most cases, the best out of 10 random solutions is already very close to the best sample overall.

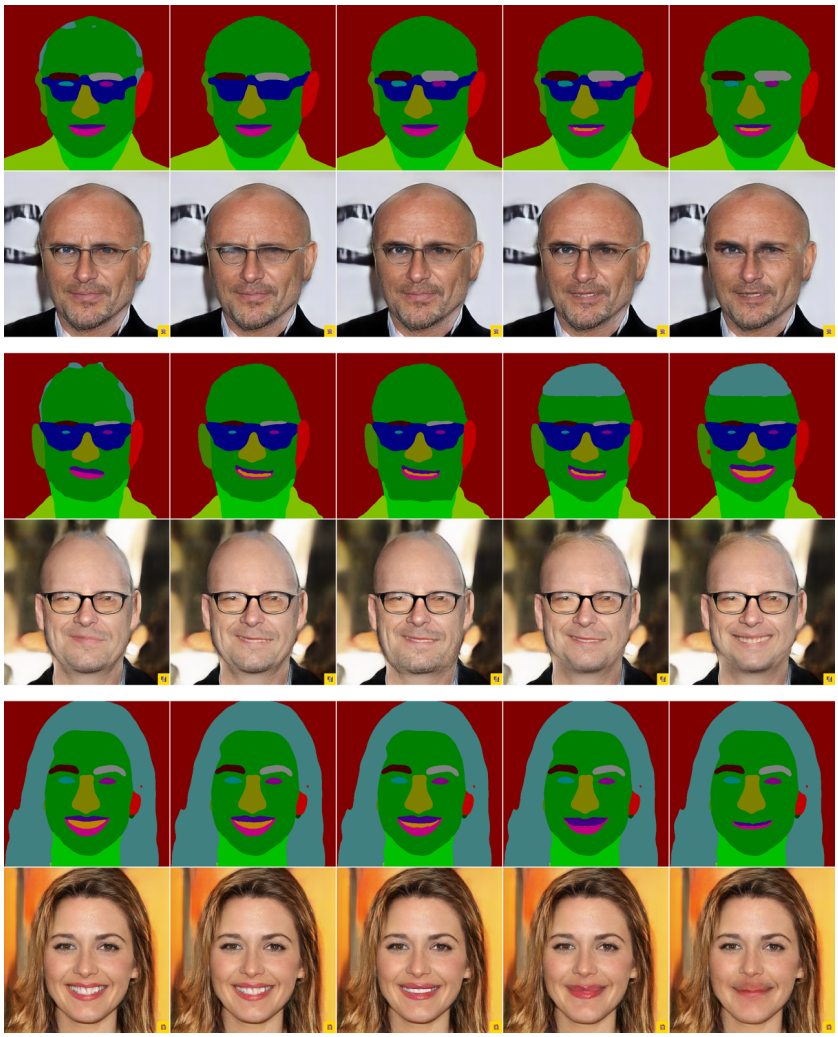

Fig. 8. Multiple Continuous Semantic Manipulations We show three examples, where we modify the semantic mask multiple times. For each sample, the first semantic mask is the original prediction and the subsequent masks have been modified. In the first example, we manipulate eyes, mouth and glasses. In the second example, we also add some hair to the top of the bold head. The third example shows a smooth transition from an open to a closed mouth.

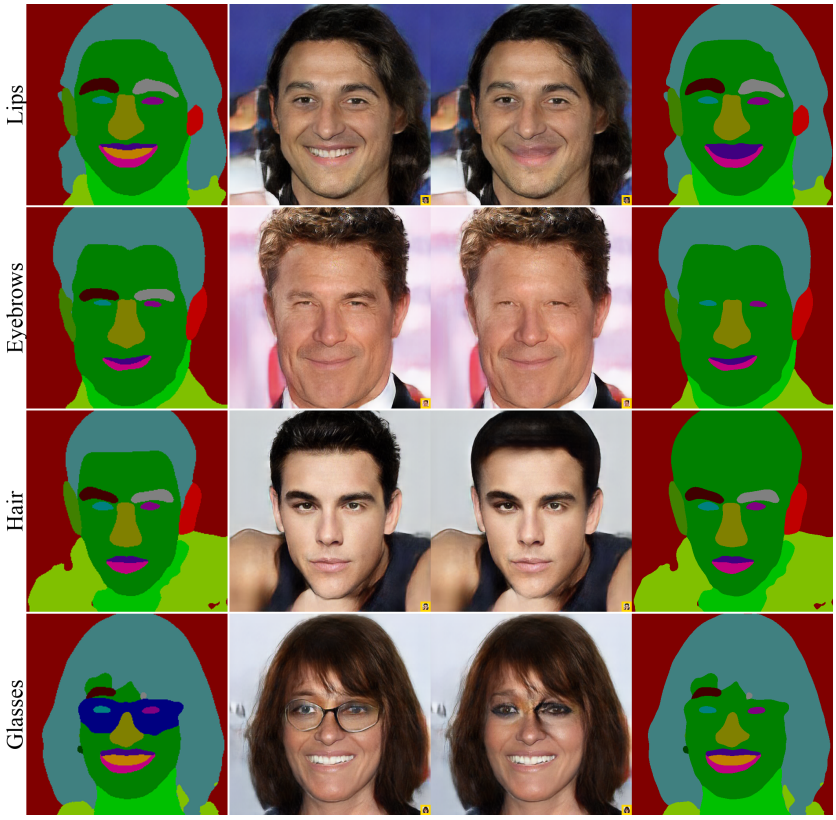

Fig. 9. Semantic Manipulations for 32× **Upscaling** In the top row, we replace the mouth region with lips, which yields a closed mouth. The second row replaces eyebrows with skin, making them disappear. The bottom two rows show cases where the manipulated semantic mask is inconsistent with the low-resolution input. For example, when replacing the annotations for hair with skin, the model still renders a dark region, resembling smooth hair. In the last row, we remove glasses, which produces a strong makeup around the eye-region. Please read Section 3.2 for a discussion.

Fig. 10. Disentangled Manipulations for the *Independent* Model. We show multiple 8× upscaled variants, where we modify the style input for the highlighted semantic region. We observe a high identity consistency across different variants, but visible changes in the selected semantic regions. For this figure, we focus on high-frequency details, which are better visible in the small, so please zoom in for better view.

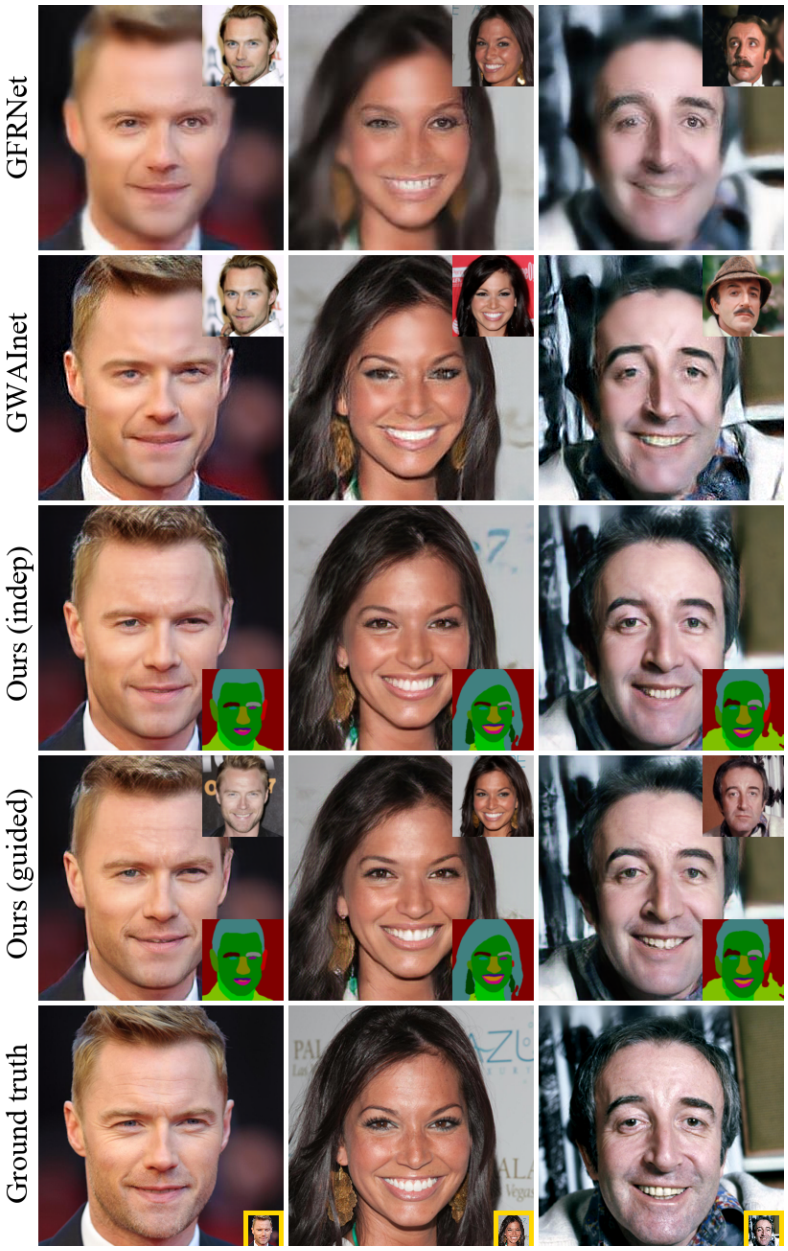

Fig. 11. Additional Comparison to Related Work (1). We visually compare to related models (GFRNet [23] and GWAInet [24]) for upscaling 8×. The small image in the top right corners shows a random image from the same person, used as guiding image. For our models, we show the predicted segmentation mask in the bottom right corner. The last row shows the ground truth with the low-resolution input image in the bottom right. Please zoom in for better view.

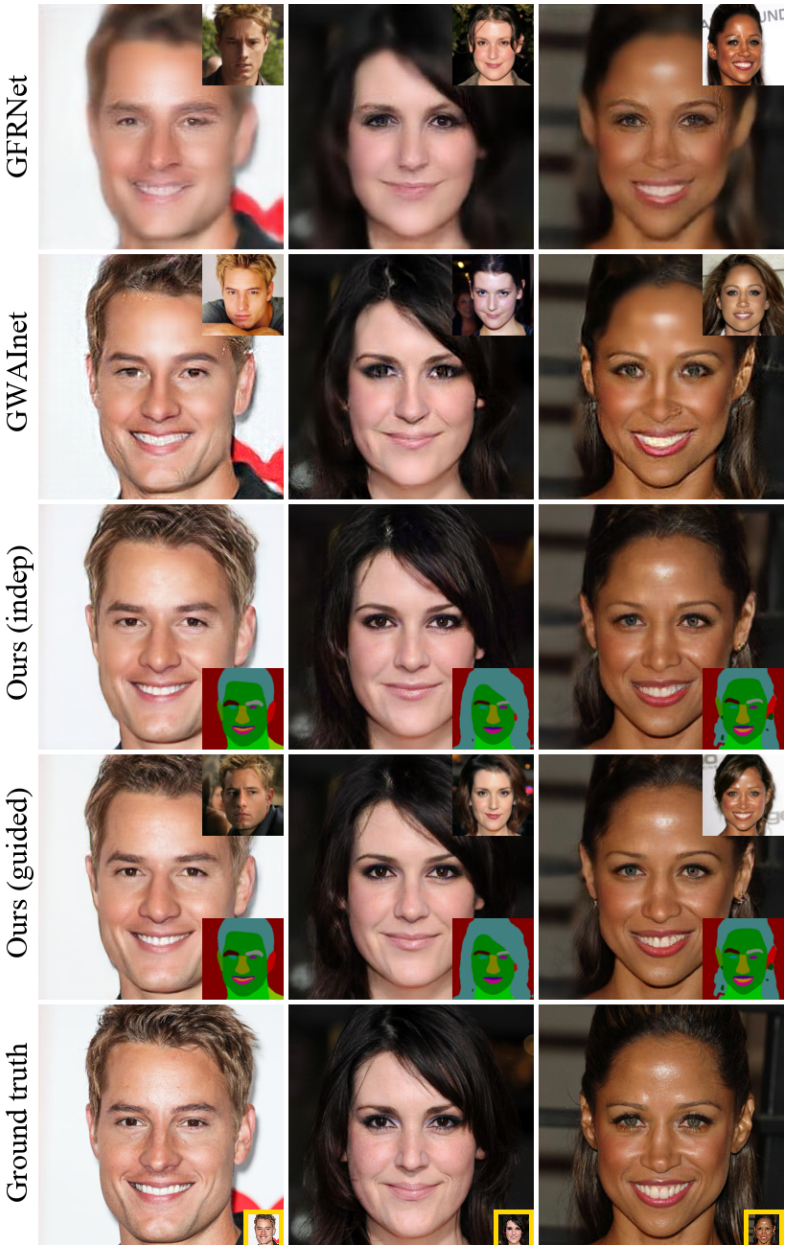

Fig. 12. Additional Comparison to Related Work (2). We extend Fig. 11 with additional examples. Please zoom in for better view.

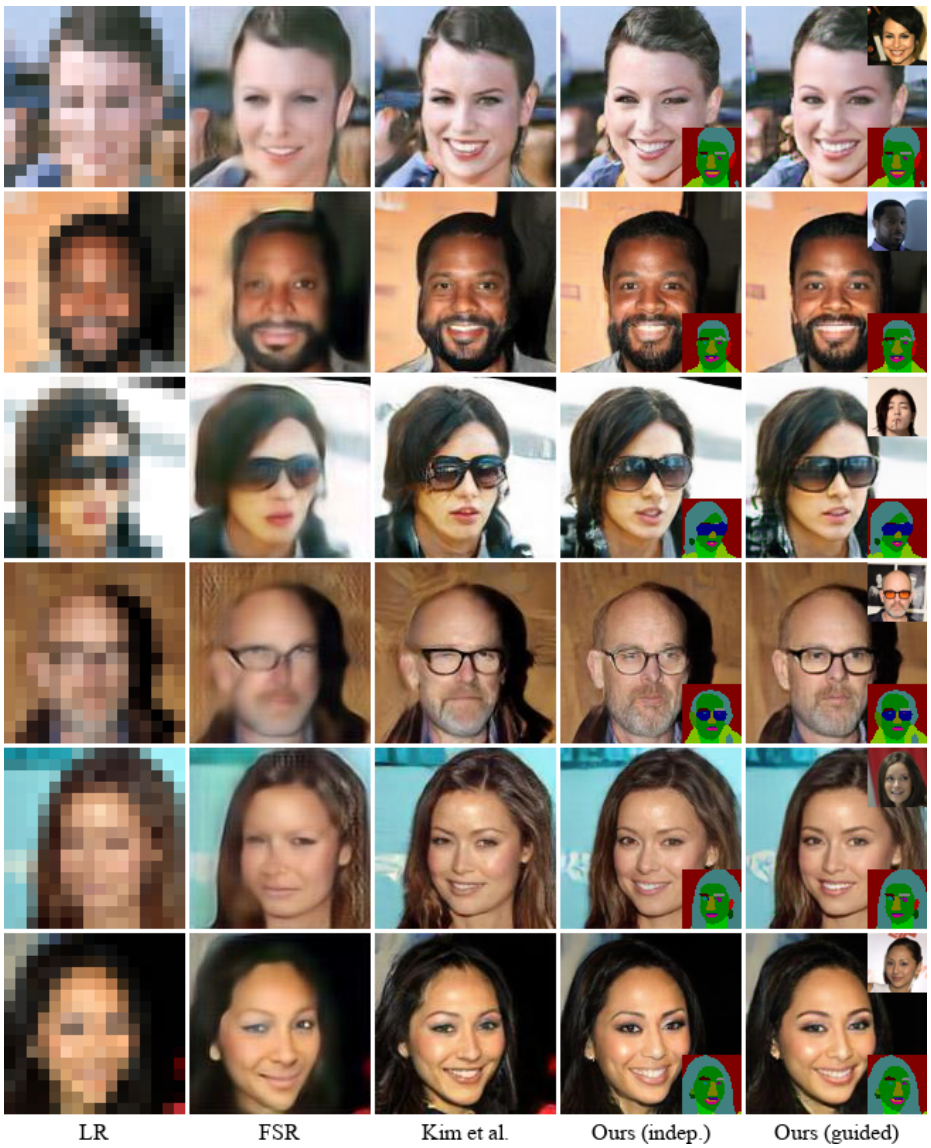

Fig. 13. Comparison with Methods Guided by Facial Landmarks. We compare to FSR [28] and Kim *et al.* [27] with upscaling factor 8×, starting at 16 × 16. For the *guided* model, the small image in the top right shows the reference image. Our method produces more realistic outputs, in particular for difficult regions, like hair, glasses or earrings.

Fig. 14. Extreme Super-resolution (1). This figure shows a larger version of the extreme super-resolution example in the main paper. The yellow image in the bottom right is the low-resolution input image, which is upscaled 32×.

Fig. 15. Extreme Super-resolution (2). This figure shows a larger version of the extreme super-resolution example in the main paper. The style encoder did not recognize the low-resolution image as a middle-aged woman, and instead produces the style code for a young woman. We discuss this observation in the main paper in more detail.

Fig. 16. Manipulating Extreme Super-resolution Output. Note how in this case the model generated a closed mouth in the default solution (top left). While a deterministic model would fail, *DeepSEE* allows the user to manipulate the segmentation mask and generate the correct version with an open mouth (top right).

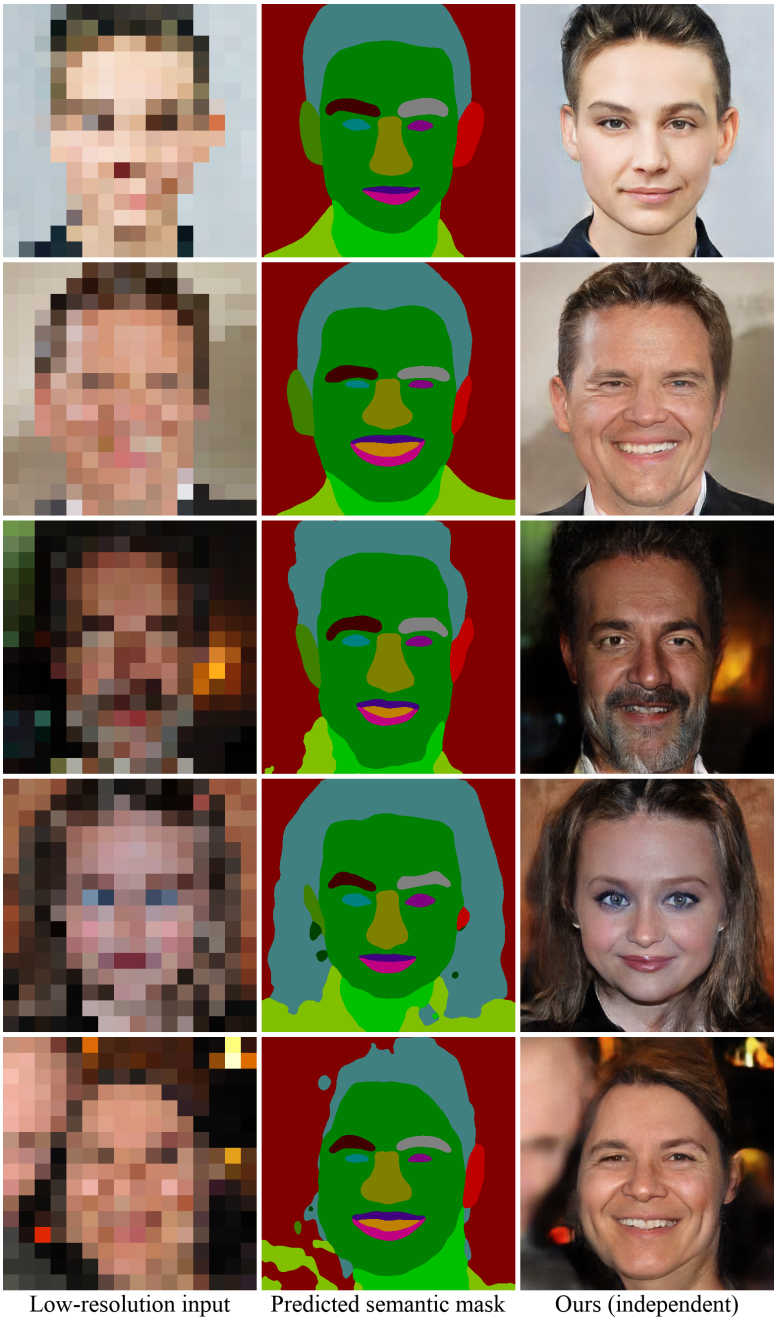

Fig. 17. Upscaling FFHQ Images [29]. We provide additional results for 32× upscaling on the Flickr-Faces-HQ Dataset [29]. We show the two inputs (*low-resolution image* and *predicted semantic mask*) and the high-resolution output of our *independent* model.

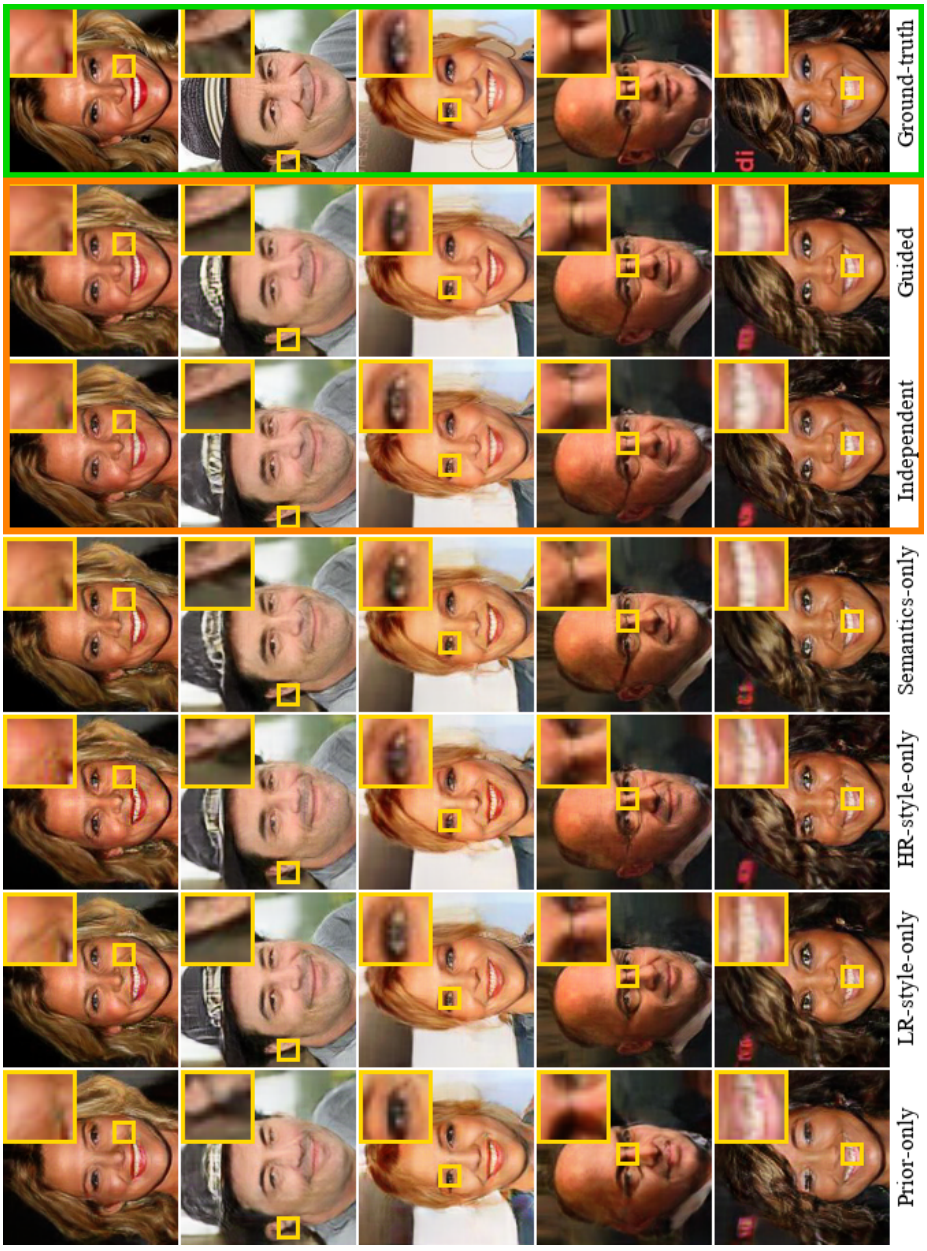

Fig. 18. Visual Ablation Study Examples. The results without any additional guidance besides the low-resolution image (*prior-only*) improve when adding style (*LR-style-only* and *HR-style-only*), semantics (*semantics-only*), or both (*independent* and *guided*). We highlight our final models (*independent* and *guided*) in orange and the ground-truth in green.

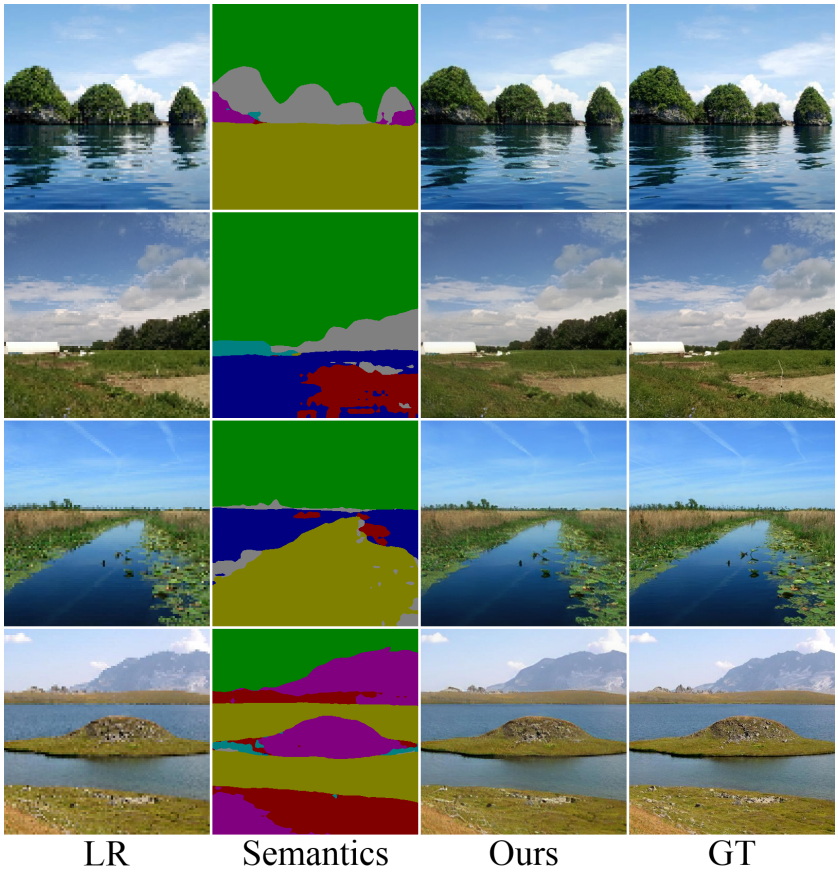

Fig. 19. Upscaling Outdoor Scenes. We show outdoor scene images from ADE20K [30, 31] for 4× upscaling.