One-Shot Synthesis of Images and Segmentation Masks
Supplementary material

Vadim Sushko[1] Dan Zhang[1,2] Juergen Gall[3] Anna Khoreva[1,2]
[1]Bosch Center for Artificial Intelligence [2]University of Tübingen [3]University of Bonn
{vadim.sushko,dan.zhang2,anna.khoreva}@bosch.com, gall@iai.uni-bonn.de

A. Qualitative comparisons to prior image-mask GAN models

A qualitative comparison of OSMIS to prior image-mask GAN models, SemanticGAN [7] and DatasetGAN [11], is presented in Fig. A, corresponding to the quantitative comparison of these models from Table 2. The displayed samples were generated with a checkpoint that achieved the lowest SIFID [9]. Like OSMIS, SemanticGAN was trained from scratch, using a single provided image-mask pair as real data. On the other hand, the training of DatasetGAN consisted of two stages: pre-training of the StyleGAN [5] backbone architecture on the single provided training image, and training a label synthesis branch with manual segmentation annotations of generated images. In our one-shot setup, since StyleGAN typically collapsed to generating the same image, annotating a single generated sample was enough to train the label synthesis branch.

As seen from Figure A, both SemanticGAN and DatasetGAN suffer from memorization issues, always producing the same image that repeats the layout of the training sample. In Table 2 this is reflected in very low LPIPS diversity scores achieved by both models. In addition, SemanticGAN shows unstable training in our one-shot regime, which results in a low visual quality of generated images and noisy annotations (note poor performance in SIFID and mIoU in Table 2). For DatasetGAN, we observed no such instabilities, which made the manual annotation of generated images straightforward. Despite a good visual image quality and accurate manual annotation of masks (high mIoU in Table 2), the low diversity of DatasetGAN prevents it from producing useful data augmentation for one-shot segmentation tasks (see Table 7).

In contrast, OSMIS achieves high diversity and visual quality of generated image-masks at the same time. For example, in the examples from Fig. A our model can change the number of sails, horse riders, sumo wrestlers, or cars, at the same time editing the layout of the backgrounds, while still preserving the realism of objects. Such structural diversity of OSMIS enables its effective generation of data augmentation for one-shot segmentation tasks (see Sec 4.2).

B. Additional details on the application of OSMIS to one-shot segmentation tasks

B.1. Details of the experimental setup

Tables 5 and 6 show the performance of one-shot segmentation networks using different data augmentation strategies. The simplest strategy is to use no data augmentation, when the fine-tuning of networks is performed only on a single provided image-mask pair. When fine-tuning with our synthesized data augmentation, we extend the pool of the available data with $n_1 = 85$ samples generated by OSMIS. Finally, when adding standard data augmentation to the two previous strategies, we apply random combinations of image-mask flipping, zooming, and rotation to the samples from the pool. The exact method of utilizing data augmentation depends on the segmentation network, as described next.

OSVOS [2] fine-tunes weights of a pre-trained segmentation network on the image and mask of the first frame of a given video sequence. At each fine-tuning epoch, we double the batch size and randomly add generated image-mask pairs to the original data. Therefore, we keep the 50%-50% ratio between real and synthetic data, which we found to yield the best video segmentation performance.

STM [8] scans a given video sequence frame-by-frame, starting from the first frame, for which a mask annotation is provided. This image-mask pair, as well as each K-th pair of a video frame and its segmentation prediction are added to a spatio-temporal memory bank. The memory bank is used to make the segmentation prediction of the latest video frames more accurate. To employ data augmentation, we added synthesized image-mask pairs to the STM memory bank at step 0, before processing the first video frame. To fit the memory bank into GPU memory, we had to limit the number of added samples to 10, which were sampled randomly from the synthetic pool.

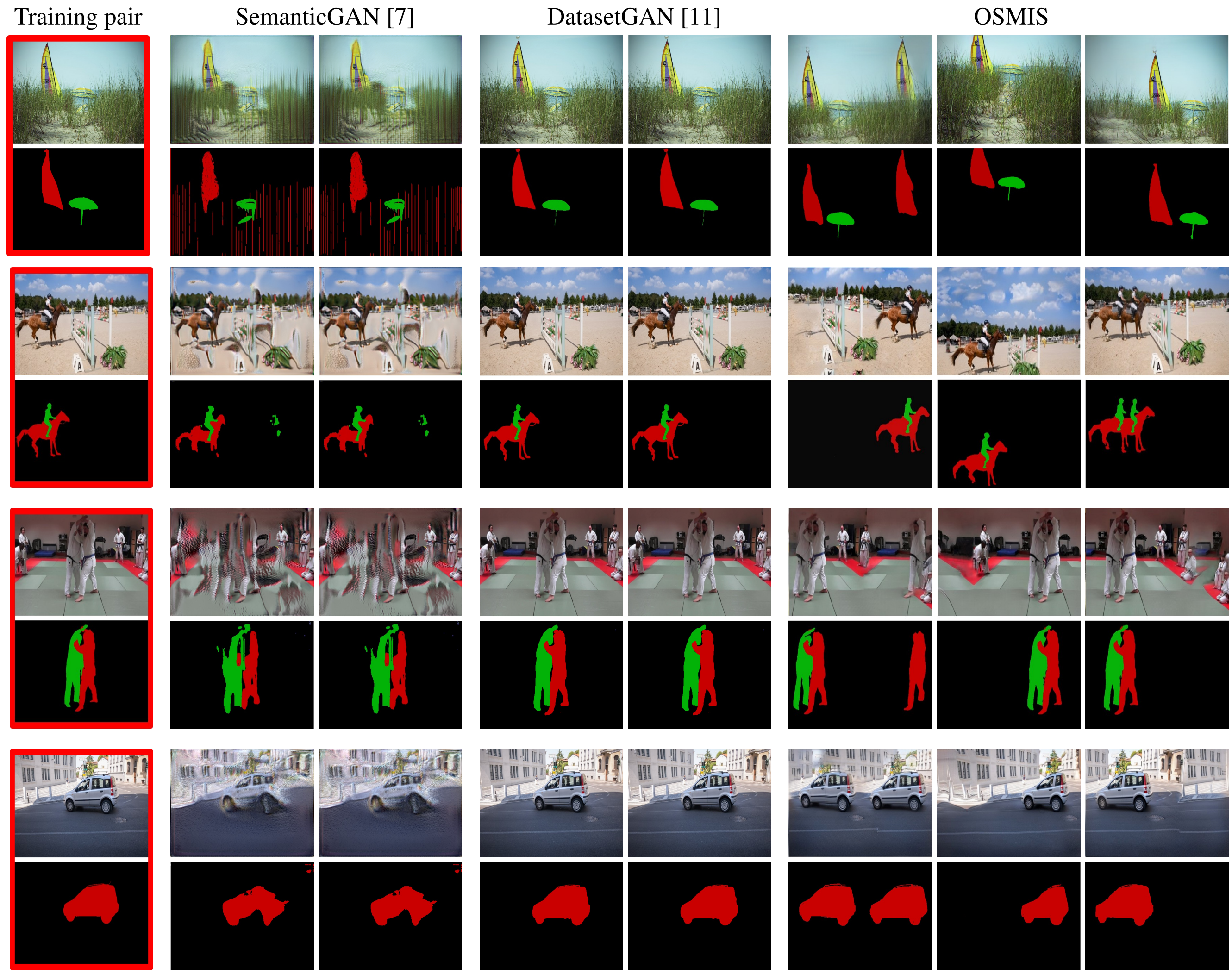

Figure A. A quantitative comparison of OSMIS to previous image-mask GAN models SemanticGAN [7] and DatasetGAN [11]. Both the comparison models suffer from memorization, repeating the layout of the training samples, while SemanticGAN also achieves poor visual quality of images and masks due to training instabilities. In contrast, OSMIS achieves both diversity and quality, placing foreground objects in different locations in the scene and editing the layouts of backgrounds.

RePRI [1] trains a small pixel-level classifier given a single support image-mask pair containing an object of a previously unseen class. We simply provide synthetic image-mask pairs as data augmentation for the original data. To fit the extended support set into GPU memory, we limited the number of added samples to 10. This way, the task of RePRI could be technically regarded as 11-shot semantic image segmentation, where all the available support data originates from a provided data sample.

B.2. Ablation on filtering out bad-quality samples

Filtering out noisy synthetic examples before forming a pool of synthetic samples is an important step to achieve good performance of data augmentation. For example, using generated image-mask pairs without filtering resulted in modest or negative performance gains for one-shot segmentation networks (see Table A). On the contrary, a simple strategy to filter out 15% of bottom-ranked generated images by SIFID, computed after the first pooling layer of the InceptionV3 network, helps to reduce the impact of bad-quality augmentation and, in effect, substantially improve the final segmentation performance.

However, we observed that the SIFID metric is biased towards low-level image statistics, such as color and texture distributions, and is not indicative of the quality of generated images at higher scales. We illustrate this in Fig. B, where we display visual examples of images at different levels of SIFID, obtained after the first pooling layer, second

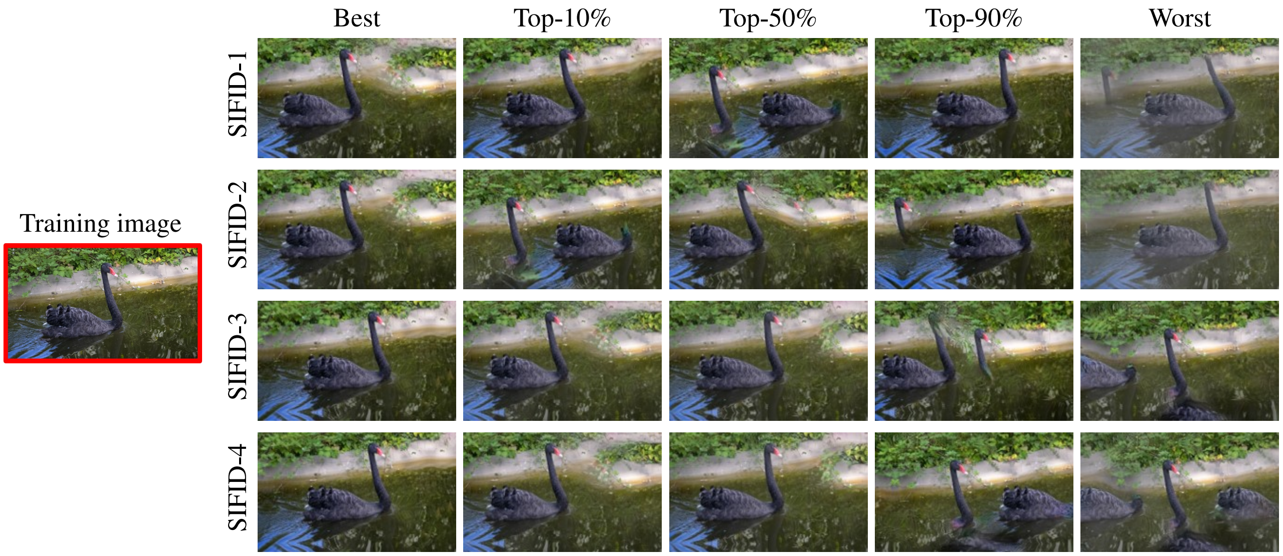

Figure B. Generated images shown for different levels of SIFID, computed at various InceptionV3 layers. We observed that SIFID at the earliest InceptionV3 layers is biased towards low-level image statistics, such as colors and small textures, and is not indicative of image quality at higher scales (appearance of objects, layout of backgrounds). Thus, to filter out noisy generated examples, we use a joint ranking of images at different InceptionV3 layers.

Data selection	η	OSVOS, DAVIS-16	RePRI, COCO^0
		$\mathcal{J}\&\mathcal{F}$	mIoU
Reference w/o augmentations		78.5 (+0.0) ±0.3	31.2 (+0.0) ±0.1
No data selection	-	78.7 (+0.2) ±0.6	30.7 (-0.5) ±0.5
Only SIFID-pool$_1$	15%	79.3 (+0.8) ±0.5	32.2 (+1.0) ±0.4
SIFID-{1,2,3,4} (ours)	5%	79.3 (+0.8) ±0.6	31.9 (+0.7) ±0.4
	10%	79.6 (+1.1) ±0.4	32.6 (+1.4) ±0.2
	15%	**79.8 (+1.3)** ±0.3	**32.8 (+1.6)** ±0.2
	25%	79.7 (+1.2) ±0.3	32.3 (+1.1) ±0.2
	50%	79.5 (+1.0) ±0.3	32.0 (+0.9) ±0.1

Table A. Impact of synthetic data selection strategies on one-shot segmentation performance. Bold and underlined show the first and second best performance.

pooling layer, pre-classifier features, and the final features of the InceptionV3 network (denoted as SIFID-1,2,3,4).

To account for the quality of generated images at different scales, we ranked synthesized examples by a joint ranking, taking the average of their ranks across different SIFIDs. As seen in Table A, filtering out noisy examples using this strategy helps to boost the performance of one-shot segmentation networks. Furthermore, we observed that it helps to significantly decrease the performance variance between different runs, which generally increased while using synthetic data augmentation in our experiments.

Finally, we conduct an ablation on how many bottom-ranked images should be filtered for optimal performance. Table A demonstrates that the filtering rate should be neither too low nor too high: filtering out only 5% or 10% leaves some low quality images that are harmful for the data augmentation efficiency, while filtering too many samples (25%, 50%) decreases the diversity of the synthetic data pool and thus also diminishes its effectiveness.

Overall, we conclude that data filtering is a crucial step that is needed to achieve high performance gains with the help of synthetic data augmentation. Table A shows that our proposed data selection scheme is effective at filtering out bad generated examples, which results in higher performance of one-shot segmentation networks without notably increasing their variance between runs.

C. Architecture of OSMIS and training details

The architecture of the OSMIS generator and discriminator is summarized in Tables B and C. We build upon the structure of One-Shot GAN [10], which utilizes ResNet blocks for both the generator and discriminator, enables multi-scale gradients (MSG) [3] by employing skip connections between the latest generator layers and the low-level discriminator $D_{low-level}$, and provides control over the final image resolution by changing the input noise shape.

To achieve image-mask synthesis at a high resolution of 384x640, we set the input noise shape to 3×5, use 8 ResNet blocks in the generator, 4 ResNet blocks for the low-level discriminator $D_{low-level}$, and 4 blocks for the object and layout discriminators D_{object} and D_{layout}. Before feeding the intermediate features $F(x) = D_{low-level}(x)$ of an input image x to D_{object}, we process it by the masked content attention module (MCA), which forms N content representations, corresponding to objects or background in the image. Thus, for the object discriminator we use a batch size which is N times higher than for other discriminator parts.

We train OSMIS with the ADAM optimizer [6], using a batch size of 3, momenta $(\beta_1, \beta_2) = (0.5, 0.999)$, and a learning rate of 0.0002. During training, we use an exponential moving average of the generator weights with a decay of 0.9999, which is used at inference. P_0 from Eq. (5) is set to 15000 epochs. We extend the differentiable augmentation (DA) pipeline used in [10] by using the whole set of transformations as proposed in [4], which we found beneficial for image quality and diversity. Considering the provided segmentation mask, we modify the discriminator feature augmentation (FA), ensuring that it does not interfere with the learning of the appearance of foreground objects. For this, the content FA is applied only to the representation of the background, while for the layout FA, the mixed spatial areas are sampled respecting the object boundaries in the segmentation mask. In our experiments, we observed this to be beneficial for the visual quality of images, as the model learnt to preserve the objects' appearance better.

References

[1] Malik Boudiaf, Hoel Kervadec, Ziko Imtiaz Masud, Pablo Piantanida, Ismail Ben Ayed, and Jose Dolz. Few-shot segmentation without meta-learning: A good transductive inference is all you need? In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[2] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[3] Animesh Karnewar and Oliver Wang. Msg-gan: multi-scale gradient gan for stable image synthesis. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[4] Tero Karras, Miika Aittala, Janne Hellsten, S. Laine, J. Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[5] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[7] Daiqing Li, Junlin Yang, Karsten Kreis, Antonio Torralba, and Sanja Fidler. Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[8] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *International Conference on Computer Vision (ICCV)*, 2019.

[9] Tamar Rott Shaham, Tali Dekel, and T. Michaeli. Singan: Learning a generative model from a single natural image. In *International Conference on Computer Vision (ICCV)*, 2019.

[10] Vadim Sushko, Jurgen Gall, and Anna Khoreva. One-shot gan: Learning to generate samples from single images and videos. In *Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2021.

[11] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Operation	Input	Size	Output	Size
ConvTransp2D	z	(64,1,1)	up_0	(256,3,5)
ResBlock-Up	up_0	(256,3,5)	up_1	(256,3,5)
ResBlock-Up	up_1	(256,3,5)	up_2	(256,6,10)
ResBlock-Up	up_2	(256,6,10)	up_3	(256,12,20)
ResBlock-Up	up_3	(256,12,20)	up_4	(256,24,40)
ResBlock-Up	up_4	(256,24,40)	up_5	(256,48,80)
ResBlock-Up	up_5	(256,48,80)	up_6	(256,96,160)
ResBlock-Up	up_6	(256,96,160)	up_7	(128,192,320)
ResBlock-Up	up_7	(128,192,320)	up_8	(64,384,640)
Conv2D, TanH	up_5	(256,48,80)	image_3	(3,48,80)
Conv2D, TanH	up_6	(256,96,160)	image_2	(3,96,160)
Conv2D, TanH	up_7	(128,192,320)	image_1	(3,192,320)
Conv2D, TanH	up_8	(64,192,320)	image_0	(3,384,640)

Table B. The OSMIS generator. The configuration is presented for the input noise of size (3×5) and the final resolution of (640×384).

Operation	Input	Size	Output	Size
Low-level discriminator $D_{low-level}$				
Conv2D	image_0	(3,384,640)	feat_0	(32,384,640)
Conv2D	image_1	(3,192,320)	feat_1	(8,192,320)
Conv2D	image_2	(3,96,160)	feat_2	(16,96,160)
Conv2D	image_3	(3,48,80)	feat_3	(32,48,80)
ResBlock-Down	feat_0	(32,384,640)	down_0	(64,192,320)
ResBlock-Down	down_0 feat_1	(64,192,320) (8,192,320)	down_1	(128,96,160)
ResBlock-Down	down_1 feat_2	(128,96,160) (16,96,160)	down_2	(256,48,80)
ResBlock-Down	down_2 feat_3	(256,48,80) (32,48,80)	F	(256,24,40)
Object discriminator D_{object}				
MCA	F	(256,24,40)	F_con	N×(256,1,1)
ResBlock-Down	F_con	N×(256,1,1)	cont_0	N×(256,1,1)
ResBlock-Down	cont_0	N×(256,1,1)	cont_1	N×(256,1,1)
ResBlock-Down	cont_1	N×(256,1,1)	cont_2	N×(256,1,1)
ResBlock-Down	cont_2	N×(256,1,1)	cont_3	N×(256,1,1)
Layout discriminator D_{layout}				
Conv2D	F	(256,24,40)	F_lay	(1,24,40)
ResBlock-Down	F_lay	(1,24,40)	lay_0	(1,12,20)
ResBlock-Down	lay_0	(1,12,20)	lay_1	(1,6,10)
ResBlock-Down	lay_1	(1,6,10)	lay_2	(1,3,5)
ResBlock-Down	lay_2	(1,3,5)	lay_3	(1,3,5)

Table C. The OSMIS discriminator. The configuration is presented for the input noise of size (3×5) and the final resolution of (640×384).