Progressive Growing of Conditional Adversarial Networks

Gabriel Rodriguez **Jakob Manz** **Ryan Cherifa** **August DuMont Schütte**

Abstract

We investigate how to progressively grow a conditional adversarial network, largely based on the pix2pix model. While layer-wise progressive growth from low to high resolution has shown remarkable results in the normal GAN setting, we demonstrate that it can be adapted in a conditional setting to achieve high resolution and visually appealing image-to-image translations with stable training.

1 Introduction and related work

The task of transforming representations of different images for computer vision and image processing has traditionally been tackled application-specifically [3,8,21,33]. Over the last decade, Convolutional Neural Networks resulted in significant advances for image related tasks [10, 20, 23]. However, the need for application-specific loss functions has limited the generalization of such methods. More recently, Generative Adversarial Networks (GANs) [13] have demonstrated high generative capabilities and produced sharp images for a wide range of applications [5, 12, 13, 26].

The conditional setting (cGANs) [24] has been exploited in various ways to develop a unified approach for image-to-image translations [24, 31, 32]. In particular, the pix2pix model [17] provides a general purpose solution with good results for up to 256×256 pixels. Overall, GANs still face significant problems: Among others, GANs and cGANs are still largely restricted to low resolution images, because at higher resolutions the vanishing gradient problem arises as discrimination is easier [4, 18].

Within the conditional setting, different strategies have resulted in improved models: In [14] the problem of mode collapse is tackled by a combination of generators in a multi-modal mapping. Most recently, sharp and high resolution images were generated with new multi-scale generator and discriminator architectures [30]. Conversely, in the unconditional setting, the progressive growth of the network structure (ProGAN) led to superb results for high resolution [18].

To provide an alternative model for high resolution image-to-image translations in a conditional setting, we explore a method of progressive layer growing for a network, similar to the pix2pix. Our model is largely based on the pix2pix implementation [1] in TensorFlow [1], as well as the ProGAN implementation [2].

2 Methods

In cGANs a generator learns a nonlinear mapping from a conditional input x and a random noise vector z to an output representation y, $G : \{x, z\} \rightarrow y$ [13]. While the generator is trained to produce representations that are indistinguishable from real output mappings, a discriminator network D acts as the counterpart and aims to expose fake outputs.

2.1 Objective

In the conditional setting the value function is expressed as: $\mathcal{V}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(\mathbf{x}, \mathbf{y})] + \mathbb{E}_{x,y}[\log(1 - D(\mathbf{x}, G(\mathbf{x}, \mathbf{y})))]$ and G and D play the minimax game so that $G^* = \arg\min_G \max_D \mathcal{V}_{cGAN}(G, D)$. Based on the work of [25] the pix2pix showed stable training and good results if an $L1$ norm is added to the objective of the generator so that: $G^* = \arg\min_G(\max_D \mathcal{V}_{cGAN}(G, D) + \lambda \mathcal{V}_{L1}(G))$ where $\mathcal{V}_{L1}(G) = \mathbb{E}_{x,y,z}[||y - G(\mathbf{x}, z)||_1]$. While new losses such as the WGAN-GP [2] resulted in higher quality outputs, we adopt the loss of the original pix2pix to focus on the layer growth objective.

[1] TensorFlow pix2pix: `https://github.com/affinelayer/pix2pix-tensorflow`

[2] ProGAN: `https://github.com/tkarras/progressive_growing_of_gans`

Moreover, in [17] it was determined that feeding a random noise vector along with the conditional representation does not increase variation in the output. We accept these limitations to the pix2pix and do not further investigate how to increase stochasticity.

2.2 Network architecture

The network architecture can be separated into the generator and discriminator as displayed in Figure 1. Each layer block that is added to the network consists of a 3×3 convolution, followed by batch normalization [16] and a nonlinear activation. In the encoder and discriminator, downsampling by average pooling is performed after each block to halve the resolution. For the decoder of the generator nearest neighbour upsampling doubles the resolution.

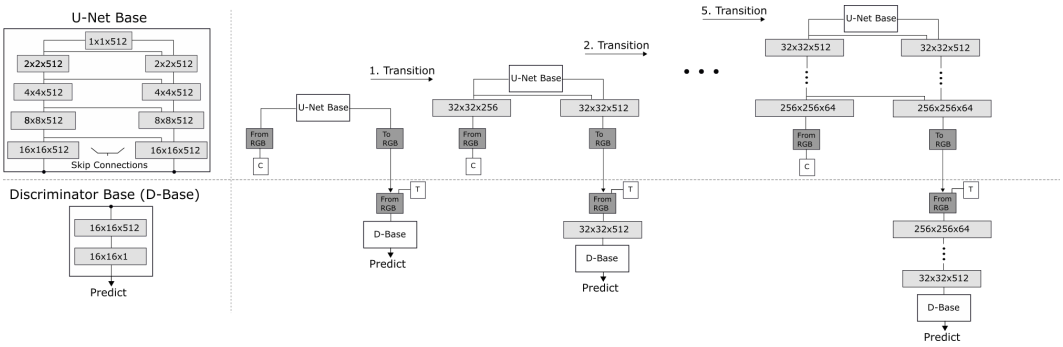

Figure 1: Training starts with the base network at a low spatial resolution of 16×16 pixels. As the training progresses, layers are incrementally added, each doubling the spatial resolution. $N \times N \times D$ refers to the spatial resolution and channel size respectively. The input depth in the generator is twice as large as the channel size due to the concatenation of the previous output and the skip layers.

2.2.1 Generator

The generator is constructed as an auto-encoder [15], in which the input is convolved and downsampled until it passes a bottleneck layer, after which it is deconvolved and upsampled again. Skip connections between opposite layers in the encoder and decoder are added to ensure that low-level information is not lost when passing the bottleneck, resulting in the "U-Net" shape [27].

2.2.2 Discriminator

In the pix2pix model, the discriminator is restricted to only model high frequency information by penalizing structure only at the scale of patches, therby effectively modeling the output as a Markov random field, with independence between pixels separated by more than one patch length [22]. This model assumption of the "PatchGAN" has been explored and demonstrated as effective in many previous works [9, 11, 22]. When adapting a PatchGAN with a discriminator receptive field of 16×16 pixels, we generated the most visually appealing images. This larger initial resolution, as compared to the ProGAN, appears sensible as our datasets show limited meaningful information at very low resolutions.

2.3 Progressive growing of layers

Progressive growing of GANs is motivated by the idea that learning output representations in steps is easier [7, 18]. Figure 2 displays how we adapt this method to our structure, with three transition zones within the network. In each zone, the problem of different channel sizes depending on the resolution block is resolved by using a "to-RGB" and "from-RGB", 1×1 convolution, that transforms the channel size to and from the RGB channels. When the resolution is doubled we smoothly interpolate between the previous layer and the newly added layer to ensure stable training [18]. After the interpolation of a new layer, the old to- and from-RGB blocks are discarded and no longer affect the outputs.

2.4 Optimization

We alternate between one gradient descent step on the discriminator, followed by a gradient descent step on the generator, by using minibatch SGD with the Adam Solver [19]. Similar to the pix2pix

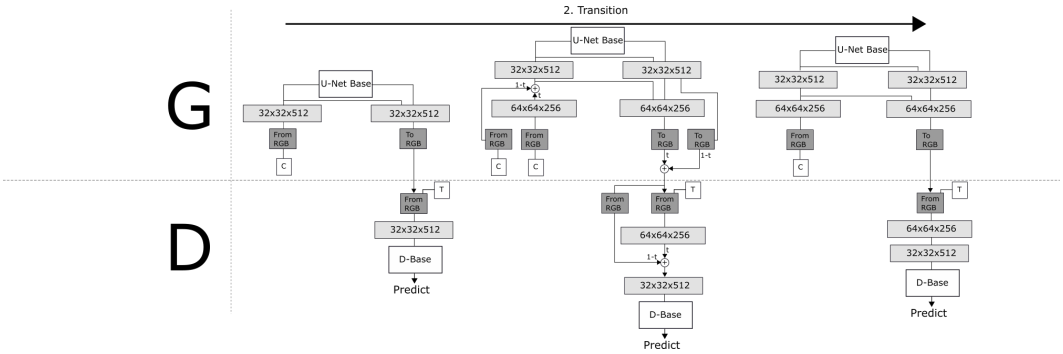

Figure 2: Here we demonstrate the transition from 32×32 to 64×64 spatial resolution. During the interpolation phase, the new layer is faded in smoothly by linearly increasing t from 0 to 1. The to-RGB and from-RGB blocks project the features to and from the RGB representation. The doubling and halving of the spatial resolution is achieved by up- and downsampling, not displayed in the Figure for readability.

model we apply batch normalization [16], using the statistics of the test batch during testing. While the learning rate of the Adam Solver is reset for each new layer in the ProGAN, we do not change the Adam defaults and leave it open how strongly this could improve training.

3 Results

We validate our model with several experiments and by comparing it to the outputs of the TensorFlow implementation of the pix2pix. Please refer to Appendix A for a larger set of translated images.

3.1 Datasets

To analyse the general network architecture we experiment on the Facade dataset [29]. We use approximately 507 training and 101 validation images, of 256×256 resolution what makes it computationally less demanding. In order to investigate the performance at high resolution and compare the synthesized images on an evaluation metric we test on the Cityscapes dataset [6]. The images display urban street scenes at a resolution of 1024×512, which we preprocess to 512×512 pixels with standard computer vision methods. Our training set consists of 3035 images, while we use 442 for validation.

Figure 3: Comparison of generated Facade images at 256×256 resolution. From left to right: Inputs, pix2pix, Ours, Targets.

3.2 Evaluation of the model

We observed that stable training and high quality output representations were closely linked to adapting the original pix2pix architecture in our growth setting. When replacing the PatchGAN discriminator by a fully connected layer to enable lower initial resolutions, we faced training instabilities, mainly in the form of negative discriminator outputs. However, our optimal PatchGAN size is significantly smaller then the 70×70 spatial resolution proposed in the pix2pix. We determined the best trade-off between a low initial resolution and a sufficiently large PatchGAN to be 16×16 pixels, by visually inspecting the output representations.

The image quality was not improved when experimenting with other than leaky-ReLU activation functions in different layers, which is in agreement with [18]. Only after the final 1×1 convolution in the to-RGB layer, a tanh-activation resulted in visually more appealing representations. However, with a nonlinear output mapping, there were stronger loss fluctuations during the early interpolation of two layers. Since these irregularities quickly faded away the improvement in quality was more significant and we believe that an adjustment of the weight initialization could solve this minor problem.

As opposed to the pix2pix, the network performance was not improved by including dropout layers after the bottleneck, nor was training more stable. This seems reasonable as we train fewer parameters at low resolution and only increase the complexity of our network gradually, once more information in form of fine-scale details in the images becomes available. Nevertheless, this means we are discarding the last source of variation in the network.

A comparison of image translations for the Facade dataset between our model and the pix2pix can be seen in Figure 3. As expected, the overall quality seems to be similar because both networks operate at the same final resolution of 256×256 pixel. Some fine-scaled details, such as the balconies, appear to be more accurate in our representation.

3.3 Performance on segmentation classifier

The evaluation of artificially generated or translated images remains a difficult problem and a range of metrics have been proposed [18, 28]. To obtain a more objective measure than visual appearance, we compute the mean intersection over union (mIoU) and the pixel accuracy of the Cityscapes outputs, as frequently done in literature [17]. As the segmentation model we use a pre-trained PSPNet for semantic segmentation based on [34]. Both metrics are calculated on the test set, by comparing the resulting segmentation of the targets (ground truth) to the outputs of our model and pix2pix respectively.

Figure 4: Comparison of generated Cityscapes images. From left to right: Inputs, pix2pix (256×256 pixels), Ours (512×512 pixels), Targets (512×512 pixels).

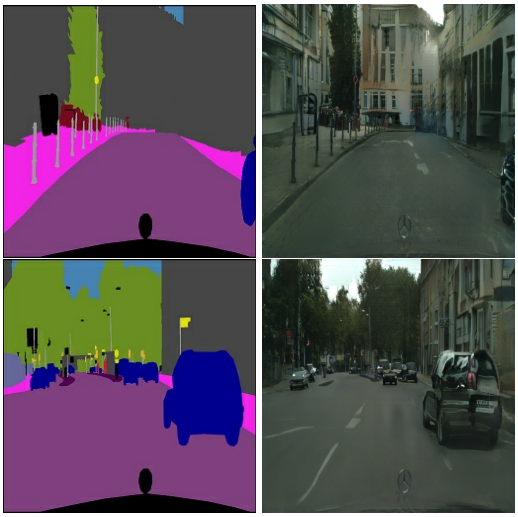

Figure 5: Removing (top) or adding (bottom) objects. See Figure 4 for the original images.

Model	Pixel Accuracy	mIoU
Our model (512×512)	0.91	0.53
Our model (256×256)	0.90	0.52
pix2pix (256×256)	0.88	0.45

Table 1: Segmentation metrics for test images from the Cityscapes dataset.

A comparison of the outputs can be seen in Figure 4, while Table 1 lists several segmentation metrics. Our generated images are visually more appealing and perform better regarding the mIoU-scores (in both the original 512×512 resolution, as well as for a downsampled representation of 256×256 pixels).

3.4 Evaluation of the conditional setting

To demonstrate the importance of the conditional setting we present additional experiments on our trained model. While we cannot generate new images by feeding in random noise, we can accurately manipulate the translation between complex image representations: Figure 5 shows how to add and subtract distinct objects, by removing the class from the conditional input, as compared to the original images in Figure 4. This tractability of the model differs strongly from the normal GAN setting. Please see Appendix A for another example.

4 Discussion

Our results demonstrate the impact of stable training at high resolution and how progressive layer growth is a viable method for achieving this objective in the conditional setting. Nevertheless, the model faces significant limitations and the translated images are far from realistic. Especially fine-scale details in the Cityscapes images, such as cars or pedestrians are deformed and less accurate. Because of little modifications to the pix2pix architecture and objective, the improvement seems to be largely based on the stable training at 512×512 pixels, without increasing the model capability of translating more complicated structures in the images.

The computationally demanding training combined with the available resources limited our fine-tuning ability. Moreover, we did not experiment with novel objectives, normalization methods or the introduction of output variation, which could significantly improve the model. Therefore, it is not surprising that the method cannot compete with state of the art networks, such as [30]. That being said, we believe that by including more recent ideas to our model, the proposed method could reach existing benchmarks.

5 Summary

We have presented a new conditional adversarial network for high resolution image-to-image translations. The model is based on the combination of progressive growing of layers and the pix2pix architecture. We validated our approach on two datasets, in which we produce visually more appealing results and outperform the pix2pix on mIoU and pixel accuracy scores. Finally, we showed how the conditional setting allows for manipulations of the input-output mapping.

A Appendix

Figure 6: Additional Cityscapes output comparision: From left to right: Inputs, pix2pix (256×256 pixels), Ours (512×512 pixels), Targets (512×512 pixels).

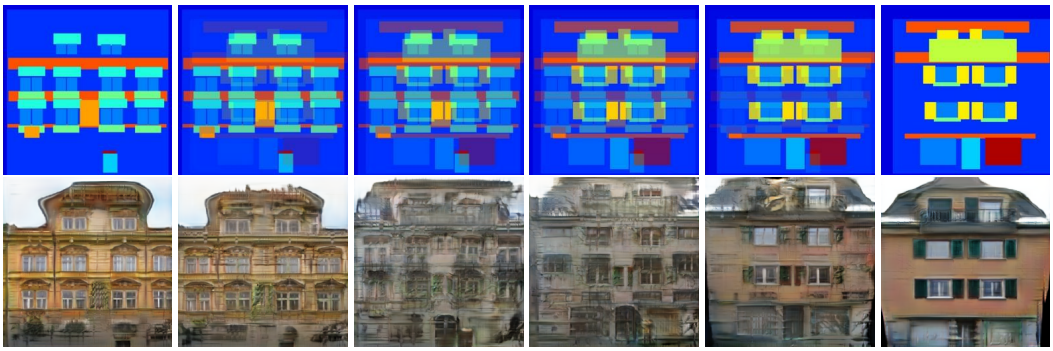

Figure 7: Post-training evaluation on the Facade dataset: Linear interpolations between two different inputs (top) and our model output (bottom).

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, et al. TensorFlow: Large-scale machine learning on heterogeneous systems. *CoRR*, 2015. Software available from tensorflow.org.

[2] Martin Arjovsky, Soumith Chintala, and Leon Bottou. Wasserstein GAN. *ArXiv e-prints*, 2017.

[3] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A non-local algorithm for image denoising. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, CVPR '05, pages 60–65, Washington, DC, USA, 2005.

[4] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. *CoRR*, abs/1707.09405, 2017.

[5] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In D D Lee, M Sugiyama, U V Luxburg, I Guyon, and R Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2172–2180. Curran Associates, Inc., 2016.

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] Ishan P. Durugkar, Ian Gemp, and Sridhar Mahadevan. Generative multi-adversarial networks. *CoRR*, abs/1611.01673, 2016.

[8] A. Alexei Efros and William T. Freeman. Image quilting for texture synthesis and transfer. *Computer Graphics (Proc. SIGGRAPH'01)*, 35, 07 2001.

[9] Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the International Conference on Computer Vision - Volume 2*, ICCV '99, pages 1033–, Washington, DC, USA, 1999.

[10] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *CoRR*, abs/1609.02087, 2016.

[11] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016.

[12] Arnab Ghosh, Viveka Kulharia, Vinay P. Namboodiri, Philip H. S. Torr, and Puneet Kumar Dokania. Multi-Agent Diverse Generative Adversarial Networks. *CoRR*, abs/1704.0, 2017.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In Z Ghahramani, M Welling, C Cortes, N D Lawrence, and K Q Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[14] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved Training of Wasserstein GANs. *CoRR*, abs/1704.0, 2017.

[15] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. *CoRR*, abs/1611.0, 2016.

[18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. *CoRR*, abs/1710.1, 2017.

[19] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[21] Pierre-Yves Laffont, Zhile Ren, Xiaofeng Tao, Chao Qian, and James Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Trans. Graph.*, 33(4):149:1–149:11, July 2014.

[22] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. *CoRR*, abs/1604.04382, 2016.

[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014.

[24] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[25] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. *CoRR*, abs/1604.07379, 2016.

[26] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *CoRR*, abs/1505.0, 2015.

[28] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *CoRR*, abs/1606.03498, 2016.

[29] Radim Tyleček and Radim Šára. Spatial pattern templates for recognition of objects with regular structure. In *Proc. GCPR*, Saarbrucken, Germany, 2013.

[30] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. *CoRR*, abs/1711.1, 2017.

[31] Xiaolong Wang and Abhinav Gupta. Generative Image Modeling using Style and Structure Adversarial Networks. *CoRR*, abs/1603.0, 2016.

[32] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image De-raining Using a Conditional Generative Adversarial Network. *CoRR*, abs/1701.0, 2017.

[33] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. *CoRR*, abs/1603.08511, 2016.

[34] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. *CoRR*, abs/1612.01105, 2016.