Improving Controllability of Unconditional Image Synthesis by Transformer-based GAN

Anonymous Author(s)
Affiliation
Address
email

Abstract

State-of-the-art results of data-driven unconditional image synthesis has been achieved by the style-based generator architecture. It enables an unsupervised separation of high-level attributes (e.g., pose and identity when trained on human faces) and intuitive scale-specific control of the synthesis by learning a less entangled latent space in layers of different scales. However, such scale-specific latent codes assign the same style to all the regions of features in a same scale, regardless of the styles for regions with different semantics. In this paper, we propose to control the styles of different regions independently, and thus enable a fine-grained control in each layer by a set of semantic-aware latent codes. For modeling the correspondences between the set of latent codes and different region features for the same semantics in each scale, we adopt Transformer as our backbone. A Transformer-based model enables regions of different semantics to attend to semantic-aware styles so that styles for different semantics can be disentangled within the same scale. We conduct both quantitative and qualitative experiments and show that our model is able to provide fine-grained style control of image synthesis and high-fidelity results.

1 Introduction

The perceived quality and controllability of data-driven unconditional image synthesis have seen significant improvements in recent years [1, 2, 3]. Notably, the style-based generator architecture has achieved state-of-the-art results and is widely used in many applications, e.g., facial editing, style transfer and image morphing. It has demonstrated that the style-based generator is able to control scale-specific styles of image synthesis. For example, when trained on human faces, the style-based generator is able to control the coarse styles (e.g., pose and identity) by the latent code in low-resolution layers, as well as the fine styles (e.g., color scheme and micro-structure) by the latent code in high-resolution layers.

The success factor of the style-based generator architecture lies in the learning of disentangled latent space. Instead of feeding the input latent code $z \in Z$ only to the beginning of the network, the style-based generator adopts a mapping network f to transform it to an intermediate latent code $w \in W$. Affine transforms then produce styles that control the layers of the synthesis network g via adaptive instance normalization (AdaIN). To separate the correlations of latent codes between neighboring layers, the style-based generator adopts a 'normalization-modulation' strategy to first remove the "old styles" from previous latent code by normalization and assign "new styles" from the latent codes in current layers via modulation. Such a strategy has demonstrated separated scale-specific attributes of image synthesis.

However, the style-based generator can suffer from two limitations. First, the normalization operation that normalizes the mean and variance of each feature map potentially destroying any information

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

found in the magnitudes of the features relative to each other. Although some alternative solutions have been proposed [2], e.g., weight normalization, in the sacrificy of latent separation. This is because the weight normalization is based on the assumption of i.i.d distribution of input activation. However, the input activations are often effect each other. Besides, all positions in the features of each scale are controlled by the same set of styles, which can make the latent space entangled under the same scale. For example, it is hard to further separate coarse styles in the same scale away, e.g.pose and identity in low-resolution layers.

In this paper, we propose to model the relations between different positions in features and different styles for different semantic by a Transformer-based generator architecture. Specifically, the Transformer-based generator inherit the architecture of mapping network from the style-based generator and embed it to an intermedia latent spcae. Instead of a single w, we use a set of w with each for different semantics. We use attention machanism to find related styles for each position. We conduct extensive experiments and show that, our model outperforms previous style-based generator in terms of both percievd results and fine-grained control of image synthesis.

2 Related works

In this section, we review [**?**]

3 Approach

4 Experiments

Implementation details We use 8 NVIDIA V100 GPUs for training. We use the code of ronsinalty.

Experiment Setup

Dataset We conduct experiments on three dataset.

4.1 Style interpolation

4.2 Style Mixing

4.3 Image attribute editing

5 Limitations and societal impacts

6 Conclusion

References

[1] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019.

[2] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, pages 8110–8119, 2020.

[3] Heliang Zheng, Jianlong Fu, Yanhong Zeng, Jiebo Luo, and Zheng-Jun Zha. Learning semantic-aware normalization for generative adversarial networks. *Advances in Neural Information Processing Systems*, 33:21853–21864, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? **[TODO]**
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]