

GANs Spatial Control via Inference-Time Adaptive Normalization

Karin Jakoel* Liron Efraim* Tamar Rott Shaham
Technion - Israel Institute of Technology

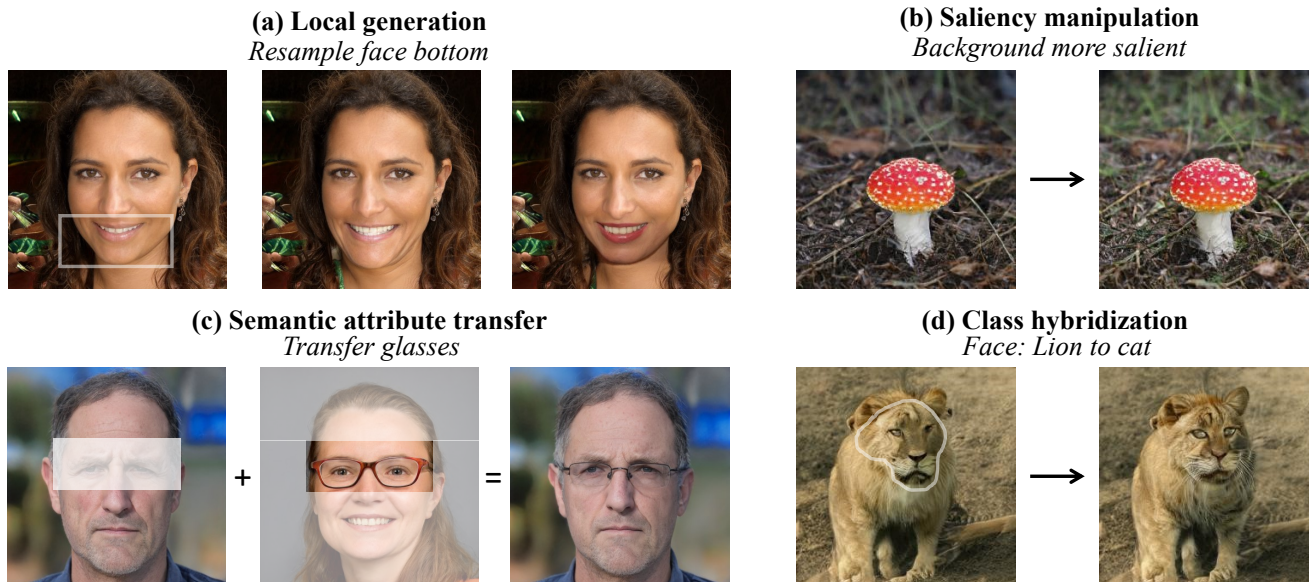


Figure 1: **Spatial control over image generation with inference-time adaptive normalization.** We suggest a new spatial-aware normalization technique applied on pre-trained GANs at *test-time* that enables *spatial* control over the generation process. This allows to use pre-trained GAN models like [12, 2] for a variety of applications that modify only specific image regions, such as (a) randomly generating sub-regions within an image (while keeping the rest of it fixed), (b) controlling the saliency of different objects, (c) transferring local attributes from one image to another and (d) modify a specific part of the image to belong to a different class.

Abstract

We introduce a new approach for spatial control over the generation process of Generative Adversarial Networks (GANs). Our approach includes modifying the normalization scheme of a pre-trained GAN at test time, so as to act differently at different image regions, according to guidance from the user. This enables to achieve different generation effects at different locations across the image. In contrast to previous works that require either fine-tuning the model’s parameters or training an additional network, our approach uses the pre-trained GAN as is, without any further modifications or training phase. Our method is thus completely generic and can be easily incorporated into common GAN models. We show our technique to be useful for solving a line

of image manipulation tasks, allowing different generation effects across the image, while preserving the GAN’s high visual quality.

1. Introduction

Since first introduced by Goodfellow *et al.* [8], unconditional GANs have led to a revolution in the computer vision community, with a rapid improvement over the visual quality of the generated scenes, as well as the ability to generate images with growing resolution [11, 2, 12, 19, 13]. Consequently, pre-trained unconditional GAN models have been incorporated as a building block in many image editing and manipulation tasks, enabling high flexibility while ensuring high quality performances [7, 10, 21, 9]. Most of such meth-

ods use the pre-trained GAN as a black box and perform the desired manipulation in the generator latent space instead of in the image space itself. However, current State-of-The-Art (SOTA) GANs models map a latent vector that has no notion of spatial coordinates, into a two dimensional image. Therefore, any manipulation of the latent code affects the whole image, causing a global manipulation effect. This negates the ability to control the generation process at different spatial locations, and impedes the use of GANs for tasks that require different operations at different locations across the image.

In this work, we suggest a new method for adapting a pre-trained GANs *at test time* to allow *spatial* control over the generation process. Our method includes modifying the model’s original normalization scheme (that was used while training) act differently on different image regions, according to guidance from the user. As we show, although the GAN model was trained with a global fix normalization, using our spatial adaptation at inference time allows to utilize the power of pre-trained GANs models, while controlling the generation process locally. This is done without any further modifications to the model or additional training, which is often a challenging task by itself when dealing with adversarial training.

We show that our method is very easy to apply and can be integrated into popular GANs models like BigGAN [2] and StyleGAN [12, 13]. In addition, as no training is required, we enjoy very short execution time compared to other methods that require a targeted training phase. We exemplify the contribution of our method for the tasks of local generation, local attribute transfer, class hybridization and saliency manipulation. As we show, for all these applications, our method allows to modify only a specific image region while keeping the rest of it intact. This is while maintaining high visual quality as we show in Fig. 1.

2. Related work

Image manipulation with pre-trained GANs. In recent years, pre-trained GAN models have been incorporated into image manipulation schemes for various applications. This is usually done by manipulating the latent code to achieve the desired effect. For example [20] finds meaningful directions in a progressively growing GAN (PGG) model trained on faces, in order to semantically edit facial attributes. Another example is [7] that suggests to optimize direction in the latent space of BigGAN [2] in order to change cognitive properties of the images such as memorability, aesthetics, and emotional valence. A recent line of works [10, 21, 9] extend this to reveal steering directions corresponding to semantically meaningful image transformations in BigGAN’s latent space. StyleGAN [12] is used in [4] to perturb a latent code of an image to obtain modified image views. However,

since all these generators map the latent code that has no notion of spatial dimensionality into the full image, any change in the latent representation affects the whole image, causing a global effect. Our approach offers spatial control over the manipulation effect by incorporating inference time spatial adaptive normalization, and allows to manipulate only a specific image region.

Spatial adaptive normalization. The idea to use location-dependent normalization mechanism for image generation was first introduced in SPADE [18] for the task of semantic image translation. This concept have been extended in various of followup works [26, 23, 22, 14] and have been quickly adopted for other tasks [24, 15, 25]. All these normalization techniques include learned parameters and therefore need to be incorporated during training. On the other hand, our approach is applied only at *inference time* and does not require any training phase.

Local control over GANs. Lately, several methods that control spatial aspects of generative process have been proposed [3, 27], capable of high quality results. However, in contrast to our work, these models require training an additional network that encodes spatial characteristics, whereas our framework uses only the pre-trained GAN without the need to train any additional component. The most closely related method to ours in this aspect is [1] which presents impressive results for the task of locally editing images according to text description. For BigGAN, this method achieves spatial control by masking feature maps, whereas we focus on adapting the normalization unit.

3. Inference-time Adaptive Normalization

In most SoTA GANs architecture ([2, 12, 13]), the latent code is embedded into the generation process through the normalization units. This is one by applying a z -dependent denormalization operation right after the normalization mechanism (*e.g.* Batch-Norm in BigGAN [2], AdaIN in StyleGAN [12]) at each of the generator layers. That is, at the n -th layer of the model, the denormalization gain and bias parameters $\gamma^n, \beta^n \in \mathbb{R}^{1 \times 1 \times C_n}$ are calculated from the latent code $z \in \mathbb{R}^{1 \times 1 \times C_z}$ by a linear layer L^n , *i.e.* $\gamma^n, \beta^n = L^n(z)$. These parameters are then duplicated along the spatial dimensions to create two corresponding maps $\hat{\gamma}^n, \hat{\beta}^n \in \mathbb{R}^{H_n \times W_n \times C_n}$ with the same dimensions of the n -th layer normalized feature map $f^n \in \mathbb{R}^{H_n \times W_n \times C_n}$. These denormalization maps are then applied to f^n such that

$$\hat{\gamma}^n \odot f^n \oplus \hat{\beta}^n, \quad (1)$$

where \odot, \oplus represent element-wise product and sum respectively, as also described in Fig. 2a. Therefore, note that this mechanism imposes that any change in the latent code z will be directly applied to the full image space causing a global

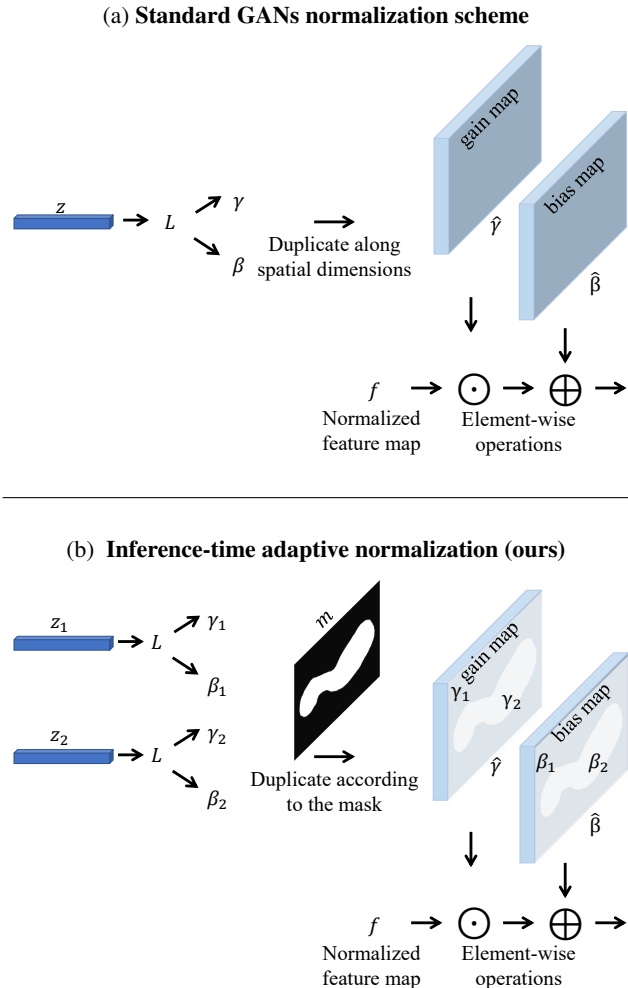


Figure 2: **Inference-time adaptive normalization.** In contrast to the standard GAN’s normalization scheme that applies the same global fix normalization at all spatial locations (a), we allow the normalization operation to vary spatially according to a guidance map (b). Therefore, instead of letting only a single latent code to govern the whole image, we are able to blend several different latent codes to control the generation process at different image regions. We perform this adaptation at test-time, and thus can use pre-trained GAN models with no additional training. In this illustration we generate an image from two different latent codes z_1, z_2 according to a binary map m .

effect, without any ability to control the manipulation effect across different spatial locations.

Our goal is to enable spatial control over the generation process. Namely, given a latent representation z , we would like different image locations h, w to be modified by different operations $\phi_{h,w}$. We achieve this by modifying the normalization mechanism *at test time*. Instead of using a simple duplication operation across the spatial dimension to create

the gain and bias maps, as done in training, we suggest constructing locally-varying denormalization maps that allow different transformations at different image locations. That is, the latent code z is manipulated by different operation at different locations $\phi_{h,w}$, creating a set of spatially-varying latent codes $\{z_{h,w}\}$, each correspond to different gain and bias parameters $\hat{\gamma}_{h,w}, \hat{\beta}_{h,w} = L(z_{h,w})$ that construct the full denormalization maps $\hat{\gamma}, \hat{\beta}$. The denormalization is then performed according to eq. 1.

Let us explore the relatively simple case of two different manipulations of the latent code $z_1 = \phi_1(z), z_2 = \phi_2(z)$, such that each is applied at different region of the image according to a binary mask m . In this case we will have a set of two corresponding denormalization parameters

$$\begin{aligned} \gamma_1, \beta_1 &= L(z_1), \\ \gamma_2, \beta_2 &= L(z_2). \end{aligned} \quad (2)$$

The final gain and bias maps are then constructed by

$$\begin{aligned} \hat{\gamma} &= m \odot \gamma_1 + (1 - m) \odot \gamma_2, \\ \hat{\beta} &= m \odot \beta_1 + (1 - m) \odot \beta_2, \end{aligned} \quad (3)$$

where \odot denotes element-wise product, as also illustrated in Fig. 2b. This technique therefore enables to control the generation process at different image regions, and thus allows local edit and image manipulations. Our method is similar to the normalization mechanism of SPADE [18] presented in the context of semantic image translation, in the sense that both let the denormalization operation to vary spatially. However note that [18] *train* their generator with the adapted normalization whereas we suggest to adapt the normalization of a pre-trained generator only at inference time. Thus we avoid any additional training which can be very unstable with GANs.

4. Applications

We next demonstrate the use of our Inference-Time Adaptive Normalization (ITAN) technique for four different applications. All are done using pre-trained fixed GAN models, and therefore our run time is equal to the inference time of the model. Please see additional results in the supplementary materials (SM).

4.1. Local generation

We first demonstrate the use of ITAN for randomly drawing only a specific part of the image. For this task we use StyleGAN2 [13] pre-trained on the FFHQ dataset [12]. We start by generating a random image according to a random latent vector z_{init} . In the next step we select an area to be re-sampled according to a new random latent code $z_{\text{re-samp}}$, and construct a corresponding spatial binary mask m that indicates which latent vector controls the generation at what



Figure 3: **Local generation.** Our approach enables to randomly resample only specific regions of an image. In these examples we first generate three different images (leftmost column) using StyleGAN [13] and then randomly re-generate only a specific part of the face indicated by the mask (left column, upper left corner) using our inference-time adaptive normalization. By doing so we are able to create new image content inside the mask (e.g. (a) new eyes appearance, (b) nose shapes, (c) lips and chin composition), while keeping the rest of the image intact.

region. We then use our scheme described in sec. 3 and follow eq. (3), (4) with $z_1 = z_{\text{init}}$, $z_2 = z_{\text{re-samp}}$, m , to construct an image that corresponds to z_{init} outside the mask and $z_{\text{re-samp}}$ inside the mask. Figure 3 shows several results of our local re-generation scheme, each row corresponds to a different initial sample z_{init} and each column represents a new re-sampled image with $z_{\text{re-samp}}$. As can be seen, our approach enables to re-generate only a specific region of the face, while keeping the rest of it identical to the initial sample. Note how although we use a relatively coarse mask, the blending effect is completely smooth. As StyleGAN is constructed with a multi-scale architecture, we are able to choose which scales to modify using ITAN normalization. In these experiments we adapt scales 1-3. The effect of choosing different sets of scales is exemplified in the SM.

4.2. Semantic attribute transfer

Here we spatially compose an image from two different sources. We exemplify this with StyleGAN2 [13]. Given two images $G(z_{\text{source}})$, $G(z_{\text{target}})$ and a binary mask m that indicates how to perform the composition spatially, we gen-

erate an image that corresponds to the attributes encoded in $z_1 = z_{\text{source}}$ outside the mask, and $z_2 = z_{\text{target}}$ inside the mask. Again we achieve this by following eq. (3), (4). This can be seen as a version of 4.1, but instead of randomly drawing $z_{\text{re-samp}}$, we choose a specific latent vector z_{target} which generates an image $G(z_{\text{target}})$ with a specific local attribute we wish to transfer to $G(z_{\text{source}})$. The results are presented in Fig. 4. As can be seen, in contrast to the global style mixing suggested in [12], we are able to transfer only local attributes like lips, eyes and nose appearance. Here as well we modify scales 1-3 of the models with ITAN.

4.3. Class hybridization

Next, we exemplify the use of our method for the task of class hybridization, our goal is to generate images that spatially combine two different classes according to a guidance mask. For this task we use BigGAN [2] which is a class conditioned model that was trained on the ImageNet dataset [6] containing 1K classes. Obviously, conditioning the generation on the image class enables to generate images from only one specific class at a time. We use our ITAN

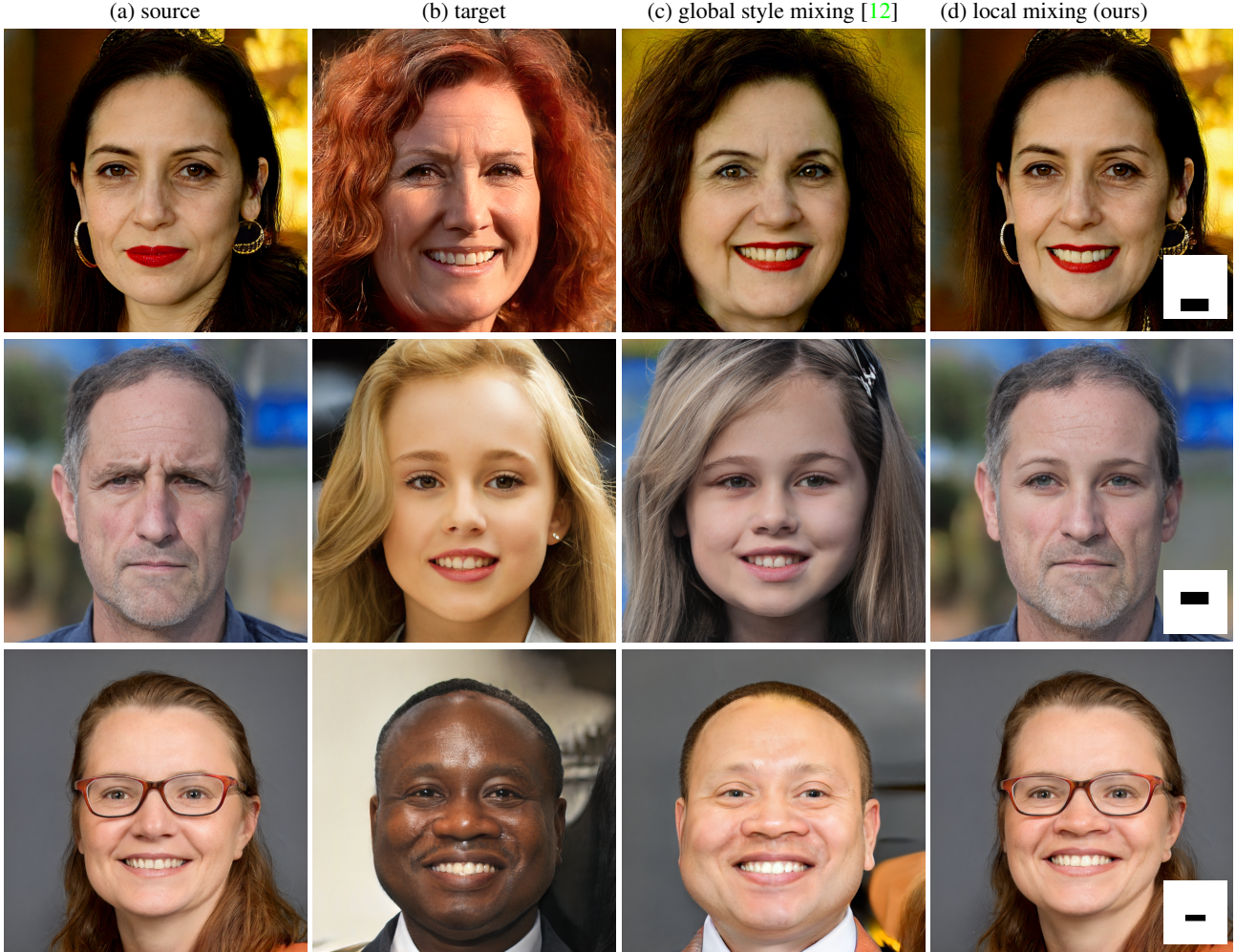


Figure 4: **Semantic attribute transfer.** The multi-scale architecture of StyleGAN [12] enables to perform global style mixing by taking one latent code to globally control a subset of scales, and another latent code to globally control the rest of the scales. Here we use the latent code of the target image (b) for the coarser scales, and that of the source image (a) for finer scales. This result in a mixed image (c) containing the global structure of the target image (*e.g.* face and hair shape) and finer image features from the source image (*e.g.* skin tone). Our test-time adaptive normalization enables spatial control over this effect; we perform the mixing *locally* (d) according to a given spatial mask ((d) right lower corner). The effect is that only local attributes are transferred from the target image to the source image, while keeping the area outside the mask fixed. Note how we manage to transfer relatively coarse structures (*e.g.* lips, eyes and nose shape) while maintaining realistic appearance.

technique to challenge this.

In BigGAN, the class representation is embedded into the generation process as part of the latent code. That is, the latent code is a concatenation of a random noise z and the class representation c such that the input to the model blocks is $[z, c]$. Therefore we are able to use our ITAN mechanism in order to synthesize images that combine two different classes spatially. We start by drawing a random vector z that will be shared across all spatial locations, then we choose the classes to be combined c_1, c_2 according to a binary mask m that represents the spatial location of each. The ITAN

gain and bias maps are then calculated according to eq. 3, 4 where $z_1 = [z, c_1]$ and $z_2 = [z, c_2]$. As mentioned before, we apply the spatial normalization only at test time on the pre-trained BigGAN model, and therefore, there is no need for additional training or fine tuning.

Figure 5 shows several examples of our class hybridization results. As can be seen, by changing only the image class but keeping z fixed, $G(z_1)$ and $G(z_2)$ share the same general layout. By spatially combining the classes using ITAN, the final result is an image that combines both of the classes, and maintains a realistic appearance (though describ-

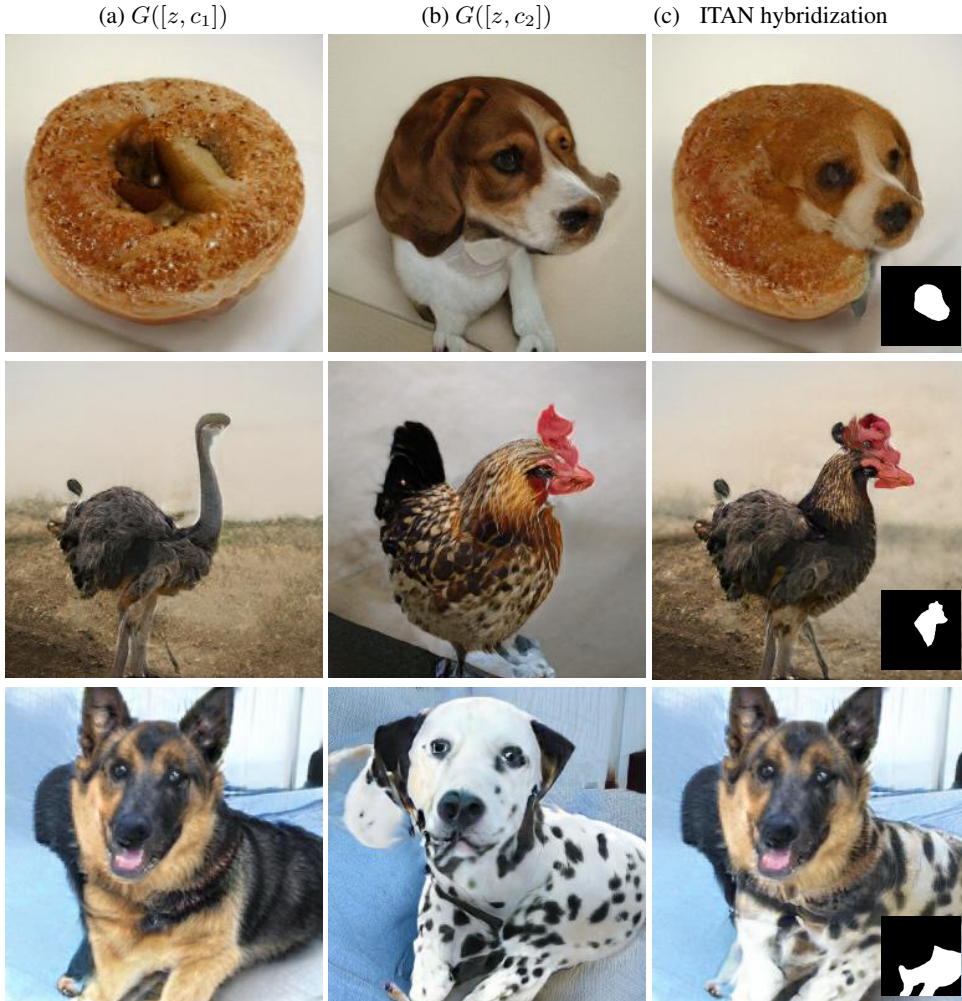


Figure 5: **Class hybridization.** Class conditioned GANs such as BigGAN [2] are trained to generate images that explicitly belong to a specific class ((a),(b)). We use our inference-time adaptive normalization technique to break this, and create images of hybrid classes, blended spatially according to a mask ((c), bottom-right corner). The result is an image that contains an hybridized object of both of the classes (c).

ing an unrealistic object combination). Note that the first BigGAN block has no notion of spatial dimension, therefore for this block only we choose only c_1 which greatly affects the general appearance of the results.

4.4. Saliency manipulation

We next use ITAN not only for combining different latent codes spatially, but also for finding an optimal spatial manipulation of them according to a desired image effect. We choose to exemplify this for the task of saliency manipulation. That is, we would like to edit an image such that a specific region will be more/less salient. Previous works suggest to perform such manipulation directly in the image space by editing pixels/patches [16, 17]. Using ITAN we harness the power of GANs for this task, enabling to change

the saliency of an image by generating a completely new image content. For example, in Fig 6 row (a), in order to make the upper-left corner of the image more salient, our approach manipulates the latent vectors such that the GAN generates a house in the background.

To perform this we use the GANalyze baseline [7], which aims to find meaningful directions in BigGAN’s latent space according to a network that assesses cognitive properties of images (*e.g.* memorability, aesthetics, emotional valence). That is, the optimization process includes finding an optimal transformation of the latent code $\phi(z)$ so that all latent vectors going towards this transformation will get a target score from the assessor A_{target} . Namely $\arg \min_{\phi} (A(G(\phi(z))) - A_{\text{target}}(G(z)))$. See [7] for details.



Figure 6: **Saliency manipulation.** We use our inference-time adaptive normalization to extend GANalyze [7] to control spatial effects. This enables to solve tasks that require treating different image regions differently, like saliency manipulations. Instead of searching for a global optimal latent code transformation, as done in [7], we find a pair of optimal codes each corresponding to a different image region (*e.g.* foreground and background in line (c), left corner and the rest of the image in line (a)). As can be seen, our technique enables to change the saliency of the image such that the indicated area is less/more salient, while keeping the global semantics of the image with minimal changes, compared to the baselines where drastic changes alter the image semantics. For validation, we check the saliency map of the result. Our images are the only to achieve the desired effect.

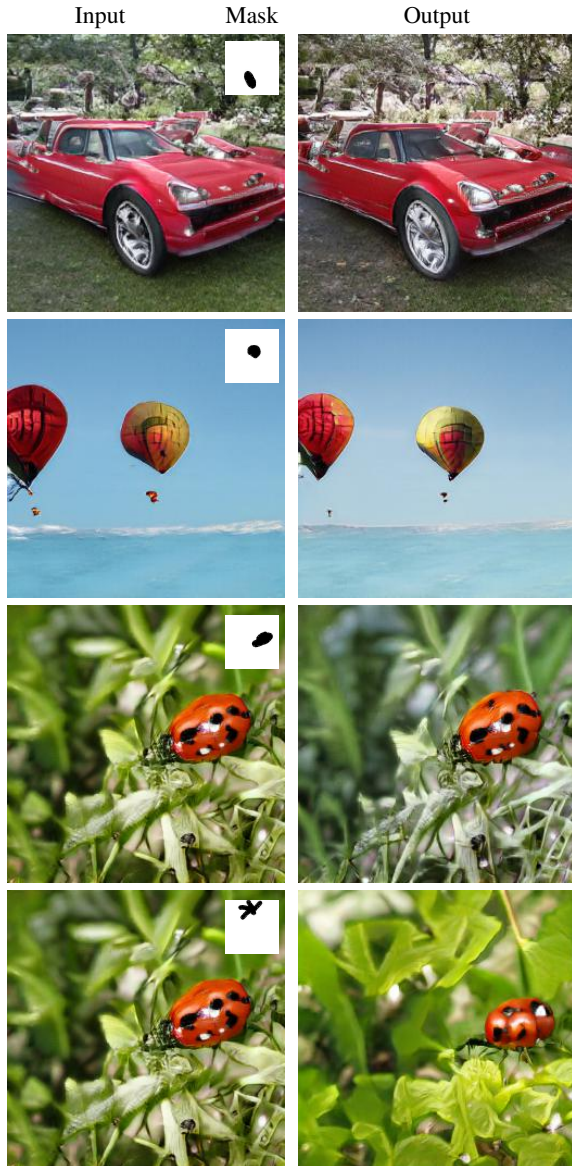


Figure 7: **Saliency manipulation results.** With ITAN we optimize the latent codes corresponding to two image areas indicated by the mask, such that the indicated object will be more salient. The latent code transformation causes different image effects such as color changes, relighting, focus adaptation, etc. both of the object and the background to achieve the desired effects.

In order to use the GANalyze framework for saliency manipulation we first make two important adaptations to the optimization process: (i) GANalyze works on a dataset of images to find a transformation ϕ that will be optimal for general images. As saliency maps are unique for each specific image, we aim to find a transformation that will be optimal only for the specific image we wish to manipulate.

Therefore we reduce the optimization process to work on a single latent code. (ii) GANalyze assessors output a scalar score. We use the saliency detection network of [5] as an assessor, which takes an image and outputs a saliency *map* that has spatial dimensions. Therefore we modify the loss function to take the ℓ_2 norm between the measured and target saliency maps (instead of a simple difference). Namely $\arg \min_{\phi} \| (A(G(\phi(z))) - A_{\text{target}}(G(z))) \|^2$.

Next, we incorporate the ITAN normalization to the BigGAN model (as described in sec. 3) to allow spatial manipulation at test time. Our optimization scheme includes finding *two* optimal transformation $z_1 = \phi_1(z)$ and $z_2 = \phi_2(z)$ such that the combination of two according to a spatial map with the ITAN normalization scheme will give an image with the desired target saliency map. As in GANalyze, we take ϕ to be a simple parametric affine transformation of the latent code. Note that in contrast to previous applications, here both the regions inside and outside the mask can change.

We compare our result with the GANalyze baseline [7] using [5] as an assessor, and with a variant of GANalyze that finds the optimal transformation only for a single image. Both of these find a global transformation ϕ that affects the whole image, whereas our approach allows the transformation to be different at different image regions. The results are shown in Fig. 6. As can be seen, both of the baselines cause relatively drastic effects that completely alter the image, whereas our approach results in more delicate manipulations that preserve the general semantics of the image. Investigating the saliency map of the manipulated images (generated with [5]) shows that our approach is the only one to achieve the desired effect; in the first example (row (a)) the latent code manipulation generates a house in the top left corner of the image, which is indeed detected to be more salient (row (b)). In the second example (row (c)) the image background becomes more vivid, and the saliency map (row (d)) indicates that relative to the background, the mushroom is now less salient. Additional examples for our saliency manipulation results appear in Fig. 7. As can be seen, the latent code manipulation with ITAN makes the indicated object more salient by causing different effects in the image space such as relighting, recoloring, modifying the object size and location, and focus changes.

5. Conclusions

We introduce a new normalization technique that is applied at test time to SoTA GANs and enables local control over the generation process. The new approach is useful for a line of tasks. Our examples include manipulation of two image regions, however these can be easily extended to the general case of N regions. In addition, in order to manipulate real images (and not only generated ones) one can first use back-project to the GAN’s latent space.

References

- [1] David Bau, Alex Andonian, Audrey Cui, YeonHwan Park, Ali Jahanian, Aude Oliva, and Antonio Torralba. Paint by word. *arXiv preprint arXiv:2103.10951*, 2021. 2
- [2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 1, 2, 4, 6
- [3] Lucy Chai, Jonas Wulff, and Phillip Isola. Using latent space regression to analyze and leverage compositionality in gans. *arXiv preprint arXiv:2103.10426*, 2021. 2
- [4] Lucy Chai, Jun-Yan Zhu, Eli Shechtman, Phillip Isola, and Richard Zhang. Ensembling with deep generative views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14997–15007, 2021. 2
- [5] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. A deep multi-level network for saliency prediction. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 3488–3493. IEEE, 2016. 8
- [6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 4
- [7] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5744–5753, 2019. 1, 2, 6, 7, 8
- [8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 1
- [9] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *arXiv preprint arXiv:2004.02546*, 2020. 1, 2
- [10] Ali Jahanian, Lucy Chai, and Phillip Isola. On the ”steerability” of generative adversarial networks. In *International Conference on Learning Representations*, 2020. 1, 2
- [11] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 1
- [12] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 1, 2, 3, 4, 5
- [13] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 1, 2, 3, 4
- [14] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2
- [15] Junjian Lian and Jinrong Cui. Anime style transfer with spatially-adaptive normalization. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021. 2
- [16] Roey Mechrez, Eli Shechtman, and Lihi Zelnik-Manor. Saliency driven image manipulation. *Machine Vision and Applications*, 30(2):189–202, 2019. 6
- [17] Youssef A Mejjati, Celso F Gomez, Kwang In Kim, Eli Shechtman, and Zoya Bylinskii. Look here! a parametric learning based approach to redirect visual attention. In *European Conference on Computer Vision*, pages 343–361. Springer, 2020. 6
- [18] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019. 2, 3
- [19] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. SinGAN: Learning a generative model from a single natural image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4570–4580, 2019. 1
- [20] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9243–9252, 2020. 2
- [21] Nurit Spingarn-Eliezer, Ron Banner, and Tomer Michaeli. Gan” steerability” without optimization. *arXiv preprint arXiv:2012.05328*, 2020. 1, 2
- [22] Zhentao Tan, Dongdong Chen, Qi Chu, Menglei Chai, Jing Liao, Mingming He, Lu Yuan, Gang Hua, and Nenghai Yu. Efficient semantic image synthesis via class-adaptive normalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2
- [23] Zhentao Tan, Dongdong Chen, Qi Chu, Menglei Chai, Jing Liao, Mingming He, Lu Yuan, and Nenghai Yu. Rethinking spatially-adaptive normalization. *arXiv preprint arXiv:2004.02867*, 2020. 2
- [24] Jiashun Wang, Chao Wen, Yanwei Fu, Haitao Lin, Tianyun Zou, Xiangyang Xue, and Yinda Zhang. Neural pose transfer by spatially adaptive instance normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5831–5839, 2020. 2
- [25] Jongsu Yoon, Taehyeon Kim, and Yoonsik Choe. Gan based single image super-resolution via spatially adaptive de-normalization. *Transactions of the Korean Institute of Electrical Engineers*, 70(2):402–407, 2021. 2
- [26] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5104–5113, 2020. 2
- [27] Alara Zindancıoğlu and T Metin Sezgin. Perceptually validated precise local editing for facial action units with stylegan. *arXiv preprint arXiv:2107.12143*, 2021. 2