Supplementary of

LT-GAN: Self-Supervised GAN with Latent Transformation Detection

Parth Patel[2*†] Nupur Kumari[*1] Mayank Singh[*1] Balaji Krishnamurthy[1]
{parpatel,nupkumar,msingh,kbalaji}@adobe.com
1. Media and Data Science Research Lab, Adobe
2. Birla Institute of Technology & Science, Pilani India

1. Additional Results and Ablation Studies

Variance analysis of FID We show the FID [2] variance box-plot of our approach LT-GAN on SNDCGAN [4] and BigGAN [1] architectures for CIFAR-10 and CelebA-HQ datasets in Fig 1. To provide a fair evaluation of our approach using FID, for each configuration, we compute the FID 3 times with different random initial seeds.

Inception Score We evaluate our approach LT-GAN using another GAN evaluation metric named Inception Score (IS) [6]. Here, we report the IS of models trained on CIFAR-10 dataset. As shown in Table 1, LT-GAN improves IS over baseline, while CR+LT-GAN approach achieves the best IS results, on both SNDCGAN and BigGAN architectures.

	SNDCGAN	BigGAN
Baseline	7.54	8.79
LT-GAN	**7.85**	**9.13**
CR-GAN	7.93	9.17
CR+LT-GAN	**8.16**	**9.17**

Table 1: Inception Score for SNDCGAN and BigGAN architectures trained using different approaches on CIFAR-10.

Choice of Architecture of Auxiliary Network A On SNDCGAN CelebA-HQ setting using optimal value of $\sigma_\epsilon = 0.5$, we experimented with different architectures for the auxiliary network A:

- Linear Network: A linear network (with a single fully-connected layer) - its capacity was not sufficient to distinguish between generative transformations resulting from different ϵ's and hence, the auxiliary task training failed.
- Non-linear Network: A non-linear network (with two fully-connected layers and ReLU activation at the hidden layer) achieved a FID score of 19.63.
- Convolutional Network: A convolutional network (with a convolutional layer, a batch normalisation layer and a fully-connected layer) achieved a FID score of 21.27.

2. Qualitative Analysis of Generated Images

2.1. LT-BigGAN [ImageNet]

Steerability of latent space We show more qualitative samples of varying the *zoom*, *brightness*, *vertical position* and *horizontal position* in generated images of classes same

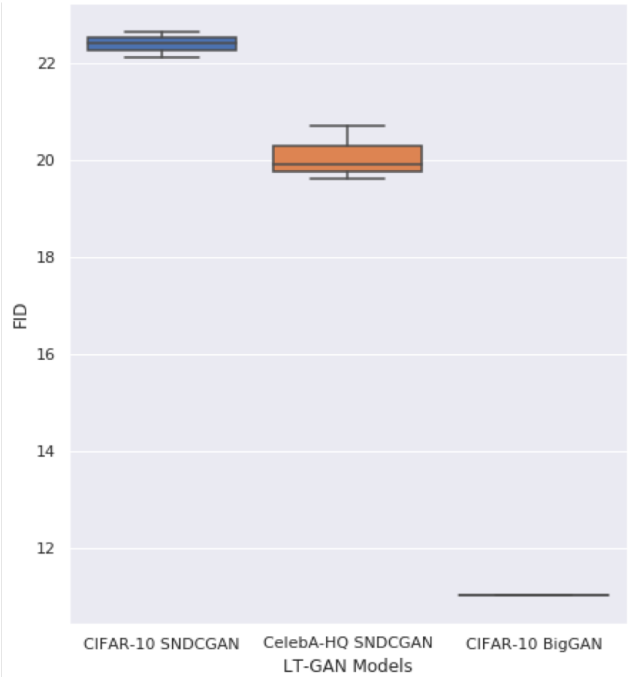


Figure 1: FID Variance box plot of LT-GAN approach on different architectures for CIFAR-10 and CelebA-HQ.

*Authors contributed equally
†Work done during Adobe MDSR internship

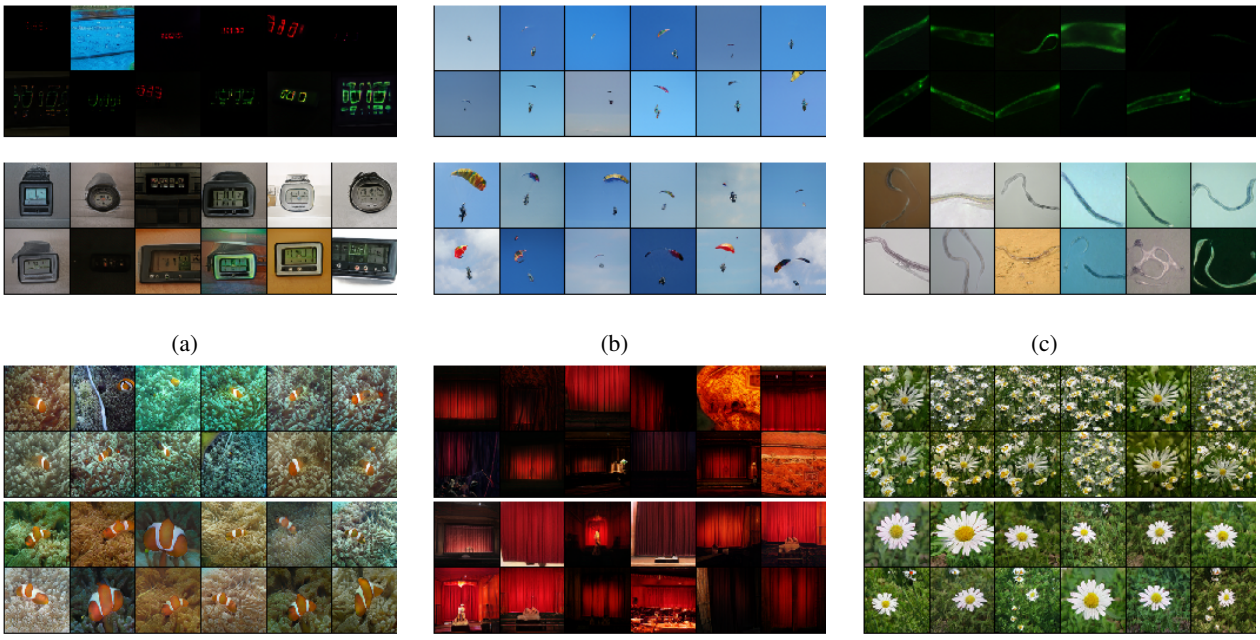


Figure 2: Samples of generated images from categories with mode collapse in Baseline BigGAN and its corresponding images generated from LT-BigGAN model. The 6 blocks of images corresponds to ImageNet classes: (a) *digital clock*, (b) *parachute*, (c) *nematode*, (d) *anemone fish* ,(e) *theater curtain* and (f) *daisy*. In each block (that comprises of 4 rows of images), the top part (1st and 2nd row) corresponds to images generated using Baseline (BigGAN) model and the bottom part (3rd and 4th rows) corresponds to images generated using our approach LT-BigGAN.

as [5], through latent space manipulation as discussed in Section 4.3 of the paper. Fig. 3 shows sample images generated from LT-BigGAN and baseline BigGAN model on perturbing the latent code in the positive and negative direction of brightness and zoom vector. Similarly, Fig. 4 shows latent space steerability for horizontal and vertical shift. We can observe that the baseline model generates distorted images at the extremes and fails to control brightness factor and semantic content for all categories. In contrast, LT-BigGAN generates smooth variations of images while preserving the content and is able to generalize the brightness even for categories that usually are not available in a dark environment e.g *cheeseburger* class.

Mode Collapse In the conditional image generation setting on ImageNet dataset using our proposed self-supervision approach, we observe that it not only improves the FID score but also helps in alleviating the issue of mode collapse. In Fig. 2, we show example images of classes which suffer from mode collapse in baseline BigGAN model trained on ImageNet and its corresponding samples generated from LT-BigGAN. We can see that images generated from LT-BigGAN are more diverse as compared to the baseline model.

2.2. Image Editing on LT-StyleGAN [CelebA-HQ]

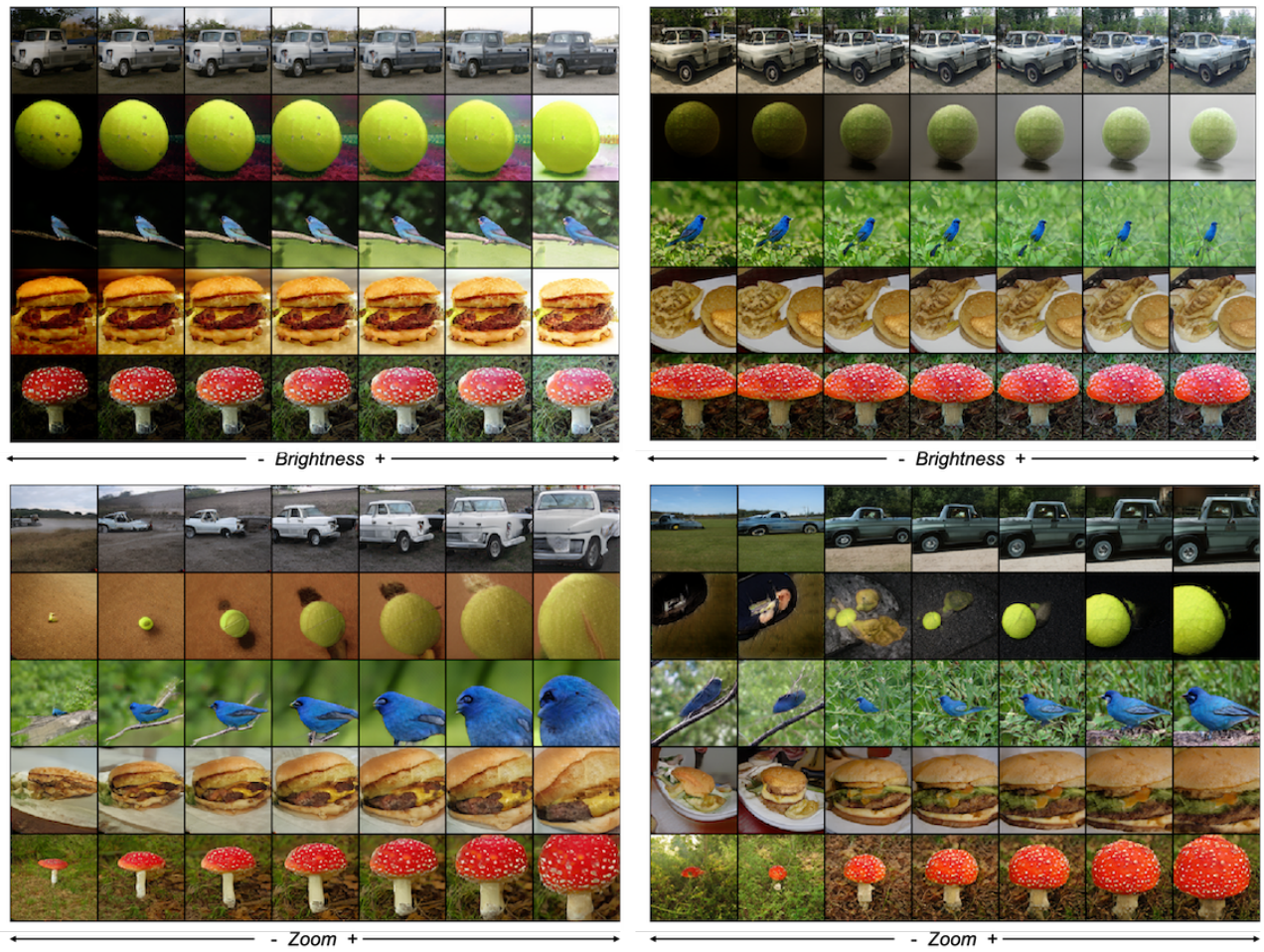
In Fig. 5, we show more examples of the manipulation of facial attributes namely age, gender, smile expression and eyeglasses by using the InterfaceGAN framework [7] on LT-StyleGAN model.

3. Hyper-parameter Details

This section mentions the choice of hyper-parameters for training LT-GAN over different datasets and architectures. For experiments in CR-GAN, as mentioned in [8], the augmentation used for consistency regularization is a combination of randomly shifting the image by a few pixels and random horiozontal flipping. The shift size is 4 pixels for both CIFAR-10 and CelebA-HQ datasets, and rest all hyper-parameters remain same as baseline. For our LT-GAN approach, we used twice the batch size for G and kept the batch size of D same as that of the baseline, because this modification achieved better results. Note that for fair comparison, we also tried doubling the batch size of G for baseline models, however the FID performance deteriorated.

3.1. SNDCGAN CIFAR-10

Following the hyper-parameter choices of [8], we use $d_{step} = 1$ and set the dimensionality of the latent space to be 128. Adam optimizer with $\alpha = 0.0002$,



(a) LT-BigGAN (b) BigGAN

Figure 3: Qualitative comparison for varying brightness and zoom between LT-BigGAN (left) and Baseline BigGAN (right) in five categories of ImageNet through latent space manipulation method of [5]

$\beta_1 = 0.5$ and $\beta_2 = 0.999$ is used for both G and D. We use a batch size of 64 for both G and D for baseline models.

LT-GAN: σ_ϵ is chosen to be 0.6 and Adam optimizer with default values of $\alpha = 0.001$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ is chosen for the auxiliary network A. λ is set to 1.0. The encoder features $E(G(z))$ corresponding to generated images $G(z)$ are taken from the fifth layer of the discriminator [1]. The features $E(G(z))$ are passed through an average-pool 2D layer with kernel size 2, stride 2 and zero padding, and then flattened before being passed to the auxiliary network. The number of warmup iterations n before introducing the self supervision task is 2000.

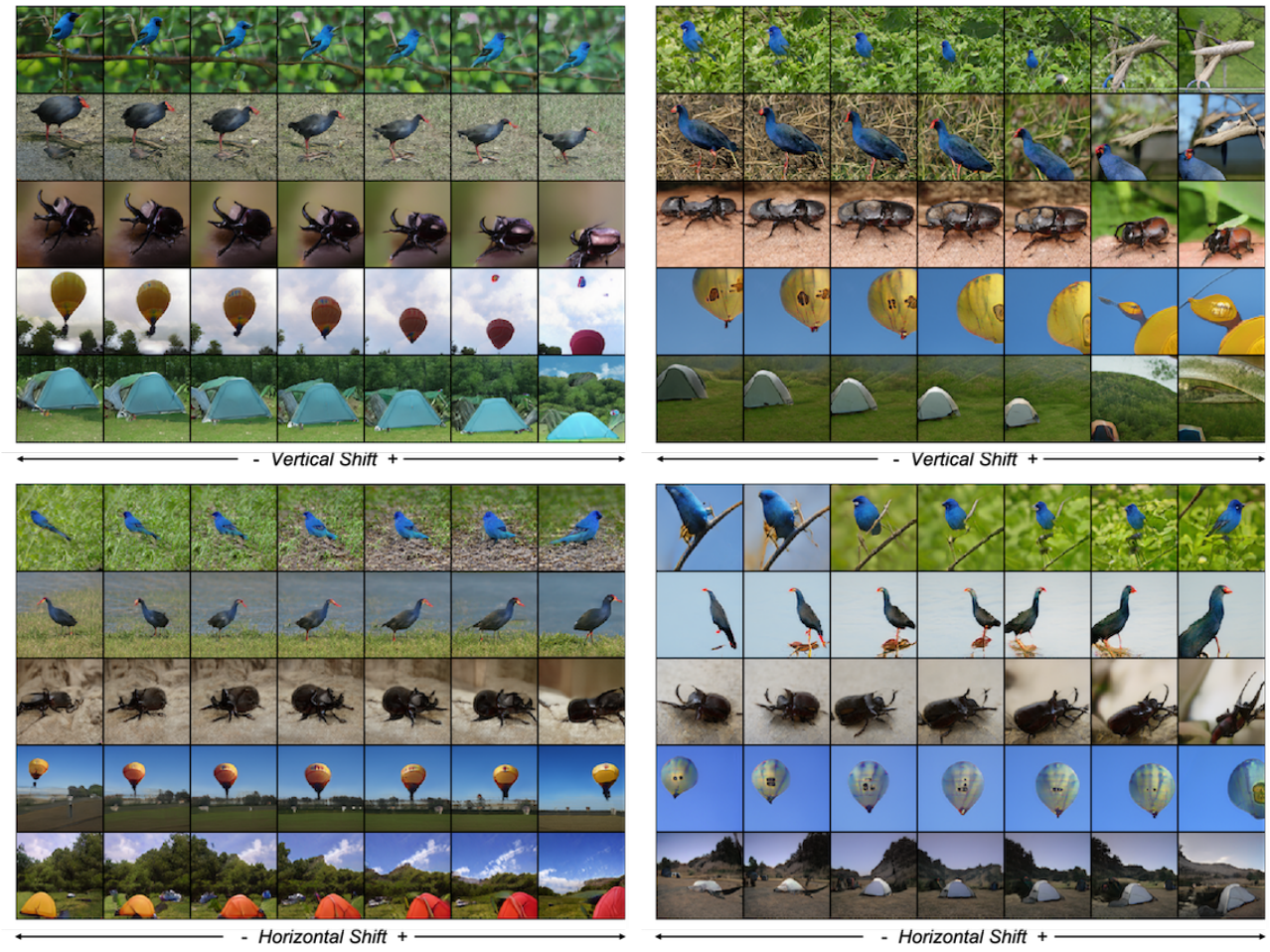
[1]SNDCGAN discriminator consists of 7 convolutional layers followed by a linear layer at the end. Each convolutional layer is followed by a ReLU activation. We treat each convolutional layer followed by its ReLU activation as a single layer. Thus, SNDCGAN discriminator consists of 8 layers.

CR+LT-GAN: σ_ϵ is chosen to be 0.55. Rest all hyperparameters are same as those mentioned for LT-GAN.

3.2. BigGAN CIFAR-10

We use standard value [1] of $d_{steps} = 4$, dimensionality of z as 128 and batch size as 64. Adam optimizer ($\alpha = 0.0002$, $\beta_1 = 0.0$ and $\beta_2 = 0.999$) is used for G & D.

LT-GAN: σ_ϵ is chosen to be 0.6. Adam optimizer with default values of $\alpha = 0.001$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ is chosen for the auxiliary network A. λ is set to 1.0. The encoder features $E(G(z))$ corresponding to generated images $G(z)$ are taken from the last layer of the discriminator just before sum pooling. The number of warmup iterations n before introducing the self supervision task is 2000.



(a) LT-BigGAN (b) BigGAN

Figure 4: Qualitative comparison for geometric transformation (horizontal and vertical shift) between LT-BigGAN (left) and Baseline BigGAN (right) in five categories of ImageNet through latent space manipulation method of [5]

CR+LT-GAN: All hyper-parameters are same as that of LT-GAN with default CR-GAN configuration.

3.3. SNDCGAN CelebA-HQ

Following the hyper-parameter choices of [8], we use $d_{steps} = 1$ and set the dimensionality of the latent space to be 128. Adam optimizer with $\alpha = 0.0002$, $\beta_1 = 0.5$ and $\beta_2 = 0.999$ is used for both G and D. We use a batch size of 64 for both G and D for baseline model.

LT-GAN: σ_ϵ is chosen to be 0.5. Adam optimizer with default values of $\alpha = 0.001$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ is chosen for the auxiliary network A. λ is set to 1.0. The encoder features $E(G(z))$ corresponding to generated images $G(z)$ are taken from the seventh layer of the discriminator. The features $E(G(z))$ are passed through an average pool 2D layer with kernel size 4, stride 4 and zero padding, and then flattened before being passed to the auxiliary network. The number of warmup iterations n before introducing the self supervision task is chosen to be 1500.

CR+LT-GAN: Number of warmup iterations n is set to be 5000. Rest all hyper-parameters are same as those mentioned for LT-GAN.

3.4. StyleGAN CelebA-HQ

StyleGAN adopts progressive growing of both the generator and the discriminator networks. In LT-StyleGAN, we introduce the self supervision task after the layer corresponding to 128 resolution has completely faded into the network architecture. The per-pixel noise added after after each convolution block in generator is kept same while generating images corresponding to latent codes z and $z + \epsilon$. For incorporating mixing regularization in LT-StyleGAN, the GAN-induced transformation of $G(z_1, z_2)$ is generated as $G(z_1 + \epsilon_1, z_2 + \epsilon_2)$, where ϵ_1 and $+\epsilon_2$ are distinct.

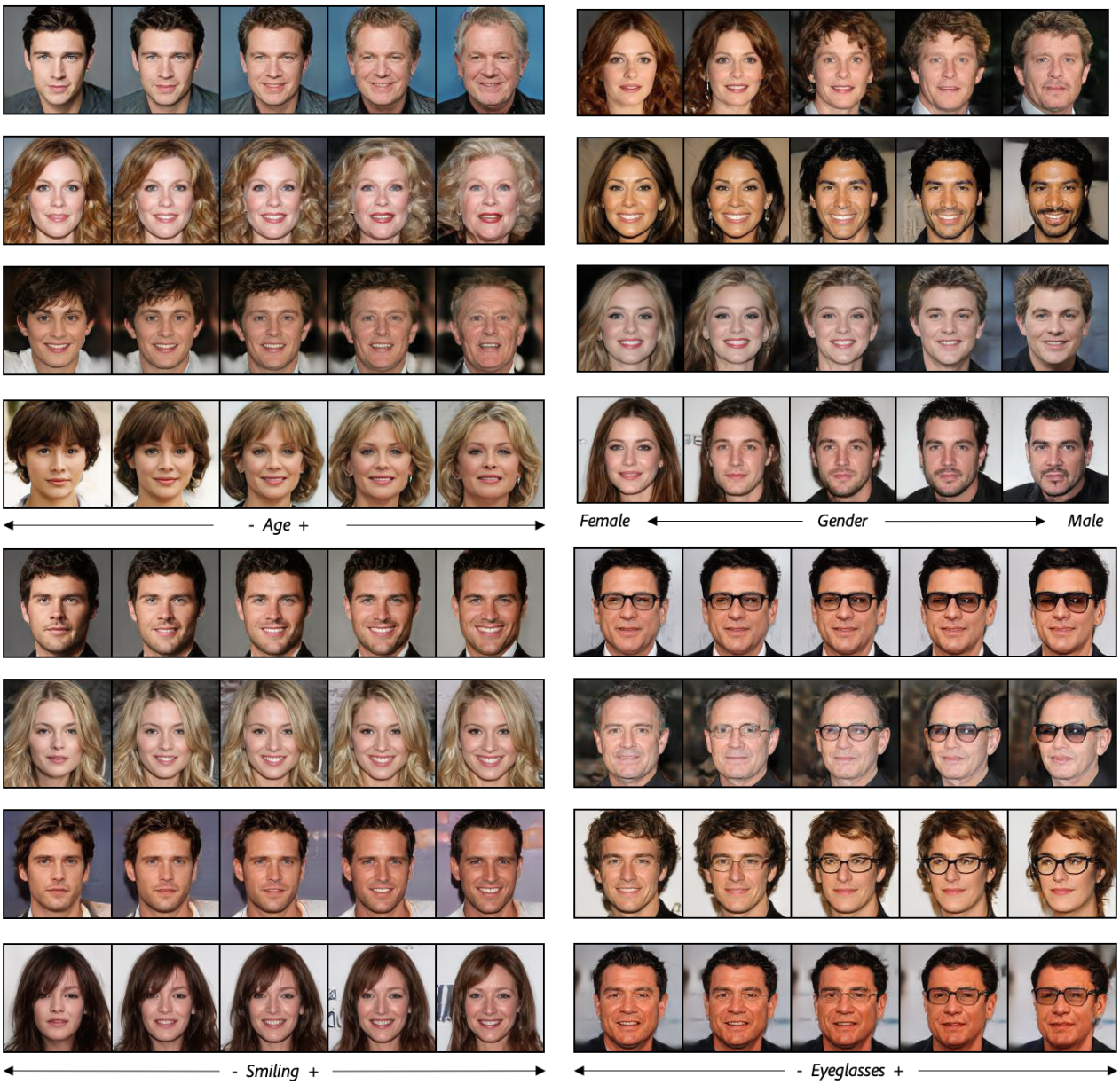


Figure 5: Manipulation of Age (top left), Gender (top right), Smile (bottom left)and Eyeglasses (bottom right) attributes by navigating the latent space of LT-StyleGAN using InterfaceGAN [7] framework. Original images are in the centre and the left and right images are generated by moving the latent code in negative and positive directions respectively.

Following the hyper-parameter choices of [3], we set the dimensionality of both the latent spaces Z and W to be 512. The mapping network from Z to W is a 8 layer MLP. While training using progressive growing, we start from 8×8 resolution, fade in a new layer during the next 600K images and then let the network stabilize for next 600K images before introducing a new layer. For 128×128 resolution, we use Adam optimizer with $\alpha = 0.0015$, $\beta_1 = 0.0$ and $\beta_2 = 0.99$ for both G's synthesis network and D. We reduce the learning rate by two orders of magnitude for G's mapping network (i.e. a learning rate of $\alpha = 0.000015$), as specified in [3]. We use $d_{steps} = 1$. We use a batch size of 32 for both G and D with mixing probability set to 0.9 for baseline model.

LT-GAN: We choose σ_ϵ to be 0.5, with a mixing probability of 0.5. Adam optimizer with default values of $\alpha = 0.001$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ is used for the auxiliary network A. We use λ value of 0.5. We take the encoder features $E(G(z))$ corresponding to generated images $G(z)$ from the layer of the discriminator corresponding to 16×16

resolution. The features $E(G(z))$ pass through an average pool 2D layer with kernel size 2, stride 2 and zero padding, and then we flatten it before passing it to the auxiliary network.

3.5. BigGAN ImageNet

We use the default configuration[4] of $d_{step} = 1$, dimensionality of z as 120, batch size of $8 * 256$. We select Adam optimizer with $\alpha = 0.0001$ and 0.0004 for G for D respectively.

LT-GAN: We choose σ_ϵ to be 0.5. We set Adam optimizer with default values of $\alpha = 0.001$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for the auxiliary network A. λ is set to 0.5. We take the encoder features $E(G(z))$ corresponding to generated images $G(z)$ from the seventh layer of the discriminator. The features $E(G(z))$ pass through an average pool 2D layer with kernel size 2, stride 2 and zero padding, and then we flatten before passing to the auxiliary network. The number of warmup iterations n before introducing the self supervision task is set to $100K$ [2].

References

[1] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019.

[2] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

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

[4] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018a.

[5] Antoine Plumerault, Hervé Le Borgne, and Céline Hudelot. Controlling generative models with continuous factors of variations. In *ICLR*, 2020.

[6] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, 2016.

[7] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, 2020.

[8] Han Zhang, Zizhao Zhang, Augustus Odena, and Honglak Lee. Consistency regularization for generative adversarial networks. In *ICLR*, 2020.

[2] We use the pretrained model of `https://github.com/ajbrock/BigGAN-PyTorch` as baseline and use the provided checkpoint at $100K$ for fine-tuning with LT-GAN