Supplementary material for: GAN Cocktail - mixing GANs without dataset access

Omri Avrahami[1], Dani Lischinski[1], and Ohad Fried[2]

[1]The Hebrew University of Jerusalem [2]Reichman University

1 Implementation Details

We evaluated our technique and the baselines using the StyleGAN2 architecture [3]. We kept most of the details unchanged, including network architecture, weight demodulation, regularization, exponential moving average of generator weights, R1 regularization [6], mini-batch size of 32 images, and using the Adam optimizer [4] with $\beta_1 = 0, \beta_2 = 0.99$ and $\epsilon = 10^{-8}$.

In order to introduce conditioning to the unconditioned StyleGAN2 architecture we add the following components to the generator and the discriminator:

- **Generator conditioning.** We add a class embedding layer to the mapping network of the generator, s.t. the input to the generator is noise vector z and one-hot class c. The embedded condition is concatenated to the input z. The first fully-connected layer of the mapping network is modified to support this new size.
- **Discriminator conditioning.** We add a mapping network to the discriminator that gets as an input only a class one-hot vector c (with no noise vector z) and calculates a w vector. We then incorporate this w vector to the final discriminator prediction by a projection [8].

If our input models are unconditioned or conditioned with an insufficient number of classes, we can easily introduce/extend the class embedding layer to the input models to the desired size by adding more rows to the embedding matrix and initialize it randomly.

It is important to notice that we do not rely on any of StyleGAN's features in our solution (or in the baselines), so our solution is agnostic to the input GAN architecture.

1.1 Hyperparameters and training configurations

We used the same hyperparameter configurations as in the PyTorch [10] implementation of StyleGAN2-ada [1], while we did not use the adaptive augmentation capabilities. We used a fixed mapping depth of 8 layers during all our experiments. The hyperparameters were chosen by a random search and are available at the source code.

We used a single NVIDIA RTX 2080 GPU per experiment. We incorporated mixed-precision training [7] in order to speed up the training process.

Table 1: Comparison of FID score w.r.t. the union of all the datasets, for several dataset combinations. Cat, horse, church and bedroom datasets are taken from LSUN [12]

Datasets	cat horse	cat bedroom	cat church	FFHQ horse	FFHQ bedroom	FFHQ church	horse church	horse bedroom	church bedroom
From scratch	20.53	22.81	21.01	13.74	14.96	11.83	12.85	16.96	13.33
TransferGAN [11]	16.73	18.7	17.22	14.4	13.88	11.19	12.27	15.22	11.84
EWC [5]	17.46	17.98	16.75	13.27	14.25	11.11	14.51	15.43	11.07
Freeze-D [9]	16.98	18.53	18.04	13.25	13.14	9.74	10.78	15.81	11.81
Our	**16.46**	**16.7**	**15.62**	**11.28**	**11.5**	**9.52**	**10.61**	**13.9**	**9.91**
Upper bound (real data training)	13.17	14.42	13.65	8.52	8.59	6.25	8.65	10.26	7.01

We trained all our experiments until convergence, which takes about 5M training steps because we start from pre-trained models. Each stage of our two-stage approach (model rooting and merging) takes about 2 days on NVIDIA RTX 2080, thus the total training time is about 4 days.

2 Additional experiments

In addition to the experiments reported in the main paper we also compared our method on other datasets. Furthermore, we experimented with mixing models of different architectures and mixing models of different quality.

2.1 Additional datasets

We compared our method using additional classes from the LSUN dataset. As can be seen in Table 1, our method outperforms the baselines in all of our experiments. In addition, we tested the effect of using multiple source datasets, as reported in Table 2. As we can see, our method outperforms the baselines even when mixing seven different models.

2.2 Mixing models with different architectures

For most of our experiments we use the StyleGAN2 framework. However, our method can be used to merge models with different architectures. In the first stage (model rooting) after we choose the root model, the remaining models serve only as data-generators, hence can be of any architecture. In the second stage (model merging), all the rooted models that we create are, by design, of the same architecture as the root.

We evaluated our solution (and the baselines) on merging models with different architectures: a StyleGAN2 model trained on LSUN cat and a custom made model that was trained on LSUN dog. The custom made model was created by removing the mapping network from the StyleGAN architecture and replace it

Table 2: Comparison of FID score w.r.t. the union of all the datasets, for several dataset combinations. Cat, horse, church, car and bedroom datasets are taken from LSUN [12].

Datasets	FFHQ cat	FFHQ cat dog	FFHQ cat dog car	FFHQ cat dog car horse	FFHQ cat dog car horse bedroom	FFHQ cat dog car horse bedroom church
From scratch	19.61	23.22	24.88	20.56	18.36	18.34
TransferGAN [11]	18.63	20.64	19.34	18.4	17.49	17.29
EWC [5]	19.45	19.47	19.14	18.14	18.56	17.18
Freeze-D [9]	18.17	19.71	19.41	17.8	17.24	17.5
Our	**16.44**	**18.98**	**18.44**	**17.35**	**17.04**	**16.41**
Upper bound (real data training)	11.86	15.93	16.45	16.19	16.88	16.33

with a simple linear embedding layer. Each of the models was chosen as root, thus in one case the merged model is a StyleGAN2, and in the other case, the merged model is a custom one. As shown in Table 3, our mixing approach outperforms the baselines, in terms of FID score, regardless of the architecture of the root model. Again, it does not mean that the root model is meaningless: choosing the StyleGAN2 architecture for the merged model produces superior results, compared to merging that uses the custom architecture.

We have noticed that both of the source models have comparable FID scores, which leads us to the next question: what happens if we mix models of different FID scores.

2.3 Mixing models of different quality

In order to isolate and identify changes that result in consistent improvements across our various experiments, we mainly focus on comparing models of the same quality: models of the same capacity that were trained on roughly the same dataset size. This raises the question of whether our method is beneficial in the cases where the models are of different quality.

To test under such conditions, we trained a StyleGAN2 model on a reduced version of LSUN dog with an order of magnitude fewer training samples: we evaluated the mixing between a model that was trained on 100K samples of LSUN cat and a model that was trained on 10K samples of LSUN dog. Table 4 demonstrates that in this scenario, our method still outperforms the baselines regardless of the choice of the root model. It is also important to notice that EWC performs significantly worse when the root model is the one that was

Table 3: **Mixing models of different architectures.** Our method outperforms the baselines, in terms of FID score, regardless of the architecture of the chosen root model

	LSUN cat + LSUN dog	
Root model	LSUN cat (StyleGAN)	LSUN dog (Custom)
Scratch	23.34	23.34
TransferGAN [11]	19.76	22.39
EWC [5]	21.57	21.79
Freeze-D [9]	19.70	21.44
Our	**19.42**	**21.28**

Table 4: **Mixing models of different dataset sizes.** Our method outperforms the baselines, in terms of FID score, regardless of the model that is chosen as a root model. In addition, we can see that EWC performs poorly when initialized with the weaker model

	LSUN cat + LSUN dog	
Root model **# Training samples**	LSUN cat (100K)	LSUN dog (10K)
Scratch	37.46	37.46
TransferGAN [11]	32.52	34.93
EWC [5]	32.18	45.10
Freeze-D [9]	32.03	35.30
Our	**31.71**	**32.59**

trained on the smaller dataset, because the inductive bias towards the weights of this weaker model is a bad prior. The other baselines, as well as our method, are much less sensitive. Nevertheless, we can see that choosing the root model to be the model that was trained on the larger dataset yields better results.

3 Datasets

We used FFHQ [2] and LSUN [12] datasets for our experiments. We used the entire FFHQ dataset which contains 70K images that are automatically aligned and cropped.

Images in the FFHQ dataset are licensed under either Creative Commons BY 2.0, Creative Commons BY-NC 2.0, Public Domain Mark 1.0, Public Domain CC0 1.0, or U.S. Government Works license. All of these licenses allow free use, redistribution, and adaptation for non-commercial purposes.

Table 5: FID comparison of merging LSUN cat + LSUN dog when training on a higher resolutions

Datasets resolution	128×128	256×256
From scratch	32.58	28.61
TransferGAN [11]	26.90	23.28
EWC [5]	28.26	27.18
Freeze-D [9]	29.00	23.65
Our	**25.12**	**22.45**
Upper bound (real data training)	18.82	18.88

The LSUN dataset contains around one million labeled images for each of 10 scene categories and 20 object categories. We used only some of the categories in the dataset (cat, dog, and car) and used only 100K images per class (in order to keep the balance between the FFHQ and the LSUN classes).

We trained the models once and then used the output of the trained models for our experiments. The dataset was used during our experiments only for calculation of the FID metrics. Note that we did not change the behavior of the training process based on the FID score in any way, because we assume that our method should be applicable without any training data. The multivariate Gaussian statistics of the inception features may not be available during the training for the end-user, hence we cannot use it.

3.1 FID calculations

The results in the tables in the main paper are calculated over images of size 64×64, for efficiency reasons. To make sure that the same trends hold for higher resolutions, we tested our method on images of sizes 128×128 and 256×256 on the LSUN cat and LSUN dog datasets and achieved similar results, as can be seen in Table 5. Each stage of our two-stage approach (model rooting and merging) takes about 4 days on NVIDIA RTX 2080 for resolution 128×128, and about 7 days on NVIDIA A10 for resolution 256×256; thus, the total training time is about 8 days and 14 days, respectively.

4 Training

In Figure 1 we can see the convergence rate of the FID that is calculated on the union of the input datasets LSUN cat and LSUN dog (which are semantically close datasets) during the training process. As we can see, our solution converges more quickly and to a lower FID than the baselines.

In Figure 2 we show the FID score that is calculated per class. As we can see, TransferGAN is suffering from catastrophic forgetting on the cat class (left) that

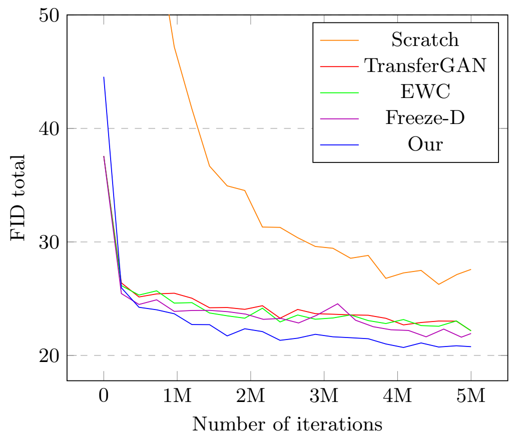


Fig. 1: Convergence rate of the total FID score during the training on LSUN cat + LSUN dog. We can see that our solution achieves the lowest (best) FID score.

is somewhat mitigated by the EWC but it comes at the expense of increasing the FID score of the dog class (right). In contrast, our method starts from a higher FID score on the cat dataset than TrasferGAN/EWC/Freeze-D methods (because they started from the pretrained cat model which achieves better results), but later on, our method achieves a better result.

In Figure 3 we can see the convergence rate of the total FID score when merging two semantically distant datasets: FFHQ and LSUN cat. We can see that our method converges more quickly and to a lower FID than the baselines. As we can see in Figure 4 again, EWC mitigates the catastrophic forgetting of TransferGAN on the FFHQ class and even achieves a better result on this class than our method. But it achieves the worst result on the second class (even worse than the naïve from scratch approach). So all-in-all it is outperformed by our method as can be seen in Figure 3.

5 Applications

Semantic editing We demonstrate additional examples for the semantic editing application. We used a merged model of FFHQ and LSUN cat. In Figure 5 we show more examples to Figure 4 in the main paper: we calculated the human pose direction in the $\mathcal{W}$ latent space of the merged generator on images of FFHQ class only by fitting an SVM that separates images with “positive” pose and images with “negative” pose, then we used the calculated hyperplane normal and applied it to images that were generated from both of the classes. Note how the pose direction also applies to the cats, even though it was calculated using human photos.

In addition, we also experimented with semantic directions whose meaning may be less clear or even undefined for some of the classes. We did not ex-

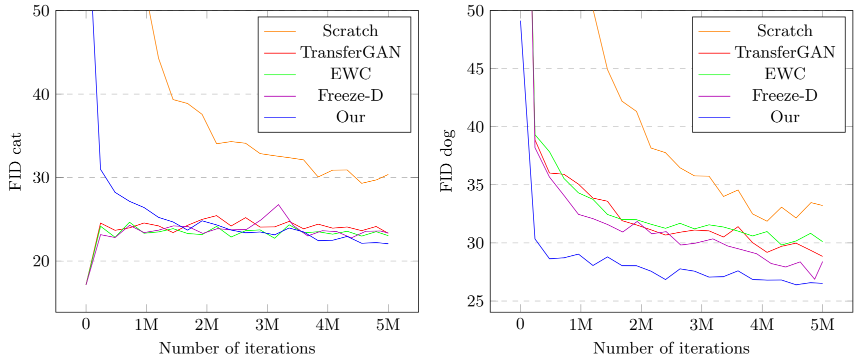


Fig. 2: The FID score that is calculated on the cat class (left) and on the dog class (right). As we can see, the TransferGAN suffers from catastophic forgetting of the cat class (left) that is somewhat mitigated by the EWC but it comes at the expense of increasing the FID score of the dog class (right).

pect these manipulations to work, but wanted to investigate their behavior. In Figure 6 we calculate the direction in the latent space that corresponds to the gender of the subject on the FFHQ class, and apply this direction to images for both classes. As can be seen, this direction has a clear effect on the FFHQ class, but not on the LSUN cat class, where it mainly affects the size of the cat. Another example can be seen in Figure 7, where we calculate the "add glasses" direction in the latent space for the FFHQ class. While this direction operates well on the FFHQ class, since the LSUN cat class does not have images of cats with glasses, it is not surprising that the effect is not carried over to cat images. Note, however, that this latent direction does affect the same semantic region — adding glasses is replaced by slightly increasing the cats' eyes.

6 Uncurated Generation Examples

In Figure 8 and Figure 9 we present uncurated images generated by the input source GAN models, by the baselines, and by our method.

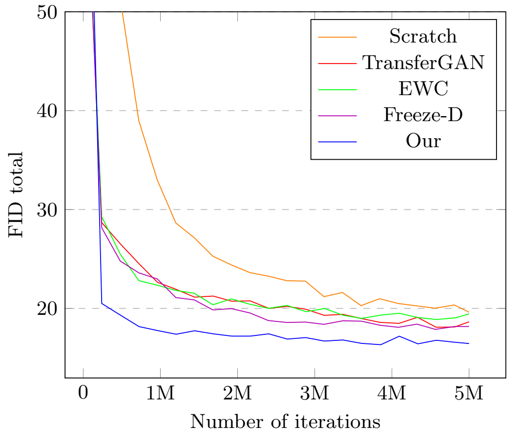


Fig. 3: Convergence rate of the total FID score during training on LSUN cat + FFHQ. Our solution achieves the lowest (best) FID score.

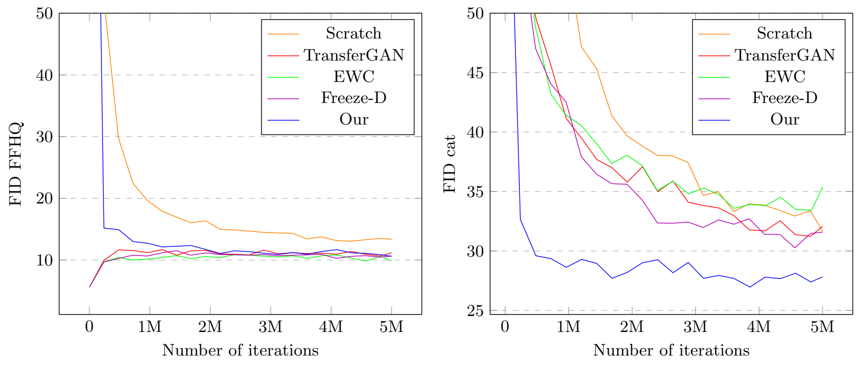


Fig. 4: FID scores calculated separately on FFHQ (left) and on cat (right). TransferGAN suffers from catastophic forgetting of FFHQ (left) that is mitigated by EWC which achieves slightly better results on this dataset than our method, but this comes at the expense of the FID score of the cat class (right), which is the worst for EWC out of all methods.

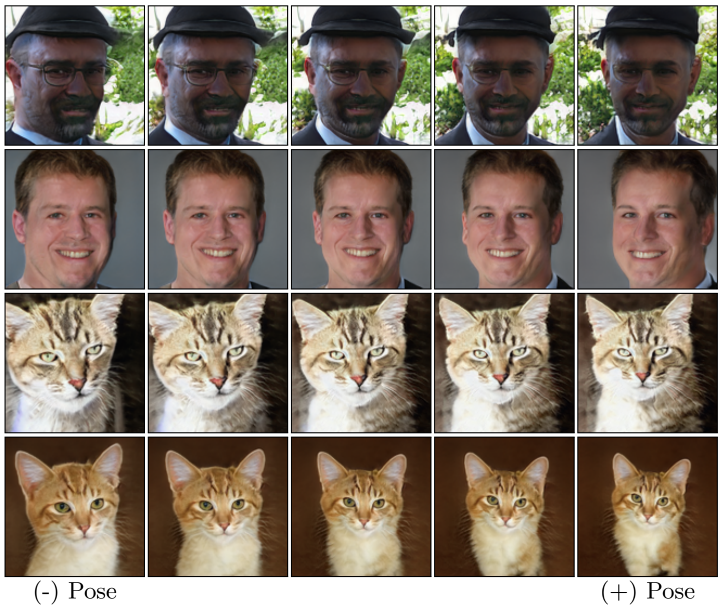


Fig. 5: We determine the pose direction in the latent space of the merged model of FFHQ and LSUN cat using images of the FFHQ class only. We then apply this direction to images from both classes and find that the semantics are largely preserved.

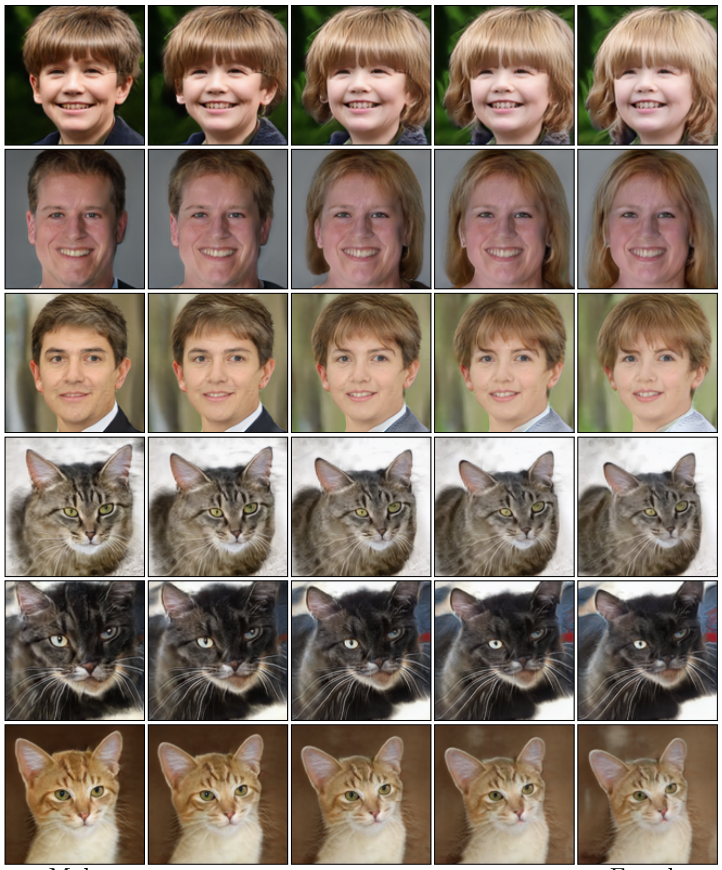


Fig. 6: We determine the gender direction in the latent space of the merged model of FFHQ and LSUN cat using images of the FFHQ class only. We then apply this direction to images from both classes. As expected, this operates accurately only on the FFHQ class.

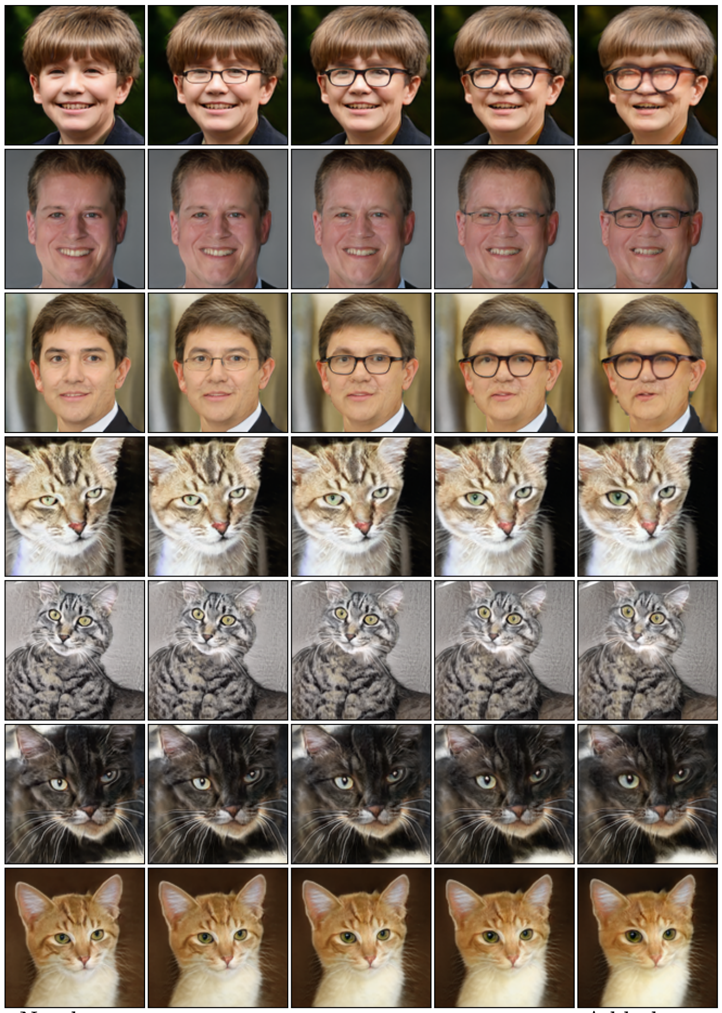
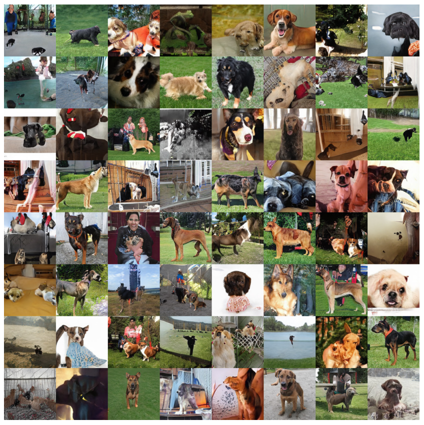
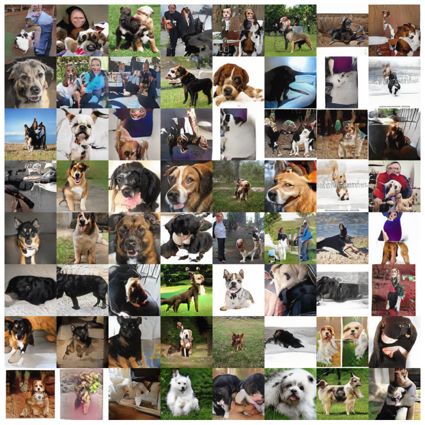
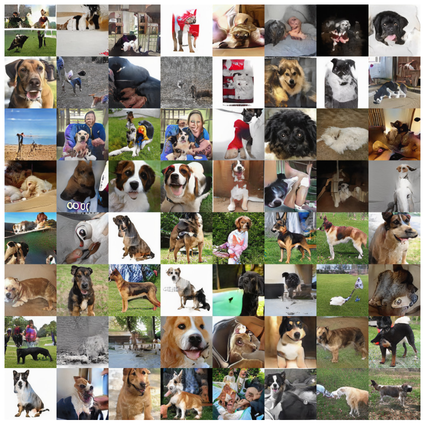


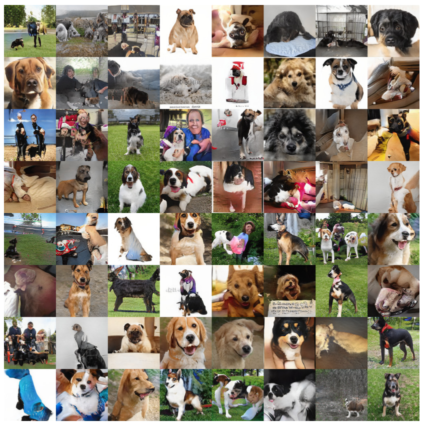
Fig. 7: We determine the glasses direction in the latent space of the merged model of FFHQ and LSUN cat using images of the FFHQ class only. We then apply this direction to images from both classes. The addition of glasses operates accurately only on the FFHQ class (as expected). On the cat class the same direction enlarges the eyes of the cats.


(a) Source model


(b) From scratch

(c) TransferGAN


(d) FreezeD


(e) EWC

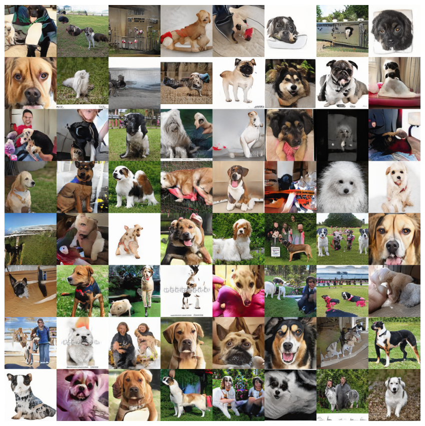
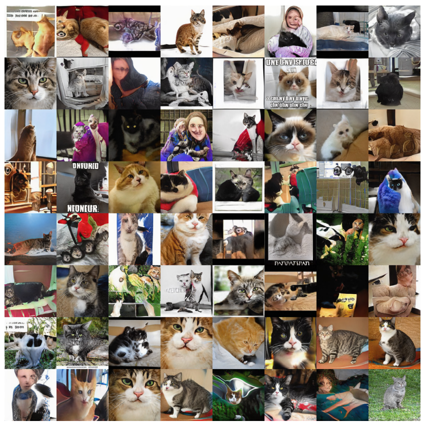

(f) Ours

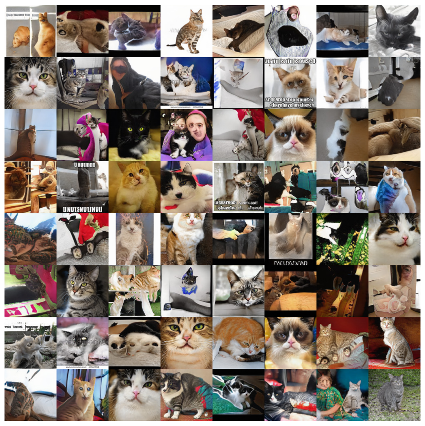
Fig. 8: Examples of uncurated images that were generated by the source model (a), the baselines (b-e), and our method (f) on LSUN dog dataset.

(a) Source model

(b) From scratch

(c) TransferGAN


(d) FreezeD


(e) EWC

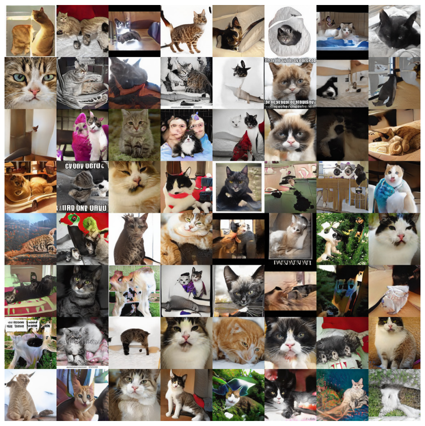

(f) Ours

Fig. 9: Examples of uncurated images that were generated by the source model (a), the baselines (b-e), and our method (f) on LSUN cat dataset.

References

1. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data. arXiv preprint arXiv:2006.06676 (2020) 1
2. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4401–4410 (2019) 4
3. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8110–8119 (2020) 1
4. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proc. ICLR (2015) 1
5. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017) 2, 3, 4, 5
6. Mescheder, L., Geiger, A., Nowozin, S.: Which training methods for GANs do actually converge? In: International conference on machine learning. pp. 3481–3490. PMLR (2018) 1
7. Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., et al.: Mixed precision training. arXiv preprint arXiv:1710.03740 (2017) 1
8. Miyato, T., Koyama, M.: cGANs with projection discriminator. arXiv preprint arXiv:1802.05637 (2018) 1
9. Mo, S., Cho, M., Shin, J.: Freeze discriminator: A simple baseline for fine-tuning GANs. arXiv preprint arXiv:2002.10964 (2020) 2, 3, 4, 5
10. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. arXiv preprint arXiv:1912.01703 (2019) 1
11. Wang, Y., Wu, C., Herranz, L., van de Weijer, J., Gonzalez-Garcia, A., Raducanu, B.: Transferring GANs: generating images from limited data. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 218–234 (2018) 2, 3, 4, 5
12. Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365 (2015) 2, 3, 4