Domain Expansion of Image Generators - Supplementary Materials

Yotam Nitzan[1,2] Michaël Gharbi[1] Richard Zhang[1] Taesung Park[1] Jun-Yan Zhu[3]
Daniel Cohen-Or[2] Eli Shechtman[1]

[1] Adobe Research [2] Tel-Aviv University [3] Carnegie Mellon University

A. Overview

In Appendix B, we consider a baseline for domain expansion and demonstrate it is inferior to our proposed method. Next follows the main part of the supplementary, Appendix C, in which we perform additional analysis and experimentation of our method. Finally, in Appendix D, we provide additional details completing the paper.

B. Domain Expansion Baseline Using Class-Conditioning

In this section, we experiment with an alternative, baseline, method to perform domain expansion. Generative models capturing multiple domains commonly use a class-conditioning mechanism [3]. Adopting this approach, we attempt to perform domain expansion by modeling domains with classes. We find that this method does not work as well as our proposed method.

Method. We start with an unconditional pretrained generator, specifically StyleGAN [13]. We then make the generator condition on a one-hot vector, using the architecture proposed by Karras et al. [11]. This change involves adding a single MLP layer, whose input is the one-hot vector. Its output is concatenated to the random latent code and then fed to the generator.

The class-conditioned generator is trained in a similar protocol to our method. The source domain uses class $c = 0$, which is analogous to the base subspace. Whenever the 0^{th} class is sampled, we apply the original loss $\mathcal{L}_{\text{src}}$ and the *memory replay* regularization (See Sec. 3.3). Formally, the loss describing this training is

$$\begin{aligned}\mathcal{L}_{\text{reg}} = \mathbb{E}_{z\sim p_{\text{src}}(z)} \big[\lambda_{\text{src}}\mathcal{L}_{\text{src}}(G(z, c=0))+ \\ \mathcal{L}_{\text{recon}}(G(z, c=0))\big],\end{aligned} \tag{1}$$

where $\mathcal{L}_{\text{recon}}$ is the memory-replay loss defined in Eq. (5) and $\lambda_{\text{src}} = 1$ is a hyperparameter weighting the losses. Other classes, analogous to repurposed subspaces, are dedicated to the newly introduced domains. Whenever the i^{th} class is sampled ($i > 0$), we apply the loss of the domain adaptation task $\mathcal{L}_i$. Applied over all new domains, the expansion loss is formally given by

$$\mathcal{L}_{\text{expand}} = \sum_{i=1}^{N} \mathbb{E}_{z\sim p_i(z)}\ \mathcal{L}_i(G(z, c=i)). \tag{2}$$

The final training objective still reads as $\mathcal{L}_{\text{full}} = \mathcal{L}_{\text{expand}} + \mathcal{L}_{\text{reg}}$.

Experiments. We expand an FFHQ [12] generator with two new domains, "Sketch" and "Tolkien Elf", introduced using StyleGAN-NADA [5]. We display the generated images using the same z latent codes for the different classes Fig. 1a.

We qualitatively observe that the expanded, class-conditioned generator preserves the source domain well, also expressed by preserving the FID [7] score. However, for new domains, we observe degraded performance from two aspects. First, the class-conditioned generator "leaks" knowledge between the classes. For example, in Fig. 1a, faces generated from the class dedicated to sketches also have long, elf-like, ears. Second, the domains are not "aligned". Despite being generated from the same z latent codes, the images differ beyond the differences between domains. For example, corresponding images from the source domain and elf domain often portray different head poses and facial expression. Therefore, it is not clear how can one obtain the elf "version" of a given face image, limiting the applications of such a model.

For reference, we display comparable results from our expansion method in Fig. 1b. As can be seen, our method does not suffer from these issues.

C. Additional Experiments

C.1. Latent Directions Analysis

Our method explicitly relies on the existence of dormant directions and their distinction from non-dormant directions. We wish to emphasize that the dichotomous distinction between "dormant" and "non-dormant" is a simplification. In Fig. 2, we report the mean LPIPS distance induced

(a) Class-conditioned baseline (b) Our domain expansion method

Figure 1. Experimenting with a class-conditioned baseline for domain expansion. (a) Images generated from a class-conditioned expanded model from the same z latent codes for the source, sketch, and elf domains. The source domain is preserved well in its dedicated class. However, the newly introduced domains "leak" information, expressed in long, elf-like, ears in the sketch domain. Additionally, the different domains are not well-aligned, as changing the domain also results in unrelated changes to head pose and facial expressions. (b) Comparable results from our domain expansion method, provided for reference. As can be seen, using our method, the domains do not interfere with each other and are well-aligned.

to images by a 3σ traversal along each direction. As can be seen, the distance is never exactly 0 and there is also no clear discontinuity. Nevertheless, it is clear that later directions, usually those beyond 100, cause significantly smaller perceptual change in the generated image. This behavior can also be qualitatively observed in Fig. 3.

As discussed in Sec. 4.1, this "almost" monotonous behavior is expected as our latent directions are right-singular vectors, sorted in decreasing order according to their corresponding singular values [36].

C.2. Effect of Choice of Direction for Domain

Our method dedicates a single dormant direction for every newly introduced domain. As mentioned in Sec. 4.1, all previous experiments used the *last* dormant directions, sorted in decreasing order according to their corresponding singular values. One might wonder: *Why should one use the last directions? And among the last directions, how should one match a direction to a domain?*

We now demonstrate that the specific choice of a latent direction has no significant impact on results, as long as it is dormant. To this end, we perform multiple expansions, each with 5 new domains introduced by StyleGAN-NADA [5], starting from a single generator pretrained on AFHQ [4]. For 4 of the new domains – "Siberian Husky", "Pixar", "Funny Dog", "Boar" – we dedicate the same directions in all experiments. Specifically, we use directions $507 - 510$, respectively. Directions numbers refer to their location in the decreasingly sorted right-singular vector set. Recall that the dimension of the latent space is 512, hence these directions are among the last ones. For the last domain, "Sketch", we vary the dedicated direction, using one of the directions $200, 300, 400, 500, 511$. We run the expansion twice with different random seeds.

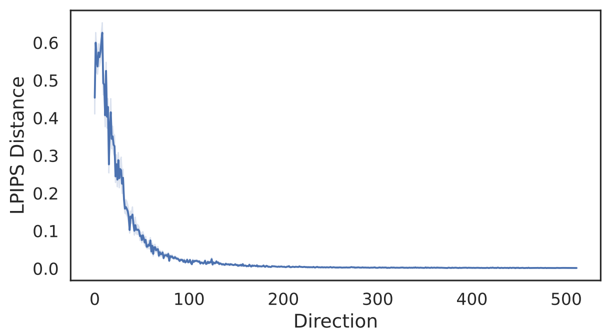
We study how the choice of direction for the Sketch domain affects its performance. In Fig. 4 (top) we report the CLIP error of images generated from the "Sketch" subspace with the prompt "a sketch" as a function of training iterations. We additionally display sample of generated images from each model in Fig. 4 (bottom). As can be seen, similar results are produced from different repurposed directions. Specifically, visual differences observed using different directions, are similar to those observed using the same directions but with different random seeds. This indicates that the differences between directions are negligible and might be entirely due to random chance.

Nevertheless, we do observe that certain directions minimize the CLIP error slightly more efficiently, across random seeds. We therefore run additional 5 expansions, using "Bear" instead of "Sketch". We now observe a different ordering of directions. We therefore conclude, that even if slight, imperceptible, differences exist between directions, they are not consistent across domains.

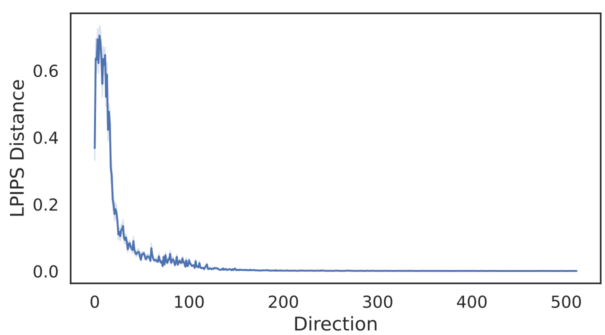
In summary, the choice of dormant direction has little to no effect. This result is arguably intuitive, as all dormant directions might be considered equivalent, having insignificant effect on generated images. Therefore, our choice of using the last directions is almost arbitrary, only motivated by the fact that they are the "most dormant". Similarly, no technique is required to match an direction to a domain, and one can simply pick a dormant direction randomly.

C.3. Repurposing Non-Dormant Directions

Aiming at domain expansion, preserving the source domain is integral. Since the non-dormant directions span the variations of the source domains, we explicitly kept them



(a) FFHQ



(b) LSUN Church

Figure 2. Magnitude of perceptual effect caused by traversing different directions. Directions are sorted in decreasing order according to their corresponding singular values. For each direction, we measure the LPIPS distance [49] between images from two latent codes distanced by a 3σ traversal along the direction. As can be seen, the effect caused by the traversal diminishes quickly and the majority of directions are dormant.

intact, and repurposed only dormant directions. Nevertheless, the training method itself could be identically applied to non-dormant directions. One simply needs to dedicate a non-dormant direction to capture the new domain. We next demonstrate that applying our method to non-dormant directions is still effective and enables capabilities beyond domain expansion.

Traversing the 1^{st} latent direction in the generator pretrained on FFHQ [12], makes people in generated images appear older and more masculine. Some users might decide that they associate having a full beard with being older and more masculine. To support such behavior, we fine-tune the generator with a transformed StyleGAN-NADA [5], to capture "a person with a beard" along the 1^{st} direction. We display images generated along traversals of the 1^{st} direction, before and after tuning, in Fig. 6a. As can be seen, the generator now represents having a beard, along its 1^{st} latent direction, in addition to its previous behavior.

The capability to add new concepts in addition to existing ones does not depend on the close relationship between the two in the last examples. To demonstrate this point, we tune the generator to capture "Elf" along its 8^{th} direction, which originally encodes head pose (and a few other properties). Results are displayed in Fig. 6b.

Previous results are clearly not solving domain expansion, as they alter the original behavior of the source domain. Instead, one might say they adapt the domain modeled by the generator. Nevertheless, there exists a profound difference to existing domain adaptation methods. Our resulting generator does not completely overriding the source domain. Instead, in a precise and controllable manner, it modifies individual factors of variation. Therefore, a user can carefully *rewrite* [1, 43] the semantic rules of a generative model, allowing greater control and freedom.

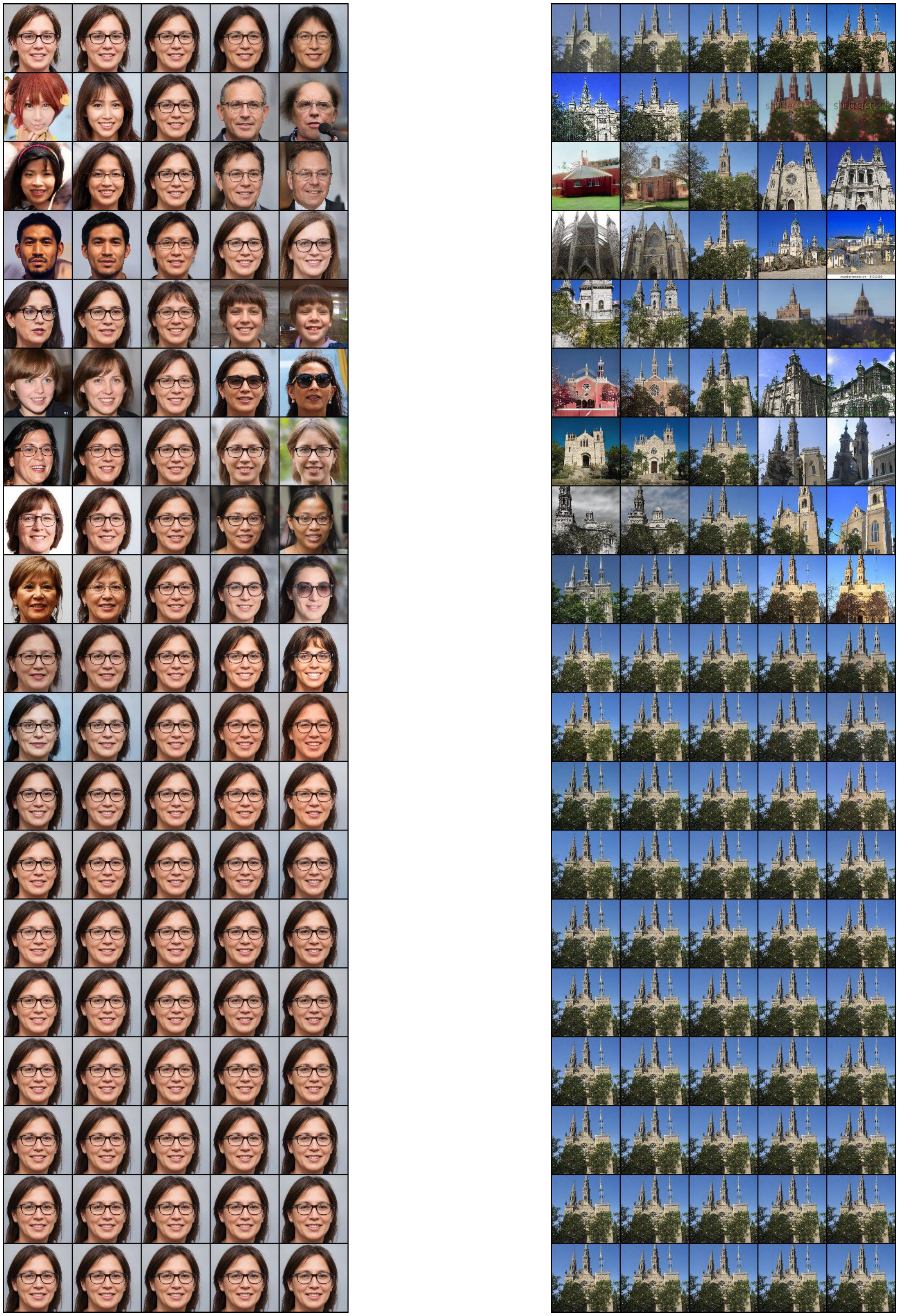
C.4. Distance to Repurposed Subspace

Repurposed subspaces are defined by transporting the base subspace along a linear direction by a predetermined scalar size s (See Eq. (3) in the main paper). All results in the paper, across domains and variations used $s = 20$. We next evaluate the effect the hyperparameter s has on results. To this end, we perform multiple expansions of an FFHQ [12] generator with 100 new variations, while varying the value of s.

We measure CLIP errors (introduced in Sec. 4.3) of images generated from repurposed subspaces and the corresponding target text used for training, as a function of training iterations. In Fig. 7a we report the results for two variations - "Marge Simpson" and "Tolkein Elf". As can be seen, for all $s > 0$, CLIP error decreases as training progresses, and it decreases "faster" for greater values of the parameter s. Even with $\times 10$ more iterations, the model trained with $s = 5$ does not reach the CLIP error of the model trained with $s = 20$.

Images generated from the repurposed subspaces are displayed in Fig. 7b. For each value of s, we use the checkpoint that resulted in the closest CLIP error to that obtained by a favored $s = 20$ checkpoint. As can be seen, not only training time is affected by parameter s, but the visual effects captured by training vary significantly.

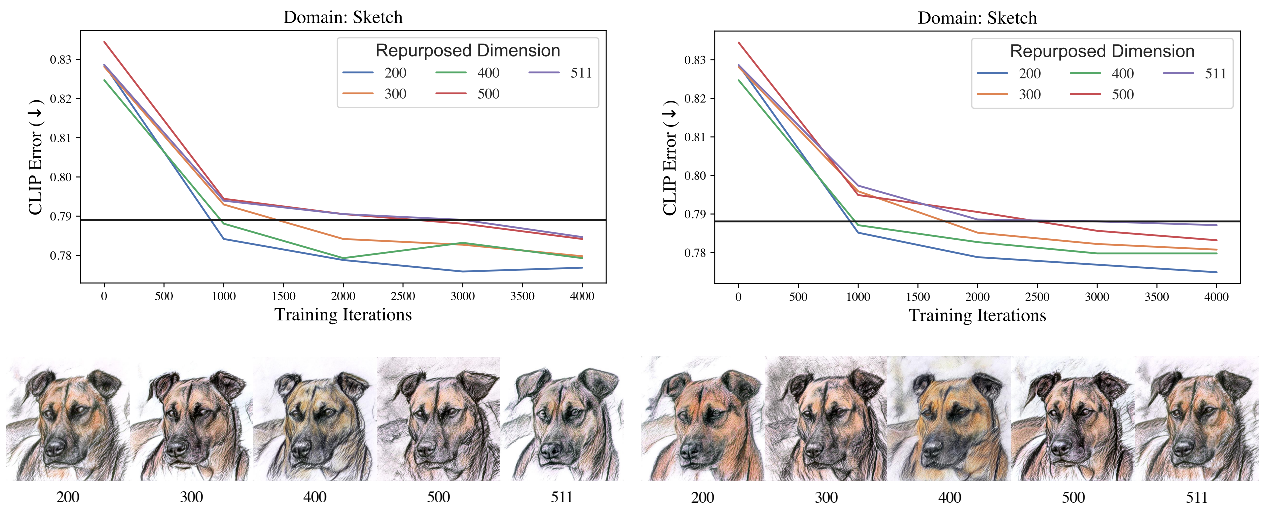
We observe that models trained with greater values of parameter s depict a more significant change with respect to the source domain. When parameter s is too small (*e.g.*, $s \leq 5$), the model captures only few, simple characteristics of the new domain. On the other hand, when parameter s is too large (*e.g.*, $s = 50$), the model commonly generates images that are blurry, have color artifacts or even do not capture the target text well. For example, with the target text "Marge Simpson", the model learns to generate images with blue skin rather than blue hair. We note that these undesired artifacts cannot be mitigated by training with a large value of parameter s originally, and use a smaller one in test-time,



(a) FFHQ

(b) LSUN Church

Figure 3. Visualization of $\pm 3\sigma$ traversal along latent directions in the FFHQ [12] and LSUN Church [48] models, obtained using SeFA [36]. Directions shown are sorted from least (v_0, top) to most (v_{511}, bottom) dormant. As can be seen, later directions are dormant – not affecting the generated image. We over-sample early directions for clarity. In practice, over 80% of directions are dormant.



(a) Seed 0

(b) Seed 1

Figure 4. We expand a generator pretrained on AFHQ [4] with 5 domains, varying the dormant direction dedicated to the "sketch" domain. We repeat the expansion twice, with different random seeds. Top - reporting CLIP error of images generated from the sketch domain with the text "a sketch". Bottom - a sample of generated images from checkpoints obtaining CLIP error closest to the horizontal black line. As can be seen, images generated using different repurposed dimensions differ only slightly. Specifically, changing the random seed induces similar difference.

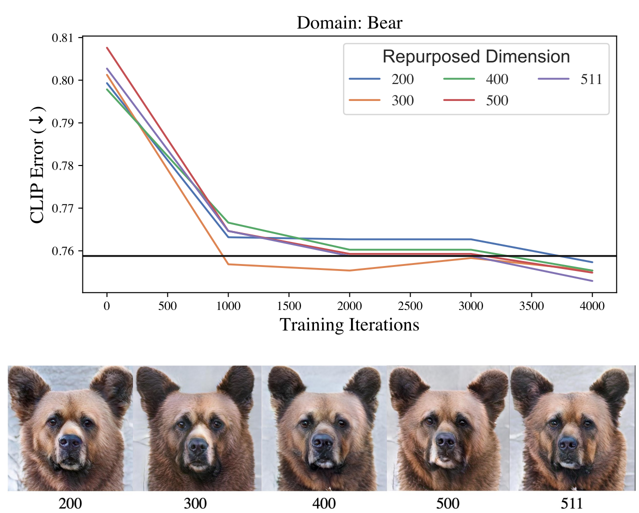


Figure 5. Similar to Fig. 4, using a "bear" domain instead of "sketch". As can be seen, dimensions are ordered differently in terms of minimizing CLIP error, as compared to their order for sketch.

as demonstrated in Fig. 8.

Following these results, we conclude that the parameter s has a regularizing effect. Placing the domains "closer" in the latent space causes them to be more similar in image space as well. Conversely, placing the domains further apart allows the new domain to capture more drastic, out-of-domain effects.

Eventually, choosing a value for parameter s is subject to user preference. In our experiments, we have found that values in the range of $[10, 30]$ offer satisfying results, across different source and expanded domains.

We last note that the regularization effect of parameter s could be explained by the existence of a globally consistent "pace of change" of the generator with respect to the latent space. With StyleGAN, such behavior is explicitly encouraged using a Perceptual Path Length (PPL) regularization term [13]. Nevertheless, we observe identical results when omitting this regularization during our expansion.

C.5. How Many Domains Can Fit?

So far, the largest number of new domains used for expansion was 105. The results from Appendix C.1 indicated that there might be up to 400 dormant directions. *Could they all be repurposed?*

We apply our method to expand a generator pretrained on FFHQ with 400 new domains, repurposing the last (and perhaps all) dormant directions. Incredibly, the expansion succeeds. We find that the expansion follows the same findings discussed in Sec. 4.3 – training is slower, yet quality is uncompromised. Specifically, the FID score from the base subspace is 2.83 compared to 2.80 in our model expanded with 105 domains. We display images generated from this model in the accompanying video and in Figs. 9 to 11.

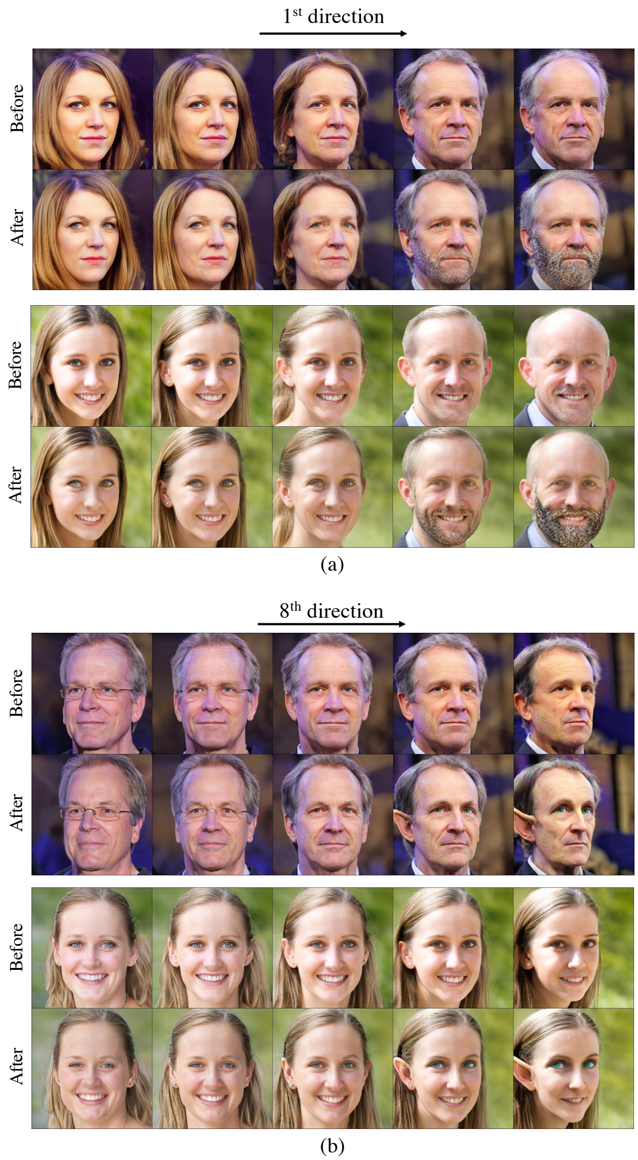


Figure 6. Using our training method with non-dormant direction rewrites existing semantic rules and adds new concepts on top of existing ones. (a) Traversing the 1^{st} direction originally made people older and more masculine. After fine-tuning, it also adds a beard. (b) Traversing the 8^{th} direction originally turned people heads. After fine-tuning it also turns them to elves.

C.6. Additional Compositions Results

In Figs. 12 and 13 we provide additional qualitative results displaying compositionality in expanded generators.

D. Additional Details

D.1. Training Time and Iterations

When expanding the generator with a single new domain, our training requires roughly twice the number of iterations to obtain comparable effects. The difference is a direct result of our additional regularization terms. With additional domains, we observe a roughly linear relationship between the number of domains and the required training iterations. For example, the FFHQ model expanded with 105 iterations was trained for $40K$ iterations, while the model with 400 iterations was trained for $150K$ iterations.

Note that different training objective might require a different number of iterations. StyleGAN-NADA [5] specifically heavily relies on early-stopping. An ideal domain expansion method could consider this issue, and sample training objectives to apply non-uniformly. In practice, we did not observe this to be an issue, probably due to our method optimizing numerous objectives simultaneously.

D.2. Transformation of Loss Function

As explained in Sec. 3.2, transforming a given domain adaptation task to perform domain expansion requires limiting the samples latent codes. The loss function itself, in principal, is left unchanged. This is exactly the case for MyStyle [21]. For StyleGAN-NADA [5], however, we made a subtle modification to the loss function.

StyleGAN-NADA computes its loss with respect to a frozen copy of the source generator (See Sec. 4.1). This is done in order to maintain access to the source domain, despite it vanishing from the adapted generator during training. Conversely, using our method, the source domain is preserved along the base subspace. We take advantage of this fact and modify the loss only slightly. Instead of using a frozen generator to generate images from the source domain, we simply use our expanded generator and latent codes from the base subspace.

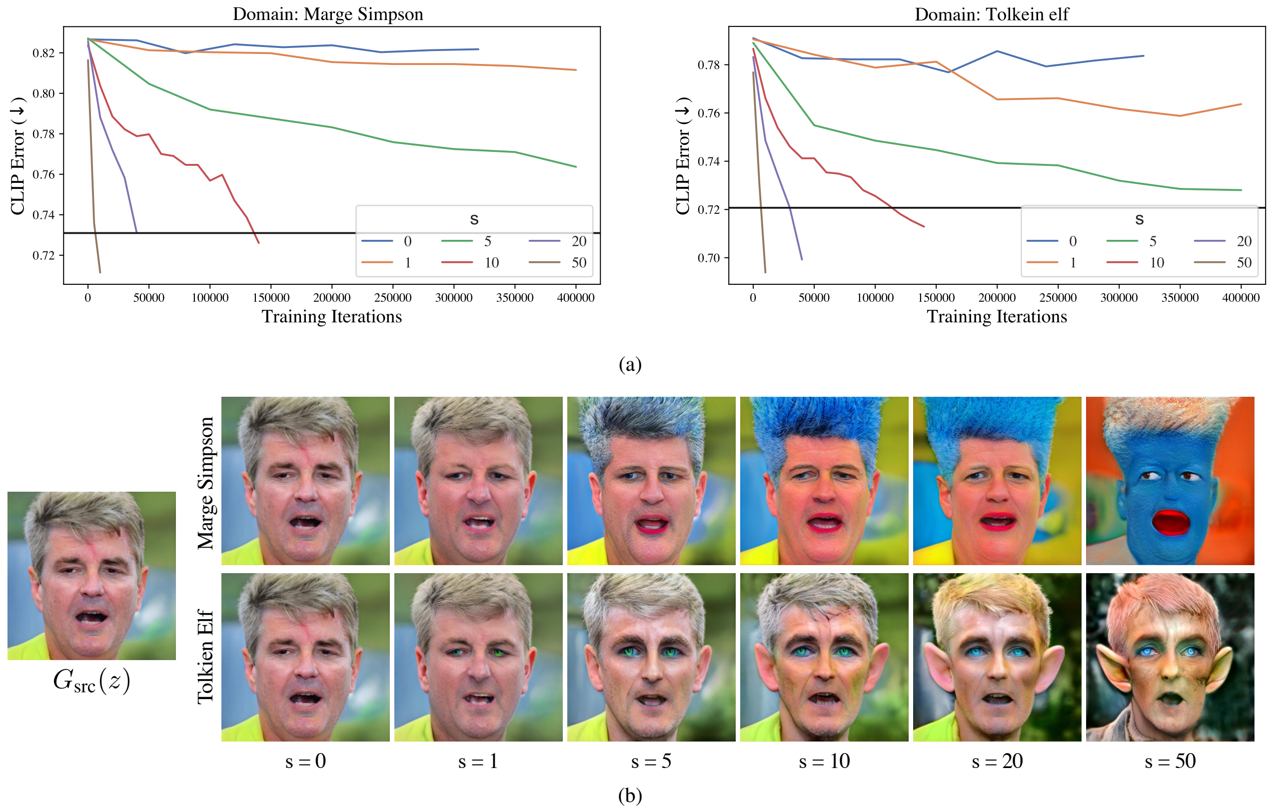


Figure 7. Evaluating the effect of the distance between the base and repurposed subspace, s. (a) We compare CLIP error as a function of training iterations, between models trained with different values of parameter s. (b) Generated images from models having CLIP error as close as possible to the black horizontal line. As can be seen, increasing s corresponds to faster minimization of CLIP error. However, even with comparable CLIP errors, visual effect might vary significantly. Large values of parameter s are often associated with undesired artifacts. We find that values between $[10, 30]$ are usually preferable.

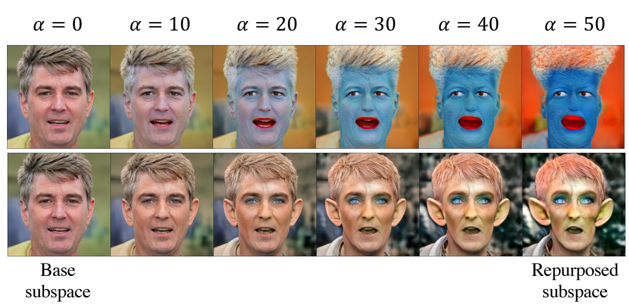


Figure 8. Interpolation between the base subspace and the repurposed subspace where $s = 50$. As can be seen, undesired behavior occurring at repurposed subspace (*e.g.* blue skin Marge Simpson) cannot be mitigated by traversing shorter distances in test time. The choice of parameter s is crucial in training time.

References

[1] David Bau, Steven Liu, Tongzhou Wang, Jun-Yan Zhu, and Antonio Torralba. Rewriting a deep generative model. In *European conference on computer vision*, pages 351–369. Springer, 2020. 3

[2] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *arXiv preprint arXiv:2005.07727*, 2020.

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 1

[4] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8188–8197, 2020. 2, 5

[5] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *arXiv preprint arXiv:2108.00946*, 2021. 1, 2, 3, 6

[6] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. GANSpace: Discovering interpretable GAN controls. *arXiv preprint arXiv:2004.02546*, 2020.

Figure 9. Subset 1/3 of generated images from a model expanded with 400 domains.

Figure 10. Subset 2/3 of generated images from a model expanded with 400 domains.

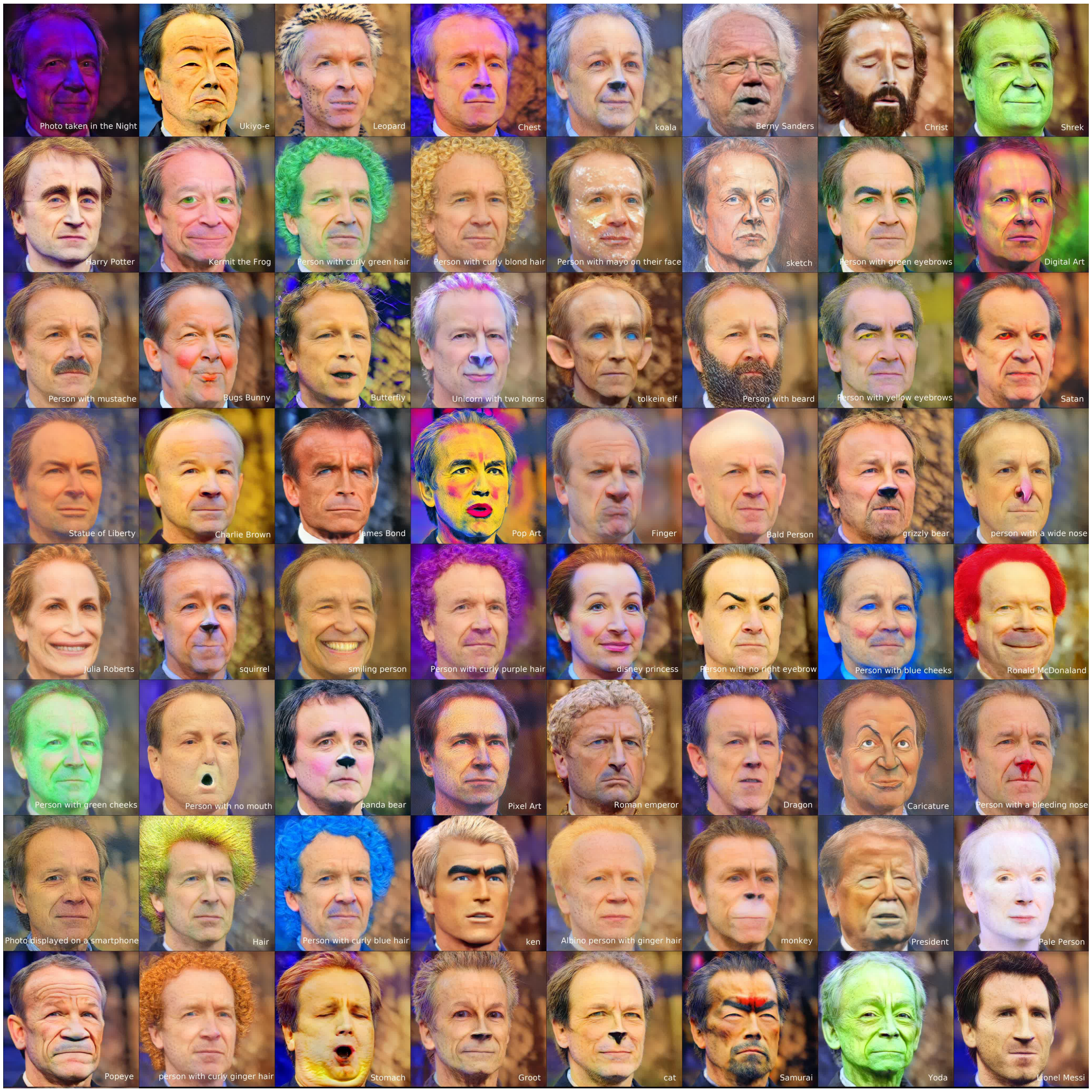


Figure 11. Subset 3/3 of generated images from a model expanded with 400 domains.

[7] 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. *Advances in neural information processing systems*, 30, 2017. 1

[8] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

[9] Yuge Huang, Yuhan Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, Jilin Li, and Feiyue Huang. CurricularFace: Adaptive curriculum learning loss for deep face recognition, 2020.

[10] Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. *arXiv preprint arXiv:1907.07171*, 2019.

[11] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020. 1

[12] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. CVPR*, pages 4401–4410, 2019. 1, 3, 4, 11

[13] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, pages 8110–8119, 2020. 1, 5

[14] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image

Figure 12. Composition of factors of variation introduced to a generator pretrained on FFHQ [12]. Following the format of Fig. 8
.

manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2426–2435, 2022.

[15] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[16] Nupur Kumari, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Ensembling off-the-shelf models for gan training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10651–10662, 2022.

[17] Yijun Li, Richard Zhang, Jingwan Lu, and Eli Shechtman. Few-shot image generation with elastic weight consolidation. *arXiv preprint arXiv:2012.02780*, 2020.

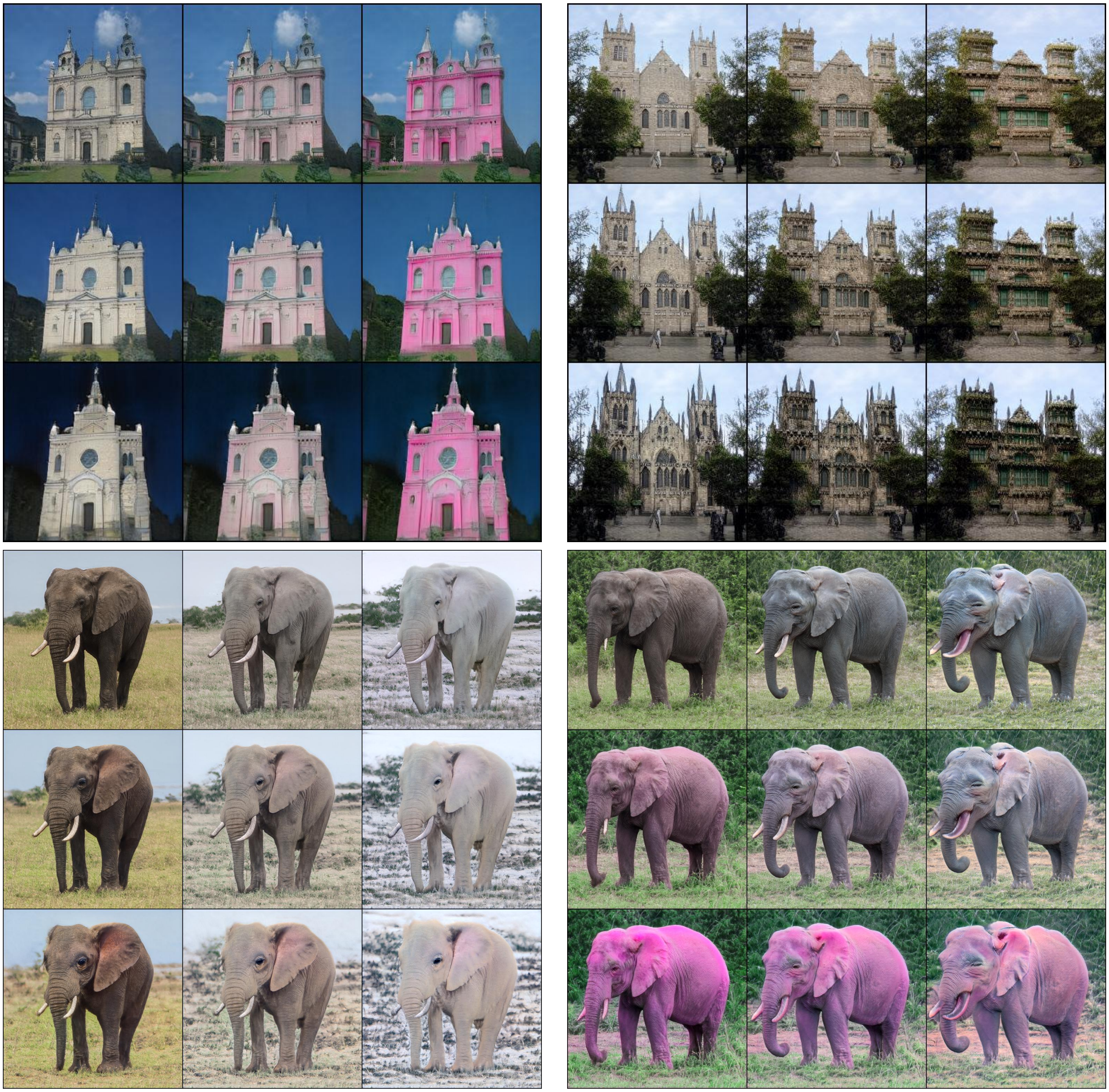


Figure 13. Composition of factors of variation introduced to generators pretrained on LSUN Church [48] and SD-Elephant [20]. Following the format of Fig. 8

.

[18] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In Gordon H. Bower, editor, *Psychology of Learning and Motivation*, volume 24, pages 109–165. Academic Press, 1989.

[19] Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Freeze the discriminator: a simple baseline for fine-tuning GANs. *arXiv preprint arXiv:2002.10964*, 2020.

[20] Ron Mokady, Michal Yarom, Omer Tov, Oran Lang, Michal Irani Daniel Cohen-Or, Tali Dekel, and Inbar Mosseri. Self-distilled stylegan: Towards generation from internet photos, 2022. 12

[21] Yotam Nitzan, Kfir Aberman, Qiurui He, Orly Liba, Michal Yarom, Yossi Gandelsman, Inbar Mosseri, Yael Pritch, and

Daniel Cohen-Or. Mystyle: A personalized generative prior. *arXiv preprint arXiv:2203.17272*, 2022. 6

[22] Utkarsh Ojha, Yijun Li, Jingwan Lu, Alexei A Efros, Yong Jae Lee, Eli Shechtman, and Richard Zhang. Few-shot image generation via cross-domain correspondence. *arXiv preprint arXiv:2104.06820*, 2021.

[23] Xingang Pan, Xiaohang Zhan, Bo Dai, Dahua Lin, Chen Change Loy, and Ping Luo. Exploiting deep generative prior for versatile image restoration and manipulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[24] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. *Advances in Neural Information Processing Systems*, 33:7198–7211, 2020.

[25] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. StyleCLIP: Text-driven manipulation of StyleGAN imagery. *arXiv preprint arXiv:2103.17249*, 2021.

[26] William Peebles, John Peebles, Jun-Yan Zhu, Alexei Efros, and Antonio Torralba. The hessian penalty: A weak prior for unsupervised disentanglement. In *ECCV*. Springer, 2020.

[27] Phillip Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. The intrinsic dimension of images and its impact on learning. *arXiv preprint arXiv:2104.08894*, 2021.

[28] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10619–10629, 2022.

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[30] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[31] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *arXiv preprint arXiv:2106.05744*, 2021.

[32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[33] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *arXiv preprint arXiv:2208.12242*, 2022.

[34] Ari Seff, Alex Beatson, Daniel Suo, and Han Liu. Continual learning in generative adversarial nets. *arXiv preprint arXiv:1705.08395*, 2017.

[35] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. InterFaceGAN: interpreting the disentangled face representation learned by GANs. *arXiv preprint arXiv:2005.09635*, 2020.

[36] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in GANs. *arXiv preprint arXiv:2007.06600*, 2020. 2, 4

[37] Nurit Spingarn-Eliezer, Ron Banner, and Tomer Michaeli. Gan" steerability" without optimization. *arXiv preprint arXiv:2012.05328*, 2020.

[38] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for StyleGAN image manipulation. *arXiv preprint arXiv:2102.02766*, 2021.

[39] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33:19667–19679, 2020.

[40] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021.

[41] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the GAN latent space. *arXiv preprint arXiv:2002.03754*, 2020.

[42] Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. Sketch your own gan. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14050–14060, 2021.

[43] Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. Rewriting geometric rules of a gan. *ACM Transactions on Graphics (TOG)*, 41(4):1–16, 2022. 3

[44] Chenshen Wu, Luis Herranz, Xialei Liu, Joost van de Weijer, Bogdan Raducanu, et al. Memory replay gans: Learning to generate new categories without forgetting. *Advances in Neural Information Processing Systems*, 31, 2018.

[45] Zongze Wu, Dani Lischinski, and Eli Shechtman. StyleSpace analysis: Disentangled controls for StyleGAN image generation. *arXiv:2011.12799*, 2020.

[46] Zongze Wu, Yotam Nitzan, Eli Shechtman, and Dani Lischinski. Stylealign: Analysis and applications of aligned stylegan models. *arXiv preprint arXiv:2110.11323*, 2021.

[47] Ceyuan Yang, Yujun Shen, Yinghao Xu, and Bolei Zhou. Data-efficient instance generation from instance discrimination. *arXiv preprint arXiv:2106.04566*, 2021.

[48] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 4, 12

[49] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 3

[50] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. *arXiv preprint arXiv:2006.10738*, 2020.

[51] Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. Mind the gap: Domain gap control for single shot domain adaptation for generative adversarial networks. *arXiv preprint arXiv:2110.08398*, 2021.