Learning Style Subspaces for Controllable Unpaired Domain Translation

Gaurav Bhatt
University of British Columbia
Vancouver, Canada
gauravbhatt.cs.iitr@gmail.com

Vineeth N Balasubramanian
Indian Institute of Technology Hyderabad
Hyderabad, India
vineethnb@cse.iith.ac.in

In this supplementary material, we present some additional ablation studies that could not be included in the main paper due to space constraints, in particular:

- Training details of our experiments (in continuation to Section 4).
- Discussion related of controllable generation of each subspace used in our work (in continuation to Section 4.2).
- Discussion related to varying number of chances given to baselines while computing $ACG@r$ (in continuation to Section 4.2).
- Experiments with different non-linearities used with partition network (in continuation to Section 4.2).
- Experiments on using multiple domains (more than 2 domains) (in continuation to Section 4.2).
- Experiments on reference-guided image synthesis on CeleA-HQ dataset (in continuation to Section 4.2).

1. Training details of our experiments

In all our experiments, we train our model on a single P100 GPU with batch size of 8. All images are resized to 256×256. We train the models for 120K iterations (about 3-4 days to train the models). The values of hyperparameters as defined in Section 3.3 are kept as λ_{par} = 1, λ_{pc} = 1 and λ_o = 1 for CelebA-HQ, and λ_{par} = 1, λ_{pc} = 1 and λ_o = 2 for AFHQ. Similar to [2], we use weight decay to stabilize training where the weight λ_o is linearly decayed to zero over the 120K iterations. The L_o loss (Equation 7) consists of the non-saturating adversarial loss [3] with R1 regularization [8] using γ = 1. For optimization, we rely on the Adam [6] optimizer with β_1 = 0 and β_2 = 0.99. The learning rates for all networks (G, E, PN, and M) are set to 10^{-4}. The weights of all modules are initialized using He's initialization [4] and all biases are kept as zero, except for biases associated with the scaling vectors of AdaIN that are set to one.

Non-linearity	male-to-female		female-to-male	
	StarGAN_v2 [2]	proposed	StarGAN_v2 [2]	proposed
Bags Under Eyes	0.504	**0.672**	0.920	**0.940**
Big Lips	0.940	**0.984**	0.672	**0.788**
Big Nose	0.568	**0.572**	0.908	**0.912**
Black Hair	0.360	**0.504**	0.444	**0.644**
Blond Hair	0.216	**0.432**	0.052	**0.256**
Brown Hair	0.504	**0.684**	0.424	**0.656**
Double Chin	0.068	**0.408**	0.164	**0.816**
Mouth Slightly Open	0.524	**0.596**	0.444	**0.464**
Smiling	0.648	**0.808**	0.492	**0.628**
Young	0.956	**0.982**	0.824	**0.972**
Young + Black Hair	0.428	**0.437**	0.376	**0.688**
Young + Blond Hair	0.128	**0.540**	0.052	**0.268**
Young + Brown Hair	0.452	**0.532**	0.344	**0.760**
Smiling + Black Hair	0.236	**0.238**	0.188	**0.304**
Smiling + Blond Hair	0.120	**0.480**	0.020	**0.092**
Smiling + Brown Hair	0.292	**0.380**	0.224	**0.312**
total	0.434	**0.572**	0.409	**0.593**

Table 1. Classification accuracies of individual subspaces used in our work. We use a total of sixteen subspaces with ten individual subspaces and six subspace combinations. Total accuracy is $ACG@1$.

2. Controllable generation of each subspace

In continuation to Section 4.2, here we present ablations with each subspace used in our work. In celebA-HQ dataset, we have 10 subspaces - 'Bags Under Eyes', 'Big Lips', 'Blond Hair', 'Big Nose', 'Black Hair', 'Double Chin', 'Oval Face', 'Mouth Slightly Open', 'Smiling', 'Brown Hair', 'Young'. We introduce 6 more subspaces that are combinations - 'Young+Black hair', 'Young+Blonde Hair', 'Young+Brown Hair','Smiling+Black Hair', 'Smiling+Brown Hair', 'Smiling+Blonde Hair'. We thus have a total of 16 subspaces. The individual accuracies are shown in 1 where we compare the proposed model with StarGAN_v2 [2]. Note that the total accuracy corresponds to $ACG@1$. Interestingly, the proposed method outperforms a strong baseline like StarGAN_v2 on accuracies of all individual subspaces. The difference in performance is larger for combinations of subspaces.

Method	ACG@1	ACG@2	ACG@3	ACG@4	ACG@5
MUNIT-GAN [5]	0.403	0.409	0.413	0.415	0.415
ACL-GAN [9]	0.413	0.417	0.424	0.424	0.424
StarGAN_v2 [2]	0.425	0.427	0.432	0.432	0.432
DosGAN [7]	0.453	0.459	0.465	0.466	0.466
Homomorphic [1]	0.471	0.478	0.484	0.484	0.484
Proposed	**0.572**	**0.582**	**0.585**	**0.585**	**0.586**

Table 2. Results on average controllable generation on male-to-female task.

Method	ACG@1	ACG@2	ACG@3	ACG@4	ACG@5
MUNIT-GAN [5]	0.371	0.375	0.387	0.389	0.390
ACL-GAN [9]	0.383	0.387	0.398	0.398	0.401
StarGAN_v2 [2]	0.409	0.415	0.423	0.425	0.425
DosGAN [7]	0.452	0.459	0.468	0.470	0.470
Homomorphic [1]	0.471	0.478	0.485	0.491	0.491
Proposed	**0.593**	**0.597**	**0.607**	**0.610**	**0.610**

Table 3. Results on average controllable generation on female-to-male task.

3. Varying number of chances given to baselines while computing $ACG@r$

In Section 4.2, we use $ACG@r$ for evaluating the performance of UDT methods that do not permit controllable generation, where we demonstrate the results with r =1 and r=5. Here, we extend the results on all values of $r \in \{1, 2, 3, 4, 5\}$. We also vary r for controllable generation methods as well. The results for male-to-female are shown in Table 2 while the results for female-to-male are shown in Table 3 We observe that for non-controllable generation methods ,the ACG improves when r is increase from 1 to 3; however, the ACG score doesn't change much when r is increased to 5 and beyond. We believe that when such non-controllable generation methods have fair chances to predict (i.e. $r = 3$), they show an increase in ACG score. Nonetheless, their scores are significantly below those of the proposed method; in particular, they do not do well on generation of certain attributes such as 'old' female translation from a 'young' male image (shown in Figure 1 - main draft). This is the reason for a steady value of ACG when r is increased further. We observe the same trend for male-to-female and female-to-male translation tasks.

4. Introducing non-linearity in the partition network (PN)

The partition network (PN, defined in Section 3.1) decomposes the style space into smaller subspaces. It is also possible to use a non-linearity with the PN. We experiment with the use of non-linearity over the representations learned:

$$f_p^x = \sigma([f_p; f_x]) \text{ and } f_p^z = \sigma([f_p; f_z]) \tag{1}$$

where σ is the non-linearity introduced.

The results are shown in Table 4, where we experiment with sigmoid, ReLU, leakyReLU, tanh, and concat (no non-linearity). We observe that with sigmoid and concat, the quality of translations (in terms of LPIPS and FID score) are better while the ACG scores don't have much difference.

5. Experiments on multiple domains

Since our method is capable of handling multiple domains simultaneously, we present our experiments on AFHQ dataset with 3 domains - cat, dog, wild animals. We peform reference-guided image synthesis and the results are shown in Figure 1. It is evident that we can handle multiple domains simultaneously without introducing much computational cost (i.e. without multiple generators).

6. Scalability

We discuss the scalability of UD2D methods in Section 2 of our paper. Herein, we present more insights to this issue. Most domain translation methods are based on generative models that are computationally expensive to train. The number of trainable parameters becomes a bottleneck when number of domains are increased, as the requirement of generators and discriminators is dependent on the number of domains needed. For n domains, there are $\binom{n}{2}$ combinations of functional mappings needed to be learned in general. Thus, the number of generators and discriminators that are needed by most methods are proportional to $\binom{n}{2}$. However, the trainable parameters of the proposed method does not grow proportionally to $\binom{n}{2}$, as there is no need to add extra generator or discriminators (we only need to add a linear MLP layer for each domain). This makes the proposed method scalable as compared to most existing methods. Note that our method has a near-constant parameter count with increase in domains, while almost all other methods increase with domains. In terms of training time, we used a single NVIDIA P100 GPU (16 GB VRAM) to train our method on all datasets with a training time on high-resolution datasets (both celeba-HQ and AFHQ) of a max of 2 days.

7. Reference-guided image synthesis on CelebA-HQ

Here, we present qualitative results on celebA-HQ (in continuation to our results from Section 4.2). The results are shown in Figure 2.

References

[1] Ying-Cong Chen, Xiaogang Xu, Zhuotao Tian, and Jiaya Jia. Homomorphic latent space interpolation for unpaired image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2408–2416, 2019.

Figure 1. Multi-domain image translation. Here we use 3 domains - cat, dog, wild animals. We perform reference-guide image translation where the content is extracted from the source image (top row) and the style is extracted from another source image (first column).

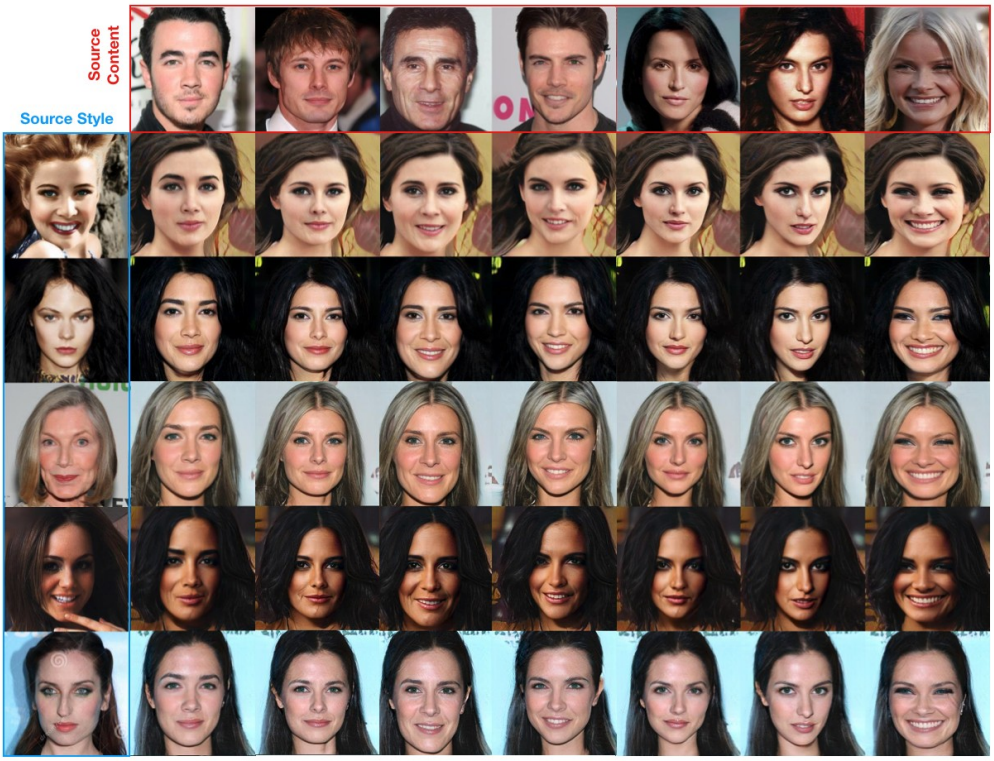

Figure 2. Reference-guided image translation on celebA-HQ dataset.

Non-linearity	male-to-female			female-to-male		
	LPIPS ↑	FID ↓	ACG@1	LPIPS	FID	ACG@1
Relu	0.44	15.02	0.56	0.49	21.43	0.58
Sigmoid	0.45	11.79	0.57	0.46	16.42	0.59
Tanh	0.44	17.10	0.56	0.43	25.22	0.58
LeakyRelu	0.45	15.42	0.57	0.45	19.98	0.59
Concat	0.45	12.65	0.57	0.46	18.57	0.59

Table 4. Introducing non-linearity to the partition network. The results are shown for gender translation task on celebA-HQ (male-to-female and female-to-male). We evaluate the performance based on quality and diversity of generated samples. ↑: higher the better; ↓: lower the better.

[2] 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.

[3] 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.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[5] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 172–189, 2018.

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[7] Jianxin Lin, Zhibo Chen, Yingce Xia, Sen Liu, Tao Qin, and Jiebo Luo. Exploring explicit domain supervision for latent space disentanglement in unpaired image-to-image translation. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[8] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018.

[9] Yihao Zhao, Ruihai Wu, and Hao Dong. Unpaired image-to-image translation using adversarial consistency loss. *arXiv preprint arXiv:2003.04858*, 2020.