Supplementary Material

Overview

This appendix is organized as follows:

Appendix A provides the mathematical formulae used in hyperbolic neural networks. Sec 3.2 & Sec 3.3

Appendix B gives more implementation details of HAE. Sec 4.1

Appendix C shows the results of the ablation study of downstream tasks for Animal Faces [5], Flowers [7] and VGGFaces [8]. Sec 4.3

Appendix D compares the embeddings of images in hyperbolic space and Euclidean space. Sec 4.4

Appendix E visualizes the interpolation on different radii in the Poincaré disk, along the geodesic, and on $\mathcal{W}^+$-space. Sec 4.3

Appendix F shows the images generated with different radii in the Poincaré disk. Sec 4.3

Appendix G compares the images generated by state-of-the-art few-shot image generation method, *i.e.* AGE [1] and our methods HAE. Sec 4.4

Appendix H gives more details of the user study we conducted. Sec 4.4

Appendix I gives more examples generated by HAE. Sec 4.4

A. Hyperbolic Neural Networks

For hyperbolic spaces, since the metric is different from Euclidean space, the corresponding calculation operators also differ from Euclidean space. Recall that in Eq. (11), we have two operations: Möbius addition and Möbius scalar multiplication [4], given fixed curvature c.

For any given vectors $x, y \in \mathbb{H}^n$, the *Möbius addition* is defined by:

$$x \oplus_c y = \frac{\left(1 - 2c\langle x, y\rangle - c\|y\|_2^2\right) x + \left(1 + c\|x\|_2^2\right) y}{1 - 2c\langle x, y\rangle + c^2\|x\|_2^2\|y\|_2^2}, \quad (1)$$

where $\|\cdot\|$ denotes the 2-norm of the vector, and $\langle\cdot,\cdot\rangle$ denotes the Euclidean inner product of the vectors.

Similarly, the *Möbius scalar multiplication* of a scalar r and a given vector $x \in \mathbb{H}^n$ is defined by:

$$r \otimes_c x = \tan_c\left(r \tan_c^{-1}\left(\|x\|_2\right)\right) \frac{x}{\|x\|_2}. \quad (2)$$

We also would like to give explicit forms of the exponential map and the logarithmic map which are used in our model to achieve the translation between hyperbolic space and Euclidean space as mentioned in Sec 3.2.

The *exponential map* $\exp_x^c : T_x\mathbb{D}_c^n \cong \mathbb{R}^n \to \mathbb{D}_c^n$, that maps from the tangent spaces into the manifold, is given by

$$\exp_x^c(v) := x \oplus_c \left(\tanh\left(\sqrt{c}\frac{\lambda_x^c\|v\|}{2}\right)\frac{v}{\sqrt{c}\|v\|}\right). \quad (3)$$

The *logarithmic map* $\log_x^c(y) : \mathbb{D}_c^n \to T_x\mathbb{D}_c^n \cong \mathbb{R}^n$ is given by

$$\log_x^c(y) := \frac{2}{\sqrt{c}\lambda_x^c}\operatorname{arctanh}\left(\sqrt{c}\,\|-x \oplus_c y\|\right)\frac{-x \oplus_c y}{\|-x \oplus_c y\|}. \quad (4)$$

B. Implementation Details and Analysis

As mentioned in Sec 3.2, the output of pSp: $\mathbf{w}_i \in \mathbb{R}^{18\times512}$. The MLP encoder $\texttt{MLP}_E$ used in HAE, is split into three parts: encoder_{low}, encoder_{mid}, and encoder_{high} for encoding lower layer attributes, middle layer attributes, and higher layer attributes. The encoder_{low} is a 5-layer MLP with a Leaky-ReLU (slope=0.2) activation function. The first three layers of $\mathbf{w}_i$ are then fed into encoder_{low}. The dimension of the output attribute is 128. Similar to encoder_{low}, the encoder_{mid} is also an 5-layer MLP. 3-7 layers of $\mathbf{w}_i$ is then fed into encoder_{mid}, the dimension of the output attribute is also 128. Different from encoder_{low} and encoder_{mid}, encoder_{high} is an 8-layer MLP. And we fed the last 12 layers attributes of $\mathbf{w}_i$ into it. The dimension of the output attribute of encoder_{high} is 256. Therefore, the final dimension of the Euclidean latent code $z_{\mathbb{R}i}$ is $128 + 128 + 256 = 512$. While the MLP decoder $\texttt{MLP}_D$ is the reversed version of $\texttt{MLP}_E$, taking $z'_{\mathbb{R}i} \in \mathbb{R}^{512}$ as input and output $\mathbf{w}'_i \in \mathbb{R}^{18\times512}$.

During the training process, the constants defined in Eq. (10) are set as $\lambda_1 = 1$, $\lambda_3 = 0.3$, and λ_2 changes dynamically based on the value of $\mathcal{L}_{\text{rec}}$ which guarantees that $0.6 \geq \lambda_2\mathcal{L}_{\text{rec}} \geq 0.3$. Besides, we employ Adam optimizer with a learning rate of $1\text{e}{-4}$, and the batch size is set to 8.

In addition, as a remark, we choose the largest radius as 6 in most of our experiments as in hyperbolic space since any vector asymptotically lying on the surface unit N-sphere will have a hyperbolic length of approximately $r = 6.2126$, which can be directly calculated by Eq. (2).

Finally, we want to show that HAE can be easily trained. The size of trainable parameters of HAE is around one hundred million which is small compared with other models with billions of parameters. It can be trained well within one day using a single NVIDIA TITAN RTX.

C. Ablation Study of Downstream Task

Similar to the ablation study in Sec 4.3. We also conduct data augmentation via HAE for image classification on Animal Faces [5], Flowers [7], and VGGFaces [8]. Due

to the limited size of Flowers and VGGFaces datasets. We randomly select 10, 15 and 15 images for each category as train, val and test, respectively. Following [2, 1], a ResNet-18 backbone is initialized from the seen categories, then the model is fine-tuned on the unseen categories referred to as the *baseline*. 30 images are generated for each unseen category as data augmentation.

The result of Animal Faces is shown in Tab. 1. It shows that the accuracy of the classifier improves after using the AdamW optimizer. The experiment result essentially confirms the original result in Sec 4.3. The data augmentation improves the performance of the classifier when the hyperbolic radius $r_{\mathbb{D}}$ is larger than 4. $r_{\mathbb{D}} = 5$ achieves the best performance on the classification experiment mainly because it achieves the best trade-off between the quality and diversity. However, the performance drops when the radius is smaller than 4. This is because the semantic attributes change too much and thus mislead the classifier.

The result of Flowers is presented in Tab. 2. Similar to the result of Animal Faces, the diversity and quality of generated images are largely controlled by the hyperbolic radii $r_{\mathbb{D}}$. As the radius becomes smaller, HAE generates images of higher diversity but categories also gradually change to others. $r_{\mathbb{D}} = 6$ achieves the best performance on the classification experiment.

However, Tab. 3 shows that all accuracy drops when we do data augmentation on VGGFaces dataset. We estimate that this is due to the low quality of inversion that harms the performance of the classifier. Besides, since we only select 10 images for each category for training, with the limited size of VGGFaces, it is easy to overfit. To evaluate our estimation, we further test the performance of the classifier trained by the original images and inversion images without any perturbation, denoted as *inversion* in Tab. 3. This result proves that our estimation is correct. The accuracy of the classifier trained by augmented images increases compared with the *inversion*, which shows that the augmentation still works and improves the generalization performance of our classifier. $r_{\mathbb{D}} = 3$ achieves the best performance on the classification experiment except the *baseline*.

It is also worth noticing that, the FID and LPIPS scores inTab. 1, Tab. 2, and Tab. 3 are different from the scores we calculated in Sec 4.4. That is because we only use a very small subset of the data in this ablation study which can not represent the distribution of all images in the test dataset. Besides, the improvement of a classifier trained on augmented data is trivial, we believe this is due to the limitation of the encoding method,*i.e*. psp [9] and generator,*i.e*. StyleGAN2 [3], which can not generate images with high enough quality.

Hyperbolic Radius	Accuracy	FID(↓)	LPIPS(↑)
baseline	67.34	-	-
6.0	68.22	**46.89**	0.4520
5.5	68.56	48.68	0.4651
5.0	**69.33**	52.08	0.4823
4.5	68.22	60.87	0.5174
4.0	67.67	65.83	0.5386
3.5	67.33	68.44	0.6034
3.0	66.89	69.40	**0.6316**

Table 1: Ablation of same perturbation on different radii on Animal Faces.

Hyperbolic Radius	Accuracy	FID(↓)	LPIPS(↑)
baseline	71.76	-	-
6.0	**79.21**	**94.35**	0.4640
5.5	77.25	98.09	0.4871
5.0	75.29	97.81	0.5110
4.5	78.82	97.53	0.5330
4.0	75.29	97.58	0.5499
3.5	73.33	101.52	0.6152
3.0	72.55	105.05	**0.6439**

Table 2: Ablation of same perturbation on different radii on Flowers.

Hyperbolic Radius	Accuracy	FID(↓)	LPIPS(↑)
baseline	**77.99**	-	-
inversion	69.53	**25.46**	0.2325
6.0	71.98	26.19	0.2702
5.5	72.53	26.46	0.2887
5.0	72.45	26.83	0.3080
4.5	72.32	26.92	0.3258
4.0	72.96	27.02	0.3405
3.5	74.05	26.35	0.4044
3.0	<u>74.44</u>	25.90	**0.4411**

Table 3: Ablation of same perturbation on different radii on VGGFaces.

D. Comparison with Euclidean space

In this section, we mainly compare hyperbolic space with Euclidean space by UMAP visualization [6], which is an extension of our analysis in Sec 4.4. Following the UMAP visualization on hyperbolic space for the Animal

Faces [8] dataset, we first show the UMAP visualization of the embeddings of images in $\mathcal{W}^+$-space, where each embedding is of 18×512-dimension and therefore we resize them for UMAP calculation. The results for Euclidean UMAP of Animal Faces dataset are shown in Fig. 1. We observe that although the transition across different categories is smooth, there are no obvious clusters in the plot even for some significantly different species (*e.g.*, polar bears and foxes).

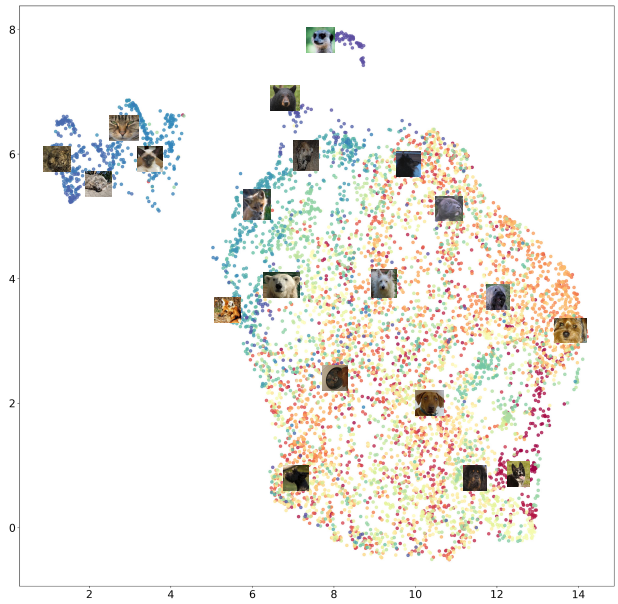

Figure 1: UMAP visualization for Animal Faces dataset, while the embeddings are in the $\mathcal{W}^+$-space of the same model.

We further carry on the experiments on the other two datasets. For Flowers [7] dataset, we use all images in the test dataset to generate the embeddings for both spaces, where there are 101 classes in total. The number of images in each class varies due to the original setting of the dataset. The results of the Flowers dataset are shown in Fig. 3, where clusters are more obvious in hyperbolic space and the similarity between classes is also well-reflected.

For the VGGFaces dataset [8], the results are shown in Fig. 4. Similarly, the clusters are better represented in hyperbolic settings. We observe that in hyperbolic space, images with similar low-level attributes (*e.g.*, wearing black frame glasses, having mutton chops beard) are clustered. We need to pay attention to the small clusters in both plots where images are represented with tall rectangles (in the plot). These images do not share semantic attributes but are clustered together, which can be the influence of heavy watermarks on the images. This also encourages us to train the model with high-quality datasets for better GAN inversions.

E. Interpolation Visualization

In this section, we offer a more detailed comparison of different radii and latent spaces. The results are shown in Fig. 5, where r refers to the ratio of the whole geodesic described in Eq. (11) starting from image A to image B, *e.g.*, when $r = 0.5$, the interpolation is exactly the hyperbolic mean of these two images. We observe that in $\mathcal{W}^+$-space, both high-level attributes and low-level attributes changed together while in hyperbolic space, we can achieve more detailed editing on low-level attributes while keeping high-level attributes unchanged, while the radius can control the degree of change more precisely. When the hyperbolic radius is large, the category of the given image remains the same before reaching the middle point of the curve. This property allows us to generate diverse images of the given image without changing its category-relevant attributes. As the hyperbolic radius becomes smaller, the higher-level attributes gradually change in the early stage of the interpolation. The interpolation visualization on geodesic shows that the image gradually changes from fine-grained to abstract to fine-grained. These results also explain our method of adding details by rescaling after taking the geodesic which will lead to a relatively abstract average.

F. Images Generated with Different Radii

In this section, we give more examples of images generated by HAE with different radii in the Poincaré disk. As Fig. 6 shows, the high-level attributes, *a.k.a.* category-relevant attributes do not change when the radius is large which allows us to generate diverse images without changing the category. However, the images generated by HAE become more abstract and semantically diverse when the hyperbolic radius $r_{\mathbb{D}}$ becomes smaller. The images gradually lose fine-grained details and change higher-level attributes as they move closer to the center of the Poincaré disk. For the few-shot image generation task, large radii work well since we want to change the category-irrelevant attributes of a given image. Nevertheless, our method HAE is not only capable of few-shot image generation but has great potential for other downstream tasks. For instance, HAE is able to generate a bunch of images of felines given an image of a cat. This can be done by rescaling the latent code to a relatively small radius in the hyperbolic space. Then we can add random perturbation to get the average code of multiple categories of felines. Finally, diverse images with fine-grained details of felines can be generated by moving those average codes to their children with larger radii.

G. Comparison with State-of-the-art Few-shot Image Generation Method

We also provide a comparison of images generated by the state-of-the-art method, *i.e*. AGE [1] and our methods on three datasets. As Fig. 7 shows, our method is able to generate images with more semantic diversity. For instance, for dogs, HAE generates images with different light conditions and angles compared with images generated by AGE. Furthermore, for the woman in the third row from the bottom, HAE can change the image from a colored photo to a monochrome photo without changing her identity. However, our method also slightly changes some attributes compared with the original images, *e.g*., the hair color of dogs and the petal color of flowers. This is because the color varies within the category in these datasets which can indicate color is a category-irrelevant attribute for those categories. Therefore, our method does not change the category-relevant attributes but has more semantic diversity. Besides, images generated by HAE look more natural compared with images generated by AGE. Most importantly, AGE requires datasets with labels to learn the feature code book. However, the hierarchical representation in HAE can be learned using unsupervised or self-supervised learning if we have enough computing resources. Therefore, our work has great potential and can be applied to many other downstream tasks in future work.

H. User Study

As mentioned, we conducted an extensive user study with a fully randomized survey. Results are shown in the main text. Specifically, we compared AGE and HAE in the following protocol:

1. We randomly chose 20 images per dataset, and for each image, we then generated 3 variants using AGE and HAE, respectively. Overall, there were 60 original images and 180 generated variants in total.

2. For each sample of each model, we grouped the 3 generated variants together, denoted as an *image block*. We then shuffled the following orders in the dataset: 1) order of images, 2) order of each block, 3) order of images in the block, an illustration is shown in Fig. 2.

3. We gathered 50 volunteers from various backgrounds who were asked to choose one image block for one sampled image based on their evaluation of image diversity and quality subjectively. The results are then re-arranged.

The result breakdowns are shown as follows: **Animal Faces**: 658/1000; **Flowers**: 523/1000; **VGGFaces**: 562/1000. We also provide more examples used in the user study in Fig. 8.

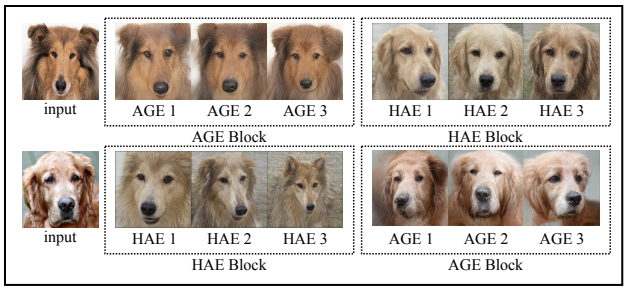

Figure 2: Illustration of shuffling in the user study, where inputs, blocks, and generated variants in each block were shuffled.

I. Additional Examples Generated by HAE

We provide more samples generated by HAE in Fig. 9, Fig. 10, and Fig. 11. The radius we choose is 6.2126 and the length of perturbation is 8.

References

[1] Guanqi Ding, Xinzhe Han, Shuhui Wang, Shuzhe Wu, Xin Jin, Dandan Tu, and Qingming Huang. Attribute group editing for reliable few-shot image generation. In *CVPR*, pages 11184–11193, 2022. 1, 2, 4

[2] Zheng Gu, Wenbin Li, Jing Huo, Lei Wang, and Yang Gao. Lofgan: Fusing local representations for fewshot image generation. In *ICCV*, 2021. 2

[3] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, pages 8107–8116, 2020. 2

[4] Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *CVPR*, pages 6417–6427, 2020. 1

[5] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsueprvised image-to-image translation. In *ICCV*, 2019. 1

[6] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018. 2

[7] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729, 2008. 1, 3

[8] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015. 1, 3

[9] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *CVPR*, pages 2287–2296, 2021. 2

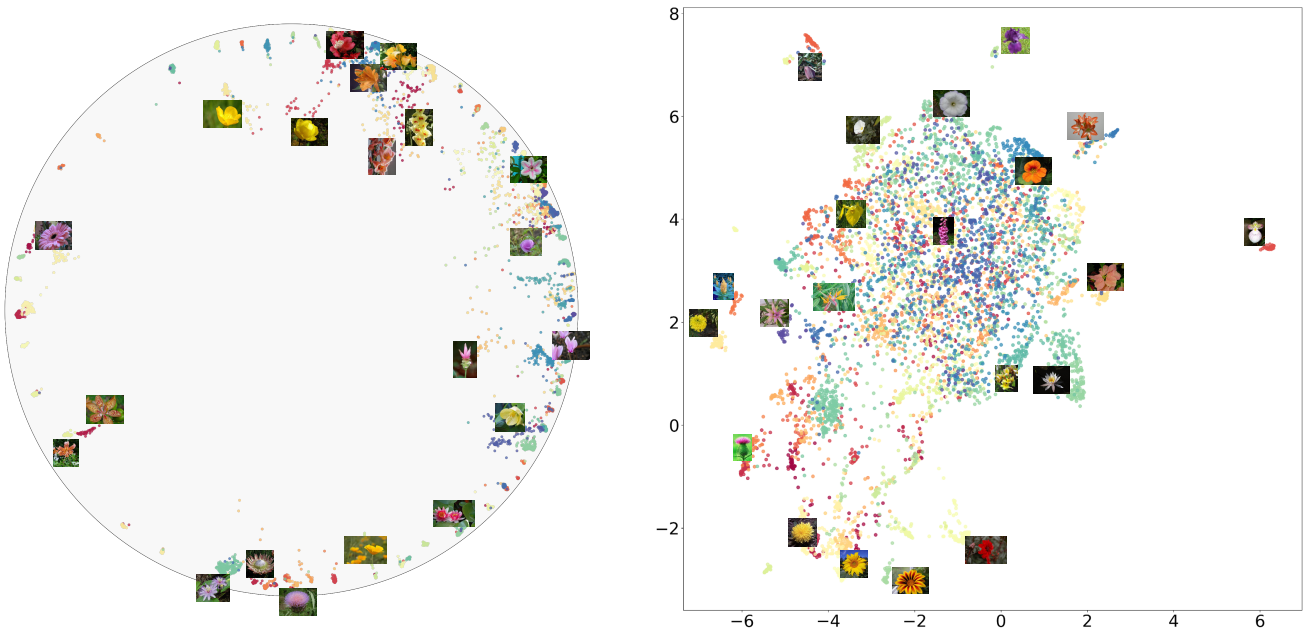

Figure 3: UMAP Visualization for Flowers dataset. Left: Hyperbolic space. Right: Euclidean space ($\mathcal{W}^+$-space).

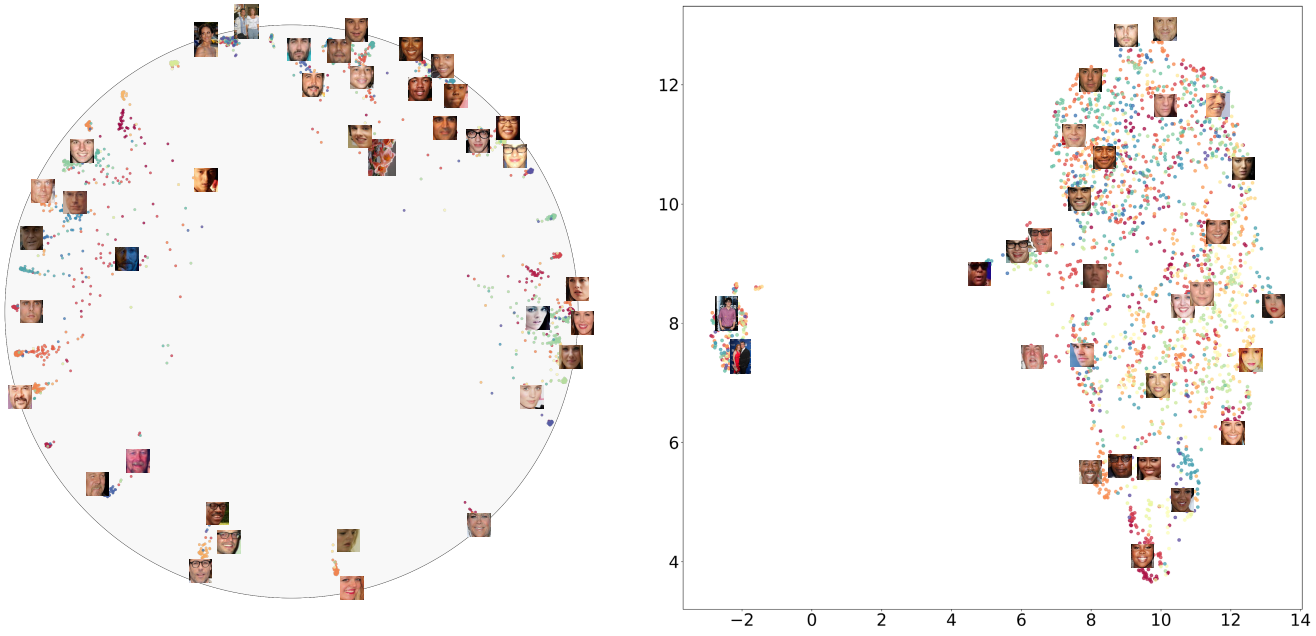

Figure 4: UMAP Visualization for VGGFaces dataset. Left: Hyperbolic space. Right: Euclidean space ($\mathcal{W}^+$-space). In each visualization, no images are from the same category.

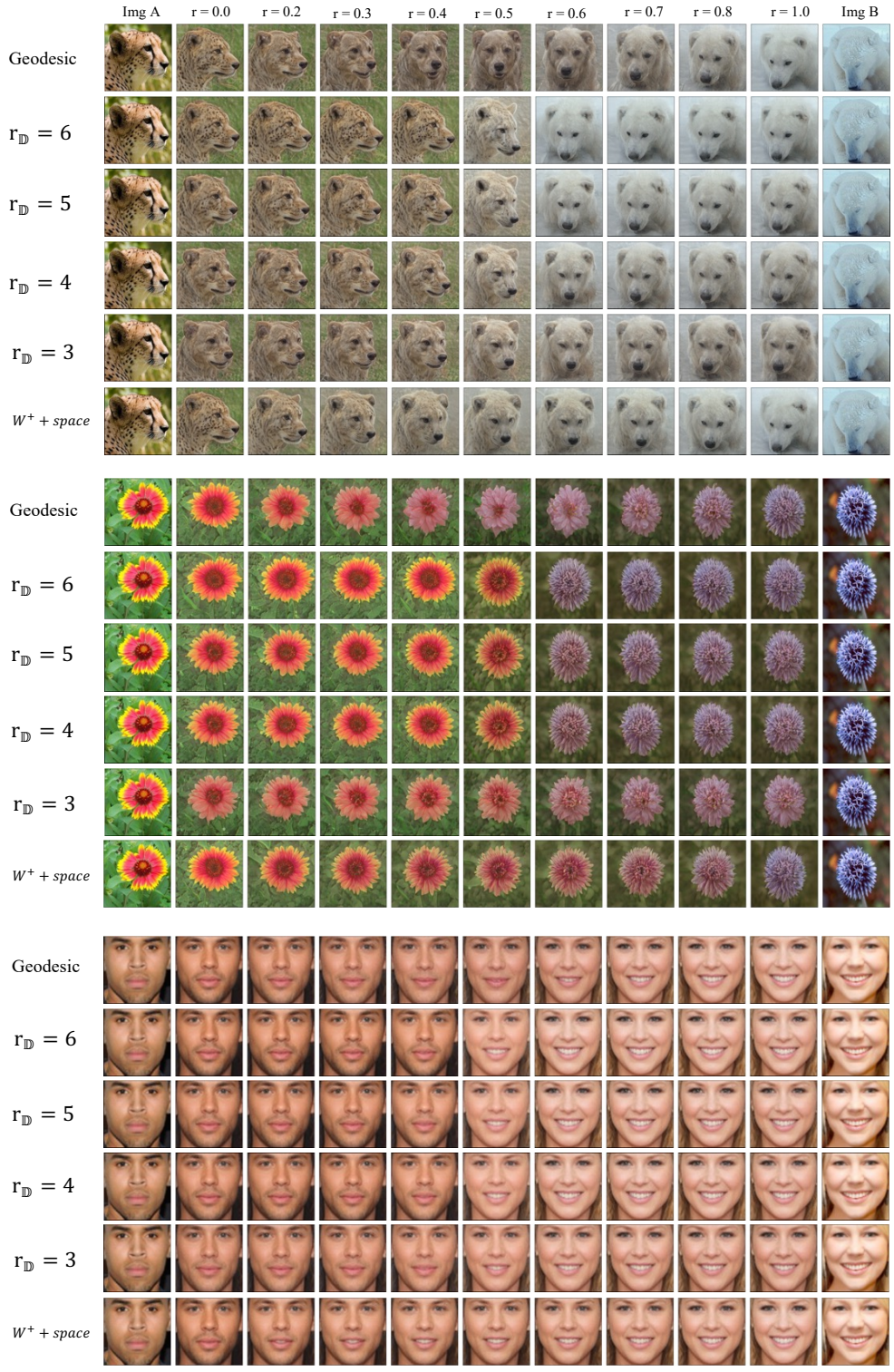

Figure 5: Interpolation along the geodesic and different radii in hyperbolic space and along the straight line in $\mathcal{W}^+$-space on Animal Faces, Flowers, and VGGFaces.

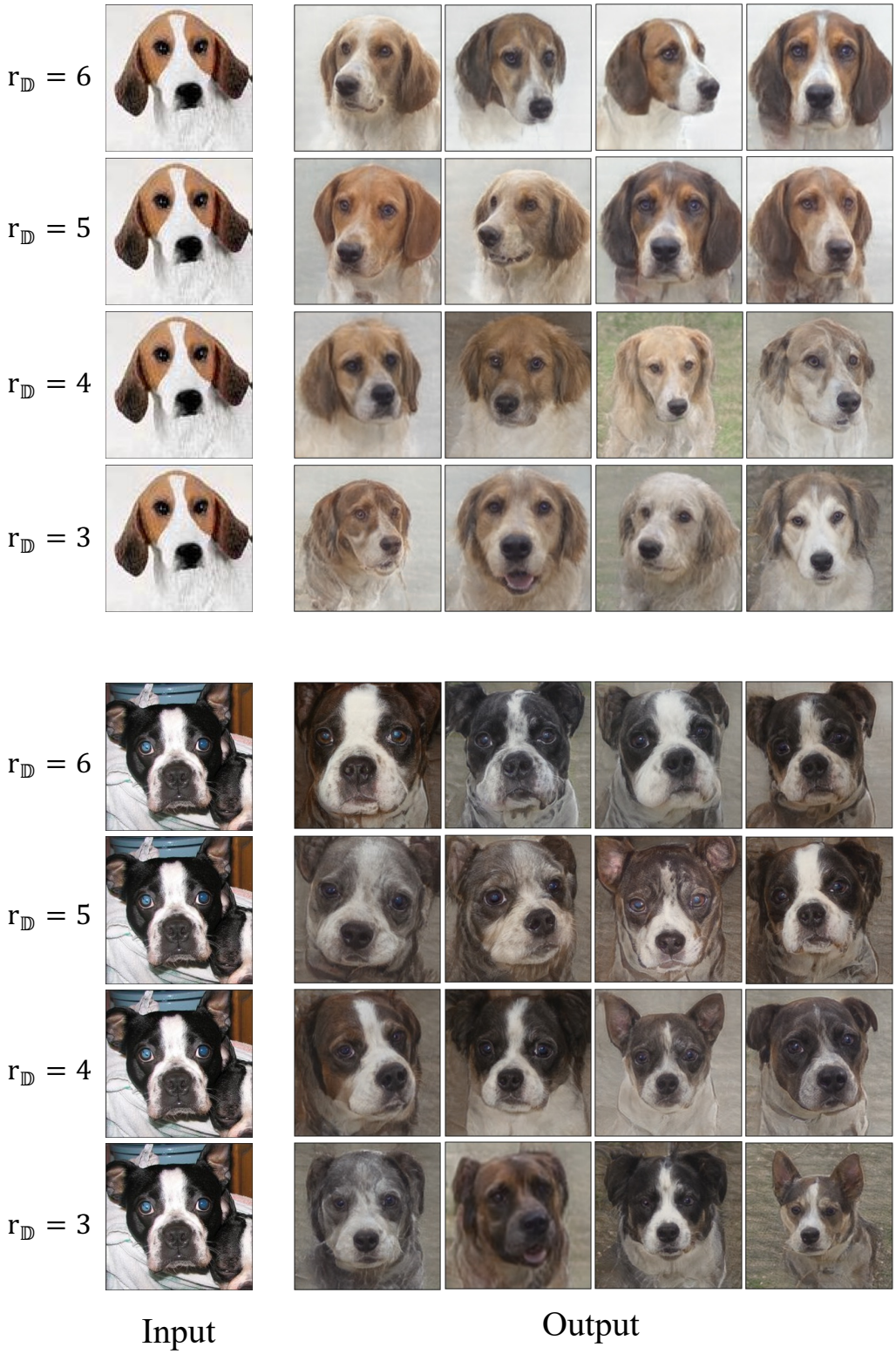

Figure 6: One-shot image generation by HAE on different radii on Animal Faces.

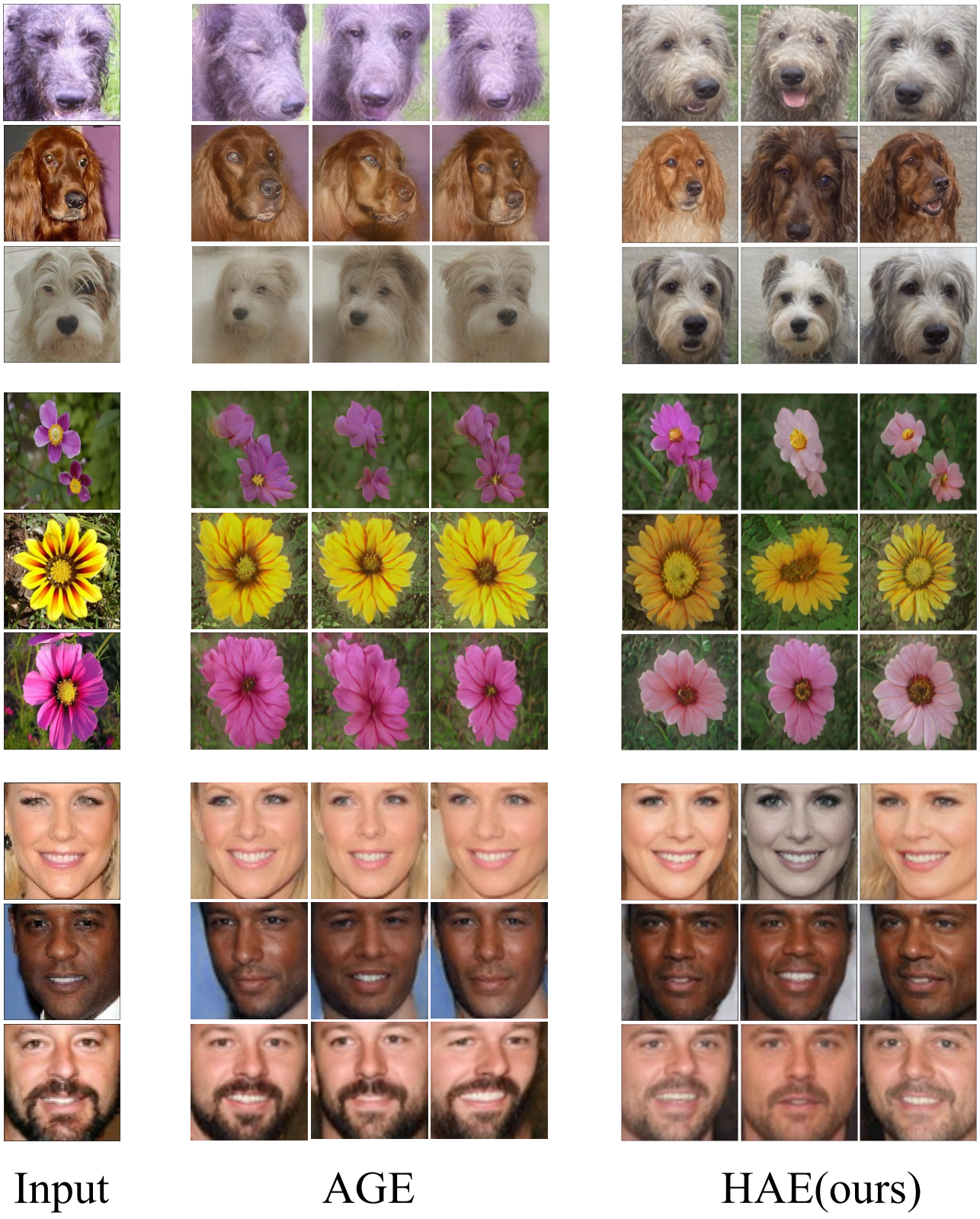

Figure 7: Comparison between images generated by AGE and HAE on Animal Faces, Flowers, and VGGFaces.

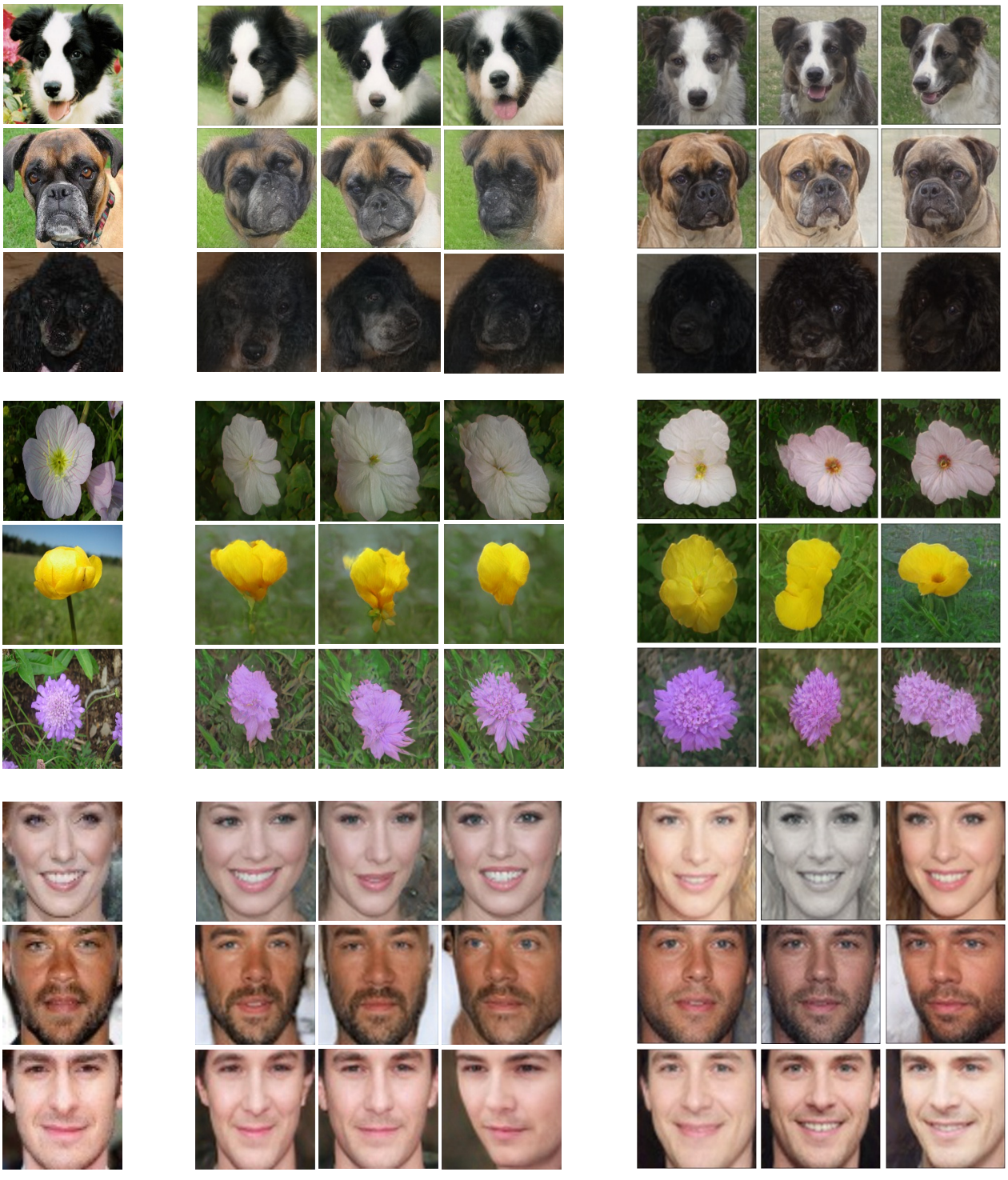

Figure 8: More examples used in the user study on three datasets.

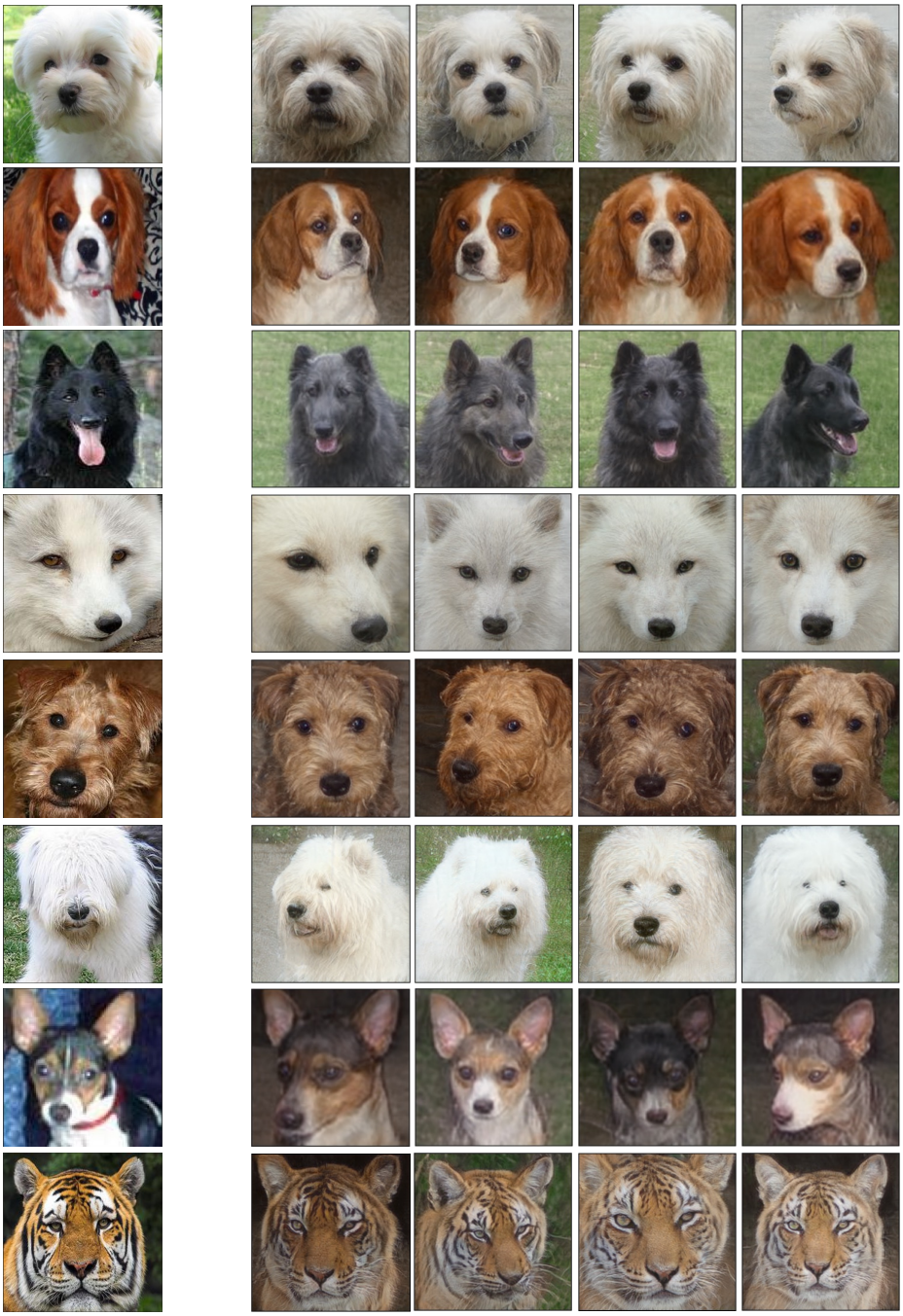

Figure 9: One-shot image generation by HAE on Animal Faces.

Figure 10: One-shot image generation by HAE on Flowers.

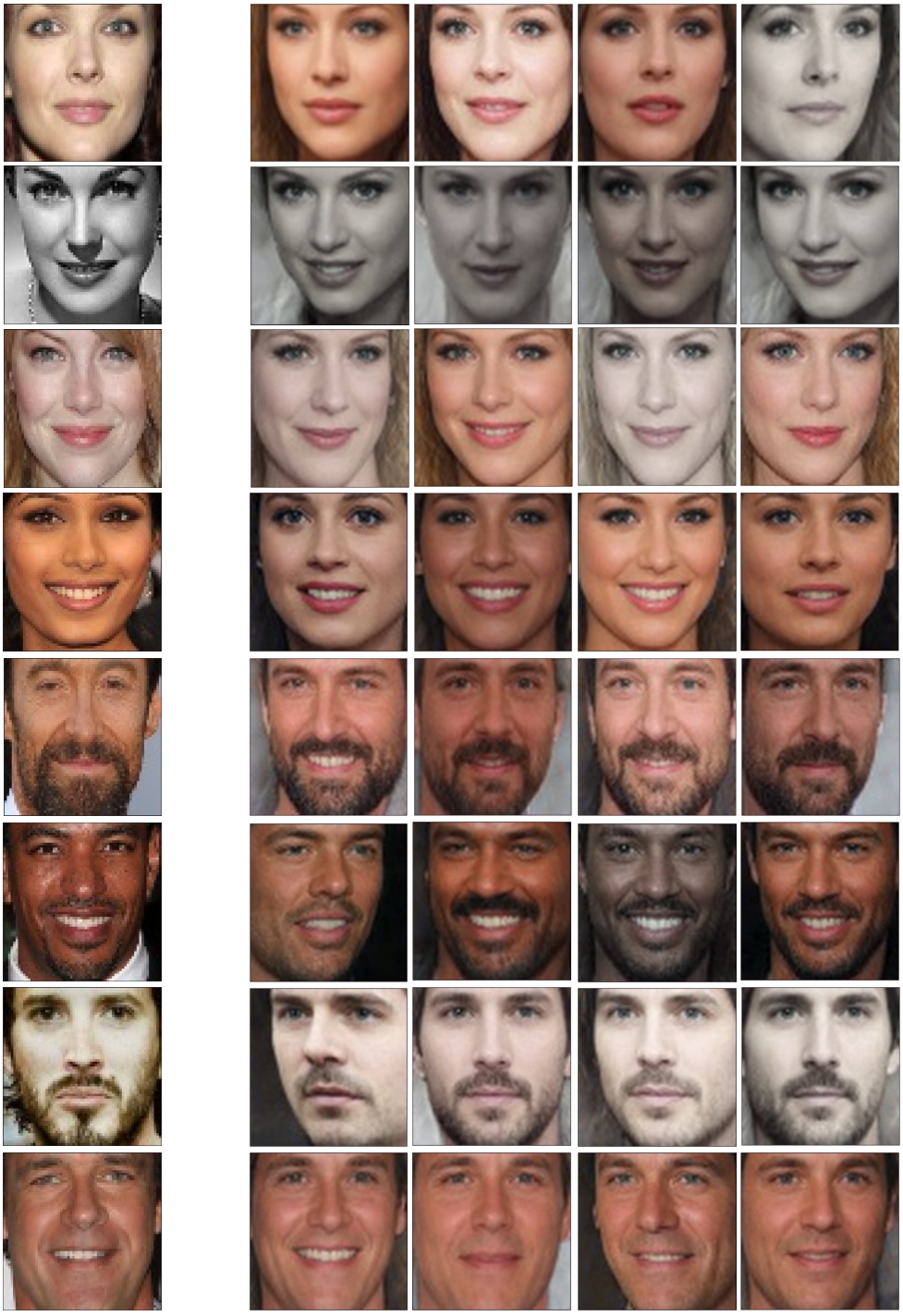

Figure 11: One-shot image generation by HAE on VGGFaces.