HYPDAE: Hyperbolic Diffusion Autoencoders for Hierarchical Few-shot Image Generation

Supplementary Material

Overview

This appendix is organized as follows:

Sec. A gives more implementation details of **HypDAE**. Sec 3.2 & Sec 4.1

Sec. B gives detail explanation of diffusion models.

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

Sec. D shows more results of the ablation study of **HypDAE**. Sec 4.3

Sec. E shows more comparisons between the latent manipulation in hyperbolic and Euclidean space. Sec 4.3

Sec. F provides examples to show the exceptional out-of-distribution few-shot image generation ability. Sec 4.4

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

Sec. H compares the images generated by state-of-the-art few-shot image generation method, *i.e*. WaveGAN [17], HAE [8] and our methods **HypDAE**. Sec 4.4

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

Sec. J gives more examples generated by **HypDAE**. Sec 4.4

A. Implementation Details and Analysis

Stage I. As mentioned in Sec 3.2, this stage does not require class labels for the images. To promote diversity, we use only the CLIP image encoder's class token (dimension 1×1024) for a compact representation, aligning it with the CLIP text feature space via a 5-layer fully connected MLP following the same settings in [16] that inject features into the diffusion process through cross-attention to replace the text feature in the original stable diffusion model.

We choose Stable Diffusion V2.1 [13] as the base generator for the base generative model. We set the image resolution to 512×512. We choose the Adam optimizer and set the learning rate as $1\mathrm{e}{-5}$. During the training process, the pre-trained CLIP image encoder and SD V2.1 models are frozen, only the Transformer block for aligning features is trainable. Since the SD model is loaded during the training process, we use $2 \times$ NVIDIA A800 (80GB) GPUs for training, and the batch size is selected as 24 for each GPU. We train about $1\mathrm{e}5$ steps to get the model to converge on each dataset.

Stage II. This stage is the only stage that requires the class labels for given images to learn the hierarchical representation. Although class labels are required in Stage II, the model only needs a small number of labeled data for pre-training and pseudo labels can be predicted by CLIP as shown in Sec. A. Furthermore, we show exceptional out-of-distribution generation ability in Sec. F. For the hyperbolic encoder mentioned in Sec 3.2, we use a single-head 5-layer Transformer block to reduce the dimensionality of the Euclidean latent vector $\boldsymbol{c}$ from 1×1024 to 1×512, which is then mapped to hyperbolic space via an exponential map. A hyperbolic feed-forward layer [2] produces the final hierarchical representation $z_{\mathbb{D}}$:

$$\boldsymbol{c}_h = f^{\otimes_c}(\exp_{\mathbf{0}}^c(\mathrm{E}(\boldsymbol{c}))), \tag{1}$$

where E is the Transformer encoder and $f^{\otimes_c}$ is the Möbius translation of feed-forward layer f as the map from Euclidean space to hyperbolic space, denoted as *Möbius linear layer*. In order to perform multi-class classification on the Poincaré disk defined in Sec 3.1, one needs to generalize multinomial logistic regression (MLR) to the Poincaré disk defined in [2]. An extra linear layer needs to be trained for the classification, and the details on how to compute softmax probability in hyperbolic space are shown in Sec. C. As mentioned in Sec 3.1, the distance between points grows exponentially with their radius in the Poincaré disk. In order to minimize Eq. (5) in the main paper, the latent codes of fine-grained images will be pushed to the edge of the ball to maximize the distances between different categories while the embedding of abstract images (images have common features from many categories) will be located near the center of the ball. Since hyperbolic space is continuous and differentiable, we are able to optimize Eq. (5) with stochastic gradient descent, which learns the hierarchy of the images.

Then we train a Transformer decoder to project the hyperbolic latent code back to the CLIP image space with exact

reconstruction. In practice, this is achieved by firstly applying a logarithmic map followed by a Transformer decoder D:

$$\boldsymbol{c}' = \mathrm{D}(\log_{\mathbf{0}}^{c}(\boldsymbol{c}_h)). \tag{2}$$

and $\boldsymbol{c}'$ will be fed into the cross-attention layer of the stable diffusion model to reconstruct the image x'. We use a single-head 30-layer Transformer block as the Transformer decoder for Animal Faces [9], VGGFaces [11], FFHQ [5], and NABirds [15] since these datasets are relatively large. Therefore, a deeper network is needed to reconstruct the latent representation of these large datasets. However, for the Flowers dataset [10], the number of images is less than 10 thousand, which is not enough to train a deep neural network. As a consequence, we use a single-head 5-layer Transformer block as the Transformer decoder for Flowers which works well.

Fine-tuning on FFHQ. As we mentioned in Sec 4.2 in the main paper, we learn the hierarchy of human faces by training Stage II with VGGFaces first. However, we visualize the human faces with the FFHQ dataset. Note that the FFHQ dataset has no class labels. Therefore, we first use the VGGFaces dataset to learn a good prior of hierarchy among human faces images with supervision, then fine-tune the model with the reconstruction loss $\mathcal{L}_{\text{rec}}$ *only* to teach the model how to reconstruct images with high resolution but maintaining the hierarchical representation prior. The results show the great potential of our model to be fine-tuned on large-scale dataset without supervision.

In Stage II, only the CLIP image encoder is loaded during the training process. Besides, the CLIP image encoder is frozen, and only the lightweight Transformer encoder and decoder are trainable. We use $1 \times$ NVIDIA RTX 4090 (24GB) GPU for training, and the batch size is selected as 256. The λ in Eq. (7) in the main paper is selected as 0.1. We choose the AdamW [7] optimizer and set the learning rate as $1\mathrm{e}{-3}$. A linear learning rate scheduler is used with a step size equal to 5000, with a multiplier $\gamma = 0.5$. We train about $1\mathrm{e}5$ steps to get the model to converge on each dataset.

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).

Although training our model requires considerable computing resources as mentioned before, the runtime cost and resources required for the inference stage are affordable. Our model can inference on a single NVIDIA RTX 4090 GPU (24GB) thanks to our multi-stage training/inference since one does not need to load all models simultaneously.

Pseudo-Labeling. For Flowers, Animal Faces and NABirds, we utilize the CLIP ViT-B/32 model. Given an image (x), we extract its embedding using the CLIP image encoder. Similarly, we compute embeddings for a predefined set of class names (y_i) using the CLIP text encoder. The cosine similarity between the image embedding and each class embedding is computed as:

$$\mathrm{sim}(x, y_i) = \frac{f(x) \cdot g(y_i)}{|f(x)||g(y_i)|}, \tag{3}$$

where ($f(\cdot)$) and ($g(\cdot)$) denote the CLIP image and text encoders, respectively. The softmax function is applied to convert similarity scores into probabilities. The pseudo-label (y) is assigned as the class with the highest probability:

$$y = \underset{j}{\mathrm{argmax}} \frac{\exp(\mathrm{sim}(x, y_i))}{\sum_j \exp(\mathrm{sim}(x, y_j))}. \tag{4}$$

For the VGGFaces dataset, we employ the DeepFace framework with the VGG-Face architecture. DeepFace predicts pseudo-labels by comparing the face embedding of an image with embeddings of known identities in the dataset. Specifically, we construct a reference database by selecting one image per class, and each input image is assigned to the class with the highest similarity score. This approach ensures robust pseudo-labeling by leveraging DeepFace's face recognition capabilities.

Dataset Settings We evaluate our method on Animal Faces [9], Flowers [10], VGGFaces [11], FFHQ [5], and NABirds [15] following the settings described in [1, 8].

Animal Faces. We randomly select 119 categories as seen for training and leave 30 as unseen categories for evaluation.

Flowers. The Flowers [10] dataset is split into 85 seen categories for training and 17 unseen categories for evaluation.

VGGFaces. For VGGFaces [11], we randomly select 1802 categories for training and 572 for evaluation.

NABirds. For NABirds [15], 444 categories are selected for training and 111 for evaluation.

FFHQ. Due to the low resolution (64×64) of the VGGfFces dataset, we use FFHQ [5] to fine-tune the model pre-trained on VGGFaces without supervision and visualize images of human faces with FFHQ.

B. Additional Background - Diffusion Models

Diffusion Denoising Probabilistic Models (DDPM) [4] are generative latent variable models that aim to model a distribution $p_\theta(x_0)$ that approximates the data distribution $q(x_0)$ and easy to sample from. DDPMs model a "forward process" in the space of x_0 from data to noise. This is called "forward" due to its procedure progressing from x_0 to x_T. Note that this process is a Markov chain starting from x_0, where we gradually add noise to the data to generate the latent variables $x_1, \ldots, x_T \in X$. The sequence of latent variables, therefore, follows $q(x_1, \ldots, x_t \mid x_0) = \prod_{i=1}^{t} q(x_t \mid x_{t-1})$, where a step in the forward process is defined as a Gaussian transition $q(x_t \mid x_{t-1}) := N(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$ parameterized by a schedule $\beta_0, \ldots, \beta_T \in (0, 1)$. When T is large

enough, the last noise vector x_T nearly follows an isotropic Gaussian distribution.

An interesting property of the forward process is that one can express the latent variable x_t directly as the following linear combination of noise and x_0 without sampling intermediate latent vectors:

$$x_t = \sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}w, \;\; w \sim N(0, I), \tag{5}$$

where $\alpha_t := \prod_{i=1}^{t}(1-\beta_i)$.

To sample from the distribution $q(x_0)$, we define the dual "reverse process" $p(x_{t-1} \mid x_t)$ from isotropic Gaussian noise x_T to data by sampling the posteriors $q(x_{t-1} \mid x_t)$. Since the intractable reverse process $q(x_{t-1} \mid x_t)$ depends on the unknown data distribution $q(x_0)$, we approximate it with a parameterized Gaussian transition network $p_\theta(x_{t-1} \mid x_t) := N(x_{t-1} \mid \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$. The $\mu_\theta(x_t, t)$ can be replaced [4] by predicting the noise $\epsilon_\theta(x_t, t)$ added to x_0 using equation 5.

C. Hyperbolic Neural Networks

For hyperbolic spaces, since the metric is different from Euclidean space, the corresponding calculation operators also differ from Euclidean space. In this section, we start by defining two basic operations: Möbius addition and Möbius scalar multiplication [6], given fixed curvature c.

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

$$x \oplus_c y = \frac{\left(1 - 2c\langle x, y\rangle - c\|y\|_2^2\right)x + \left(1 + c\|x\|_2^2\right)y}{1 - 2c\langle x, y\rangle + c^2\|x\|_2^2\|y\|_2^2}, \tag{6}$$

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:

$$r \otimes_c x = \tan_c\left(r \tan_c^{-1}\left(\|x\|_2\right)\right)\frac{x}{\|x\|_2}. \tag{7}$$

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

$$\exp_x^c(v) := x \oplus_c \left(\tanh\left(\sqrt{c}\frac{\lambda_x^c\|v\|}{2}\right)\frac{v}{\sqrt{c}\|v\|}\right). \tag{8}$$

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

$$\log_x^c(y) := \frac{2}{\sqrt{c}\lambda_x^c}\operatorname{arctanh}\left(\sqrt{c}\,\|-x \oplus_c y\|\right)\frac{-x \oplus_c y}{\|-x \oplus_c y\|}. \tag{9}$$

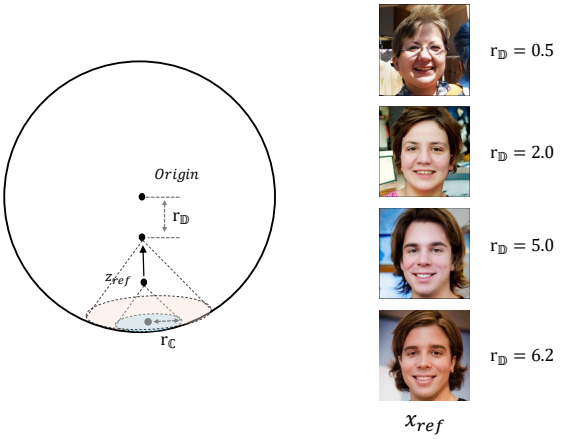

Figure 1. **The illustration of hierarchical data sampling in hyperbolic space**

We also provide the formula to calculate the softmax probability in hyperbolic space used in Eq. (5) in the main paper: Given K classes and $k \in \{1, \ldots, K\}, p_k \in \mathbb{D}_c^n, a_k \in T_{p_k}\mathbb{D}_c^n \backslash \{\mathbf{0}\}$:

$$p(y = k \mid x) \propto \exp\left(\frac{\lambda_{p_k}^c\,\|a_k\|}{\sqrt{c}}\sinh^{-1}\left(\frac{2\sqrt{c}\,\langle -p_k \oplus_c x, a_k\rangle}{\left(1 - c\,\|-p_k \oplus_c x\|^2\right)\|a_k\|}\right)\right), \quad \forall x \in \mathbb{D}_c^n, \tag{10}$$

where $\oplus_c$ denotes the Möbius addition defined in Eq. (6) with fixed sectional curvature of the space, denoted by c.

Hierarchical Data Sampling. As illustrated in Fig. 1, as the latent code z_{ref} of the reference image x_{ref} moves from the edge to the center of the Poincaré disk, the control of the identity of the sampled images becomes weaker and weaker. The identity of the generated images becomes more ambiguous. To conduct hierarchical image sampling in hyperbolic space, one can move the latent code z_{ref} of the reference image x_{ref} towards the origin of the Poincaré disk. Then, sampling latent points among the "children" of the rescaled reference images. In practice, the semantic diversity of the generated images can be controlled either by setting different values of $r_{\mathbb{D}}$, then calculating $r_{\mathbb{C}}$ based on $r_{\mathbb{D}}$, or by setting different values of $r_{\mathbb{C}}$ directly.

Recall that, in hyperbolic space, the shortest path with the induced distance between two points is given by the geodesic defined in Eq. (2) in the main paper. The geodesic equation between two embeddings $z_{\mathbb{D}\mathbf{i}}$ and $z_{\mathbb{D}\mathbf{j}}$, denoted by $\gamma_{z_{\mathbb{D}i} \to z_{\mathbb{D}j}}(t)$, is given by

$$\gamma_{z_{\mathbb{D}i} \to z_{\mathbb{D}j}}(t) = z_{\mathbb{D}i} \oplus_c t \otimes_c \left((-z_{\mathbb{D}i}) \oplus_c z_{\mathbb{D}j}\right), \; t \in [0, 1], \tag{11}$$

where $\oplus_c$ denotes the Möbius addition with aforementioned sectional curvature c. Therefore, for hierarchical data sampling, we can first define the value of $r_{\mathbb{C}}$, then sample random data points in hyperbolic space. If the distance between the sampled data point and z_{ref} is equal to or shorter than the value, we accept the data point. Otherwise, we move the latent codes along the geodesic between the sampled data point and z_{ref} until the distance is within the scope we define. We show more hierarchical image sampling examples in Sec. F.

D. Ablation Study

There are a few hyperparameters of **HypDAE** that control the generation quality and diversity. We conduct ablation studies on each of them in this section.

Hyperbolic Radius. By varying the radii of "parent" images, **HypDAE** controls the semantic diversity of generated images (Fig. 11), where the "parent" images can be viewed as the image with the average attributes of its children. The quantitative results of the Flowers dataset are presented in Tab. 1. We can see that the diversity increases as the radius becomes smaller, therefore, the value of LPIPS increases accordingly. However, changing too many attributes changes the identity or category of the given images, therefore, the FID decreases when the radius is smaller than 5.5. In practice, we select 5.5 as the radius of the parent images for few-shot image generation.

Classifier-free Guidance. To achieve the trade-off between identity preservation and image harmonization, we find that classifier-free sampling strategy [3] is a powerful tool. Previous work [14] found that the classifier-free guidance is actually the combination of both prior and posterior constraints. In our experiments, we follow the settings in [18].

$$\epsilon_{\text{prd}} = \epsilon_{\text{uc}} + s\left(\epsilon_{\text{c}} - \epsilon_{\text{uc}}\right), \tag{12}$$

where ϵ_{prd}, ϵ_{uc}, ϵ_{c}, s are the model's final output, unconditional output, conditional output, and a user-specified weight, respectively. The visualizations are shown in Fig. 3, and quantitative results for the Flowers dataset are presented in Tab. 2. Consistent with findings in Tab. 1, diversity, measured by LPIPS, increases as the cfg scale grows. However, excessive cfg scaling can alter the identity or category of the input images, leading to a decline in FID when the cfg scale exceeds 1.3. Based on these results, we select a cfg scale of 1.3 to achieve optimal few-shot image generation with a balance between fidelity and diversity.

Encoding Strength of the Stochastic Encoder. As described in Sec. 3.2, the encoding strength of the stochastic encoder determines the extent of information encoded from the given images. For instance, attributes such as rough posture, color, and style are encoded during the early steps of the diffusion process. A higher encoding strength deconstructs more information from the input images, while a lower encoding strength retains more original information. An encoding strength of 1 implies full deconstruction, where the initial latent of the denoising process is Gaussian noise. Conversely, an encoding strength of 0 results in exact reconstruction without information loss.

Hyp Radius	6.2	6.0	5.5	5.0	4.5
FID(↓)	27.89	24.67	**23.96**	24.89	26.63
LPIPS(↑)	0.7585	0.7589	0.7595	0.7643	**0.7725**

Table 1. **Ablation study** of different radii on Flowers.

CFG	1.0	1.1	1.3	1.5	1.7
FID(↓)	25.52	24.89	**23.96**	26.04	25.63
LPIPS(↑)	0.7391	0.7534	0.7595	0.7660	**0.7737**

Table 2. **Ablation study** of the influence of CFG on Flowers.

Strength	1.0	0.98	0.95	0.9	0.8
FID(↓)	28.97	24.59	**23.96**	24.94	26.48
LPIPS(↑)	**0.7631**	0.7606	0.7595	0.7585	0.7375

Table 3. **Ablation study** of the influence of encoding strength on Flowers.

While lower encoding strength preserves the identity and style of the input images, it reduces diversity. This trade-off is visualized in Fig. 4, and quantitative results for the Flowers dataset are presented in Tab. 3. Consistent with Tab. 1, diversity, measured by LPIPS, increases with higher encoding strength, while excessive encoding strength can cause changes in identity or category. This is reflected in a decrease in FID when encoding strength exceeds 0.95 (*i.e.*, 5% of the information is encoded). Based on these findings, we set the encoding strength of the stochastic encoder to 0.95 to achieve reliable few-shot image generation.

Hyperparameter Ablation. To validate the robustness of the trade-off parameter in Eq. (7), we rewrite Eq. (7) as: $\mathcal{L} = \lambda \cdot \mathcal{L}_{\text{hyper}} + \mathcal{L}_{\text{rec}}$, ablate the loss to analyze this trade-off. As shown in Fig. 2, increasing λ improves semantic consistency (lower FID) while reducing diversity (lower LPIPS), validating the controllability introduced by the hyperbolic component.

E. Comparison with Euclidean space

In this section, we present a detailed comparison of different latent spaces, as shown in Fig. 5. Compared to classical Euclidean space, hyperbolic space enables smoother transitions between two given images. In hyperbolic space, identity-irrelevant features transition first, followed by a gradual change in identity-relevant features. In contrast, Euclidean

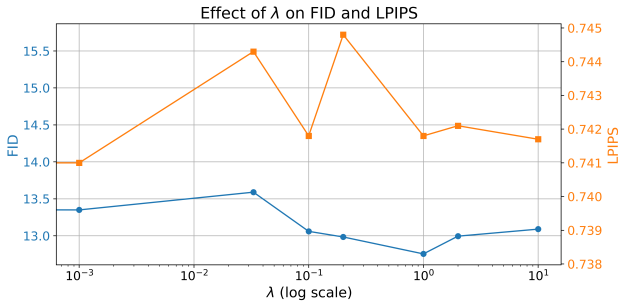

Figure 2. **Ablation Study** of trade-off adaptive hyperparameter λ in the loss function.

space exhibits simultaneous changes in both identity-relevant and identity-irrelevant features, leading to less structured transitions.

These results confirm that our method effectively learns hierarchical representations in hyperbolic space, enabling few-shot image generation by selectively modifying category-irrelevant features—a capability that Euclidean space cannot achieve. Additional interpolation results, provided in Fig. 6, demonstrate that smooth and distortion-free transitions are achievable in hyperbolic space. These findings highlight that **HypDAE** enables precise geodesic and hierarchical control during editing, offering a significant advantage over traditional approaches.

F. Out-of-distribution Few-shot Image Generation

In Sec 4.2 of the main paper, we mentioned we fine-tuned the model trained with VGGFaces using the FFHQ dataset. The model shows exceptional out-of-distribution generalization ability on the FFHQ dataset. To further verify the OOD generalization ability of **HypDAE**, we select two images for Animal Faces [9], Flowers [10], and NABirds [15] datasets with three styles from DomainNet [12] including " painting", "sketch", "quick draw", and "clipart" styles where are model never seen during the training stage. The OOD style transfer can be done by slightly increasing the encoding strength of the stochastic encoder to capture more style information of the given new images. The results in Fig. 7 Fig. 8 Fig. 9 Fig. 10 show that our proposed method has exceptional OOD generalization ability even for new domains with a big gap from the original domain. Although our model still generates images with some real detail for the style "clipart", the performance in other styles is satisfying. Such an OOD generalization ability is significantly better than any of the previous works.

G. Hierarchical Image Generation

In this section, we provide additional examples of images generated by **HypDAE** at varying radii in the Poincaré disk. As illustrated in Fig. 11, Fig. 12, and Fig. 13, high-level, category-relevant attributes remain unchanged when the radius is large, allowing for the generation of diverse images within the same category. Conversely, as the hyperbolic radius $r_{\mathbb{D}}$ decreases, the generated images become more abstract and semantically diverse. Moving closer to the center of the Poincaré disk results in the gradual loss of fine-grained details and changes to higher-level attributes.

For the few-shot image generation task, larger radii are optimal as they allow for the modification of category-irrelevant attributes while preserving the category identity. However, **HypDAE** is not limited to few-shot image generation and shows significant potential for other downstream applications. For example, **HypDAE** can generate a diverse set of feline images from a single cat image. This is achieved by scaling the latent code to a smaller radius in hyperbolic space and introducing random perturbations to approximate the average latent code for various feline categories. Finally, fine-grained and diverse feline images are generated by moving these average codes outward to larger radii, representing their "children" in the hierarchical space.

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

We compare images generated by state-of-the-art methods, including WaveGAN [17], HAE [8], and our proposed method, across four datasets. As shown in Fig. 14, WaveGAN produces high-fidelity images, but the diversity is limited (*i.e.*, blending features from two input images without significant variation). HAE improves diversity but suffers from low fidelity and quality, with missing details and changes in category or identity compared to the original images. In contrast, **HypDAE** achieves an excellent balance between maintaining identity and enhancing diversity while delivering significantly higher image quality than other methods. These results highlight the potential of **HypDAE** for broader applications in future downstream tasks.

I. 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 **HypDAE** with three other models WaveGAN [17], HAE [8], and SAGE [1]:

1. We randomly chose 5 images from four datasets, and for each image, we then generated 3 variants in 1-shot setting (WaveGAN used 2-shot setting), respectively. Overall, there were 20 original images and 60 generated variants in total.
2. For each sample of each model, we present one masked background image, a reference object, and the generated image to annotators. We then shuffled the orders for all images.

Figure 3. **Ablation study** on the influence of classifier free guidance.

3. We recruited 30 volunteers from diverse backgrounds and provided detailed guidelines and templates for evaluation. Annotators rated the images on a scale of 1 to 4 across three criteria: "Fidelity", "Quality", and "Diversity". "Fidelity" evaluates identity preservation, while "Quality" assesses quality of the images (*e.g.*, details of the image). "Diversity" measures variation among generated proposals to discourage "copy-paste" style outputs.

The user-study interface is shown in Fig. 17.

J. Additional Examples Generated by HypDAE

Finally, we provide more examples generated by **HypDAE** in Fig. 15 and Fig. 16 for four datasets. The results show that our method achieves a balance between the quality and diversity of the generated images which significantly outperforms previous methods.

Figure 4. **Ablation study** on the influence of the encoding strength of the stochastic encoder on FFHQ (strength equals 1 means x_0 is fully deconstructed, *i.e.*, x_T is a Gaussian noise).

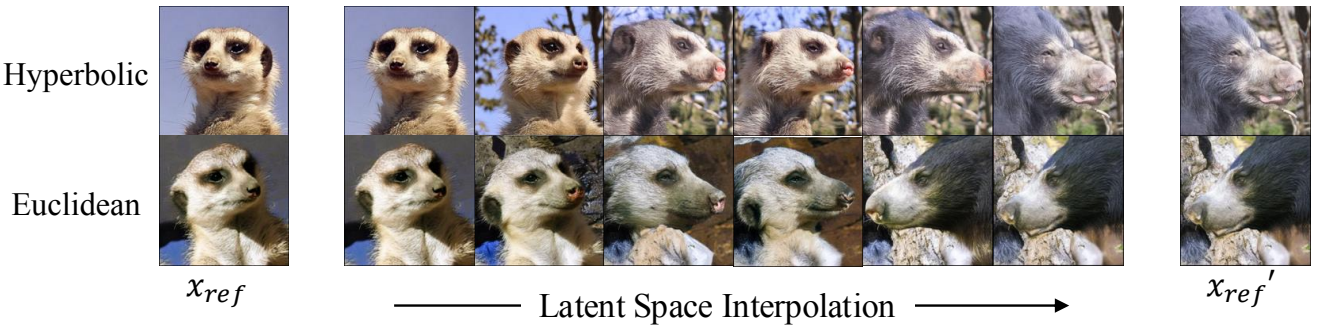

Figure 5. **Comparison of interpolation in hyperbolic space and Euclidean space** on Animal Faces dataset.

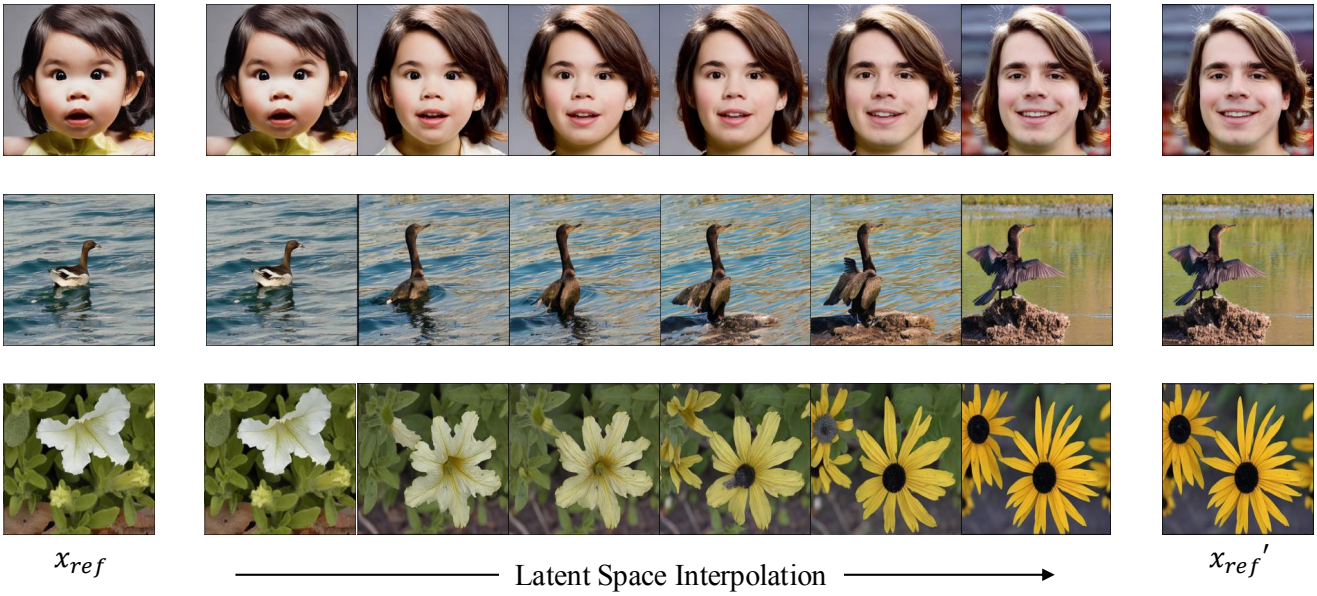

Figure 6. **More results of interpolation in hyperbolic space** on FFHQ, NABirds, and Flowers datasets.

Figure 7. Few-shot image generation on **out-of-distribution examples in painting style**.

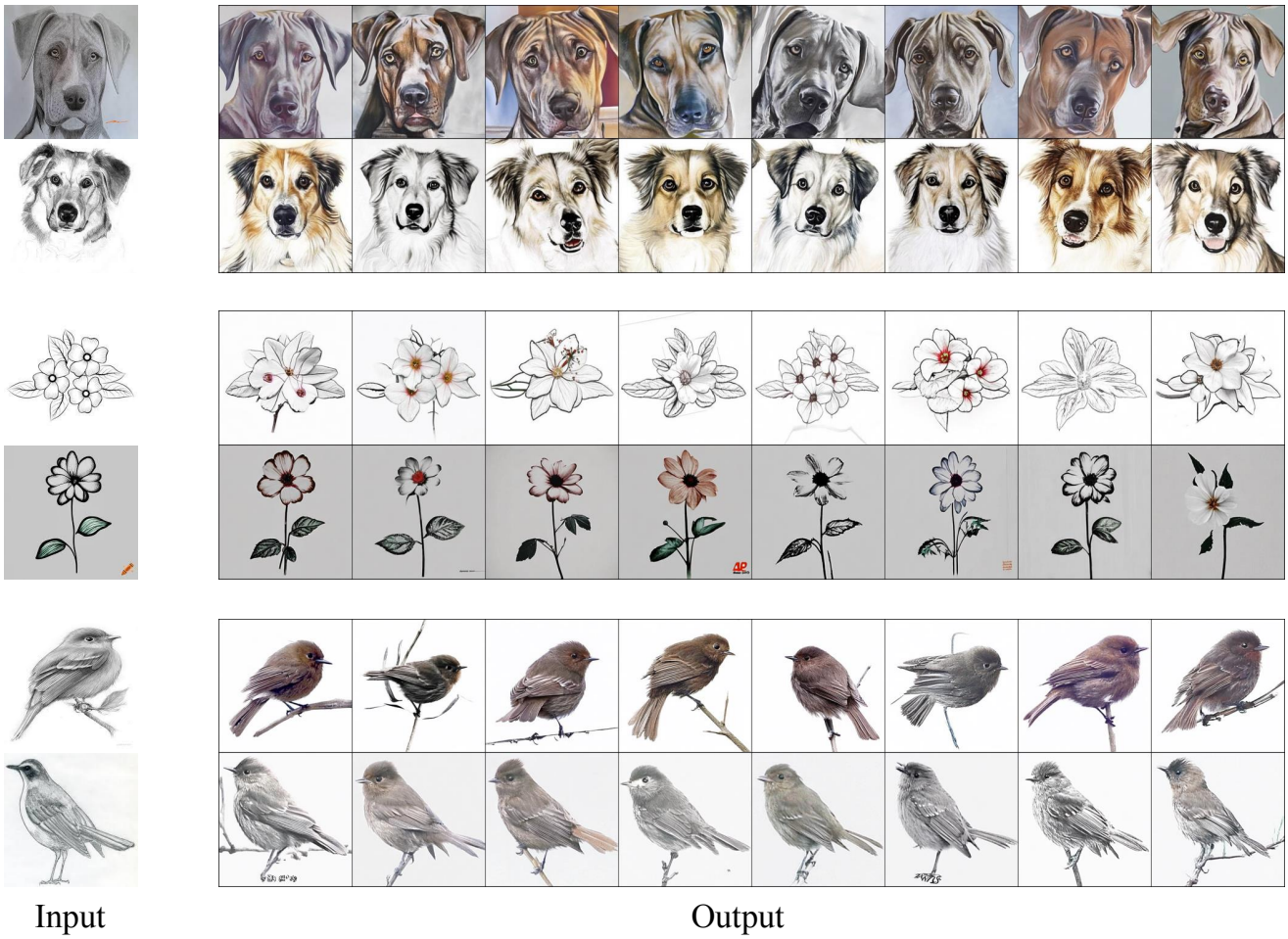

Figure 8. Few-shot image generation on **out-of-distribution examples in sketch style**.

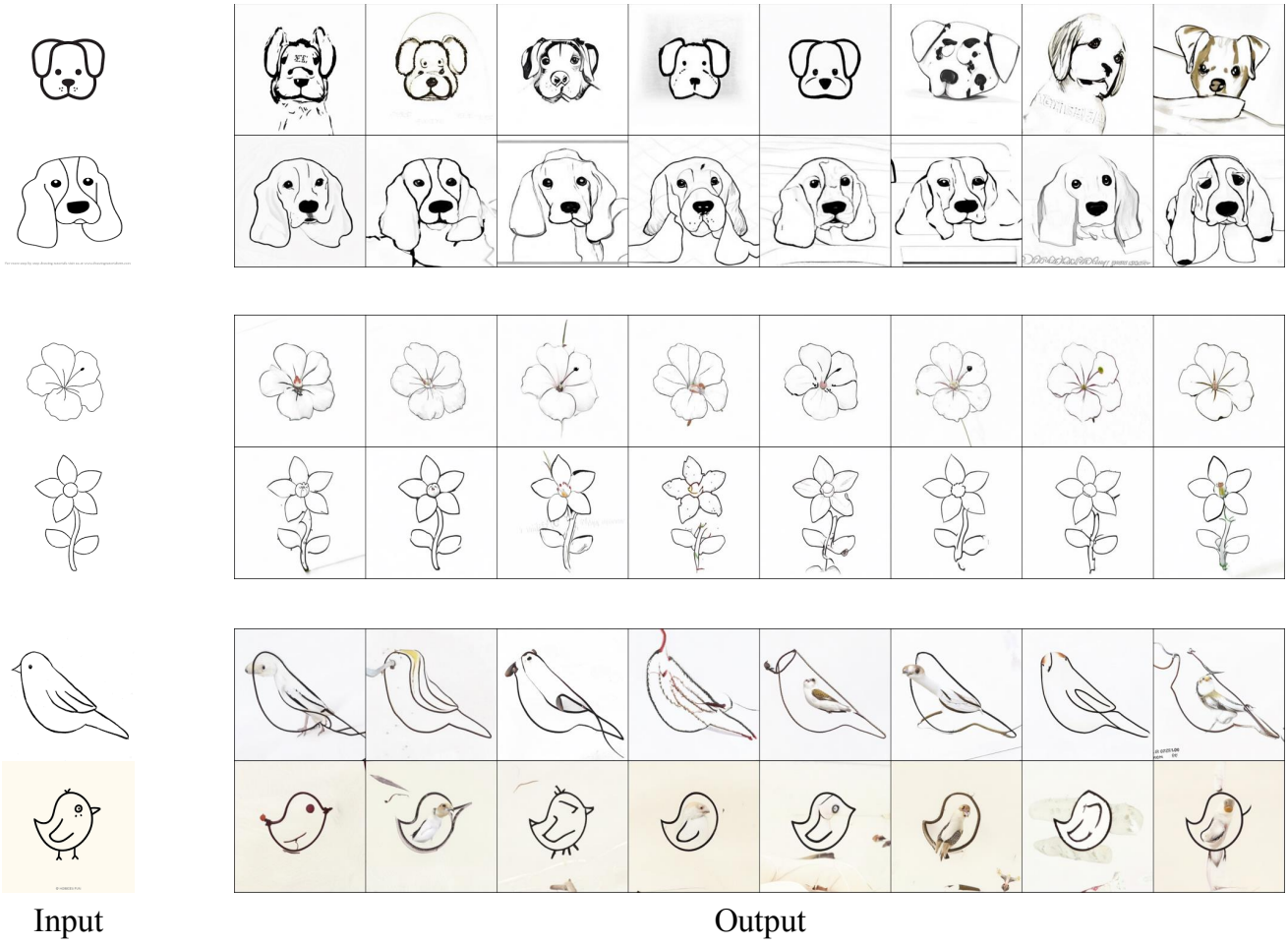

Figure 9. Few-shot image generation on **out-of-distribution examples in quick draw style**.

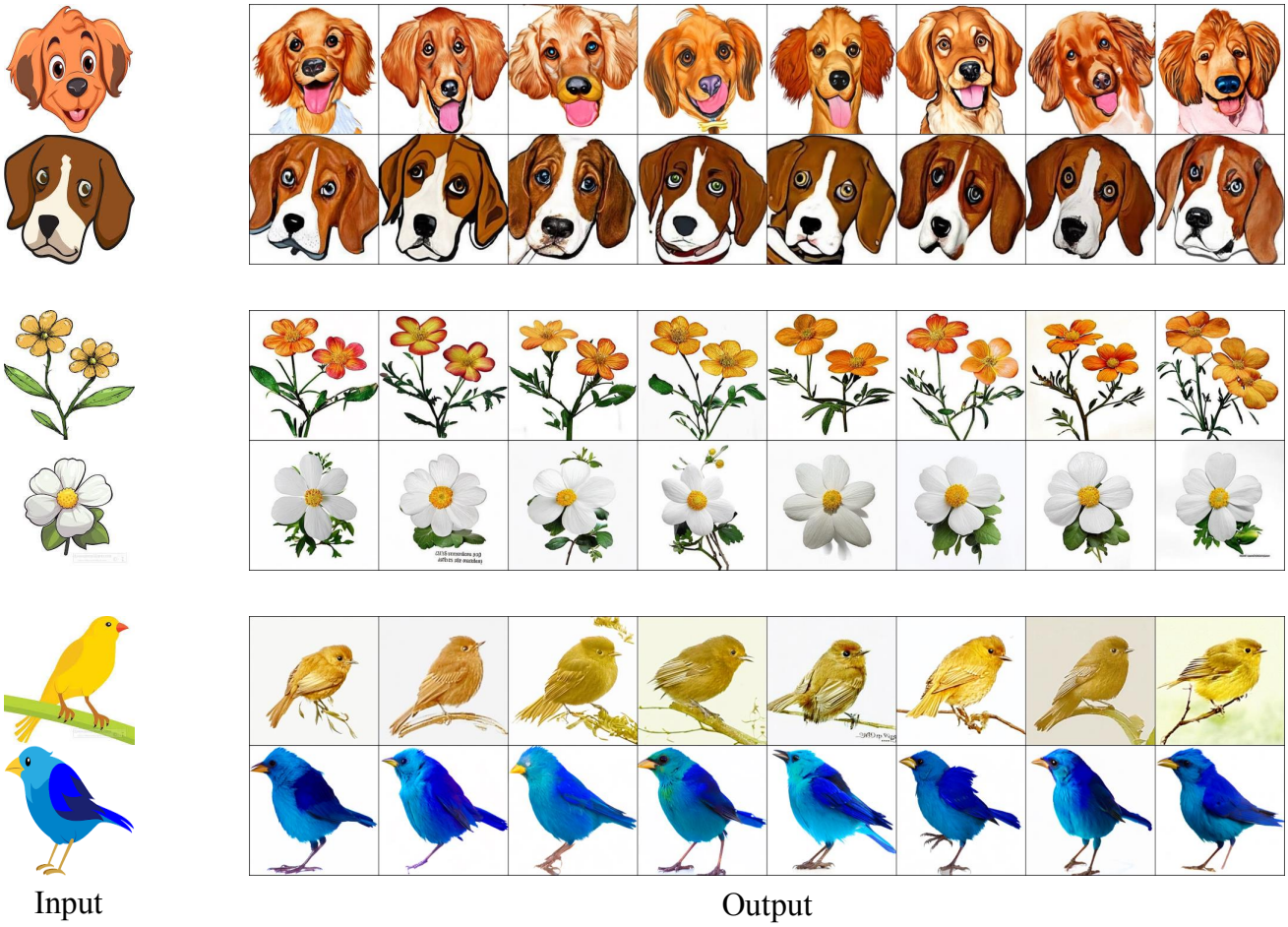

Figure 10. Few-shot image generation on **out-of-distribution examples in clipart style**.

Figure 11. **Images with hierarchical semantic similarity generated by HypDAE**.

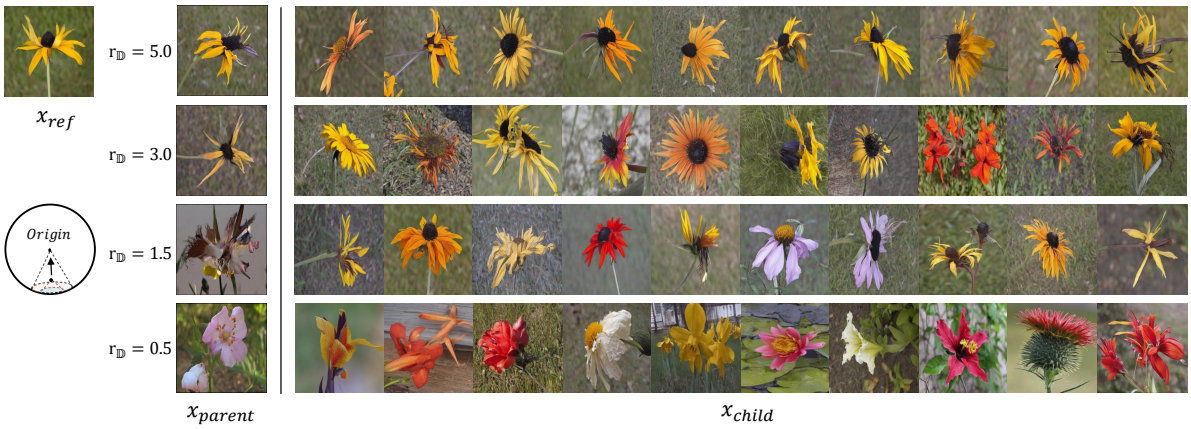

Figure 12. **Images with hierarchical semantic similarity generated by HypDAE**.

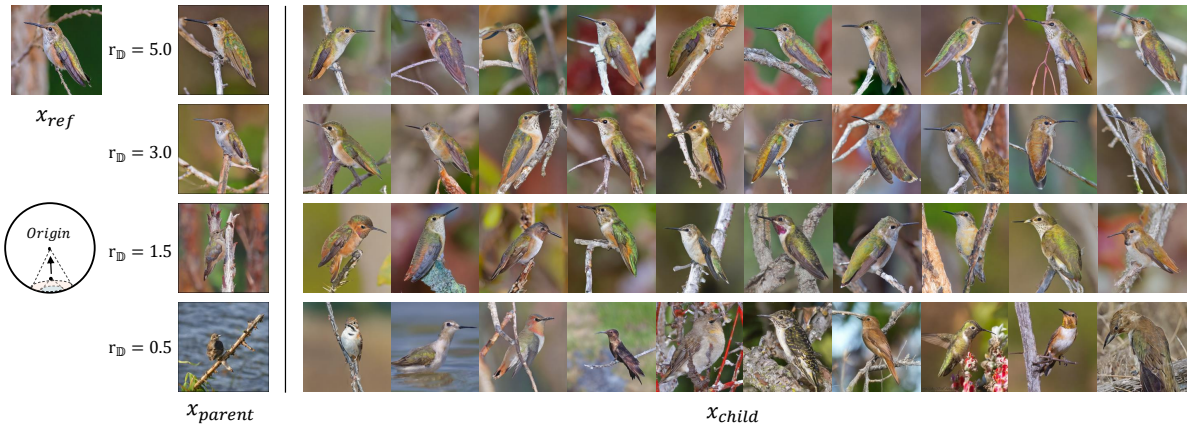

Figure 13. **Images with hierarchical semantic similarity generated by HypDAE**.

Figure 14. **More comparison between images generated by WaveGAN, HAE, and HypDAE** on Flowers, Animal Faces, VGGFaces, and NABirds. Note: WaveGAN uses a 2-shot setting; HAE and **HypDAE** are both in a 1-shot setting. **Zoom in to see the details.**

Figure 15. **More examples generated by HypDAE** on Animal Faces and Flowers.

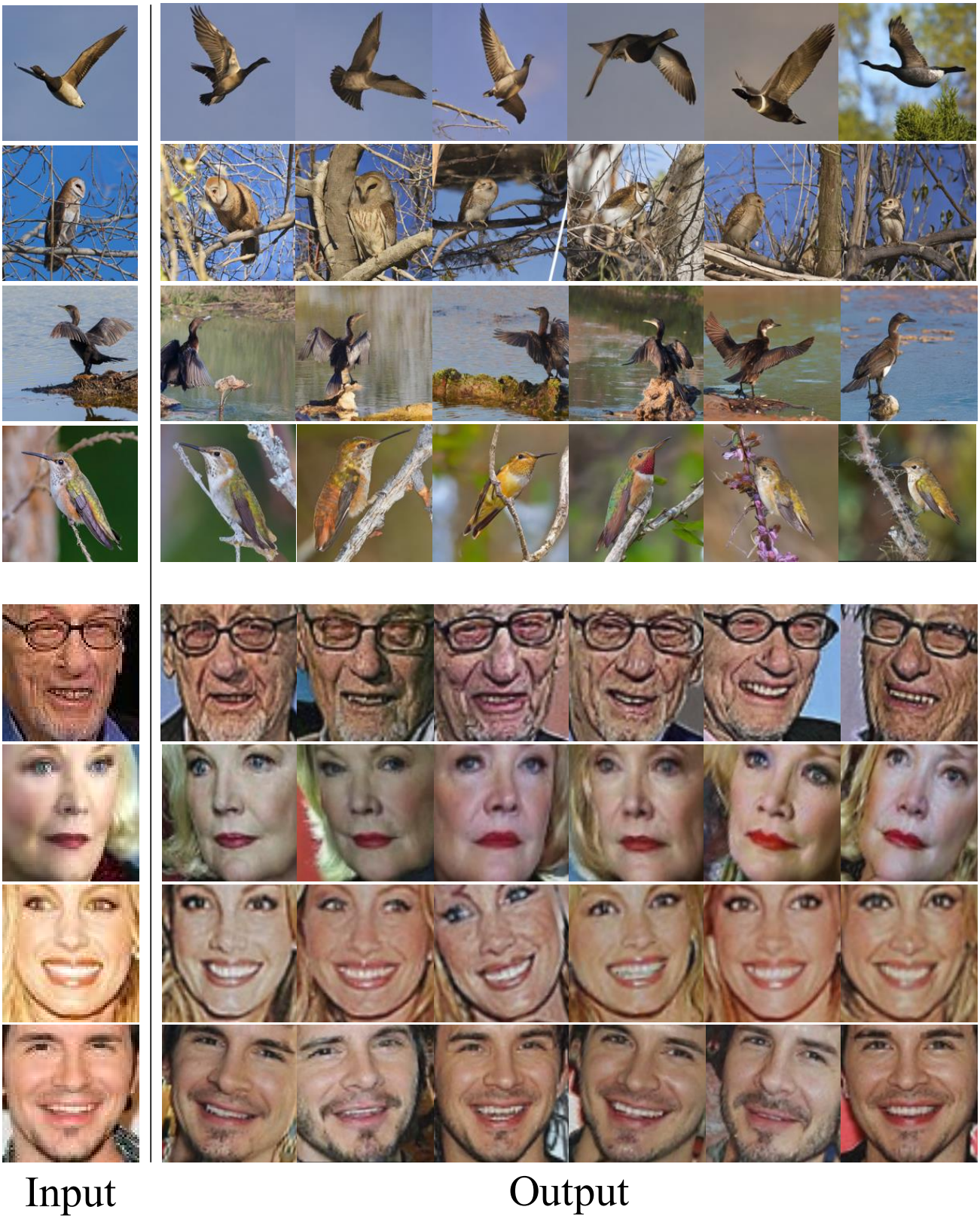

Figure 16. **More examples generated by HypDAE** on NABirds and VGGFaces.

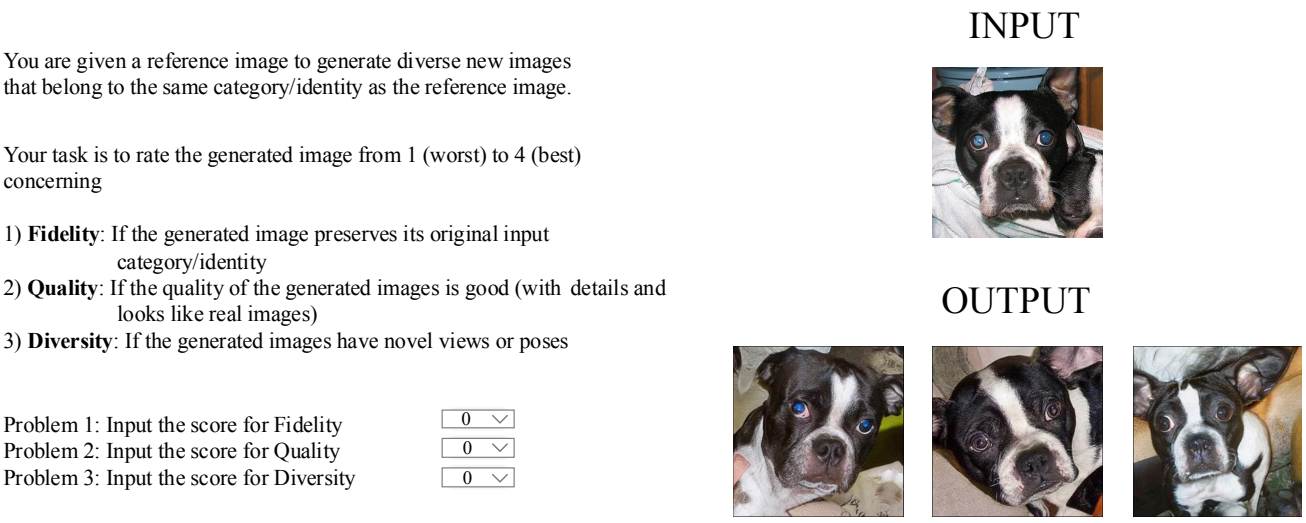

Figure 17. **The illustration of the user study interface.**

References

[1] Guanqi Ding, Xinzhe Han, Shuhui Wang, Xin Jin, Dandan Tu, and Qingming Huang. Stable attribute group editing for reliable few-shot image generation. *arXiv preprint arXiv:2302.00179*, 2023. 2, 5

[2] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. In *NeurIPS*, pages 5345–5355, 2018. 1

[3] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NIPS Workshop*, 2022. 4

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, pages 6840–6851, 2020. 2, 3

[5] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4217–4228, 2019. 2

[6] Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *CVPR*, pages 6417–6427, 2020. 3

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 2

[8] Lingxiao Li, Yi Zhang, and Shuhui Wang. The euclidean space is evil: Hyperbolic attribute editing for few-shot image generation. In *ICCV*, pages 22714–22724, 2023. 1, 2, 5

[9] 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. 2, 5

[10] 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. 2, 5

[11] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015. 2

[12] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019. 5

[13] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1

[14] Zhicong Tang, Shuyang Gu, Jianmin Bao, Dong Chen, and Fang Wen. Improved vector quantized diffusion models. *arXiv:2205.16007*, 2022. 4

[15] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *CVPR*, 2015. 2, 5

[16] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *CVPR*, pages 18381–18391, 2023. 1

[17] Mengping Yang, Zhe Wang, Ziqiu Chi, and Wenyi Feng. Wavegan: Frequency-aware gan for high-fidelity few-shot image generation. In *ECCV*, pages 1–17. Springer, 2022. 1, 5

[18] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *CVPR*, pages 3813–3824, 2023. 4