A. Algorithms

We list the training and translation (sampling) strategy in Algorithm 1 and Algorithm 2, respectively.

Training To learn the latent energy-based model $E_{x\to y}$, we take the latent codes z_y of the target domain $P_{\mathcal{Y}}$ as our ground truth data. The latent codes z_x of the source domain $P_{\mathcal{X}}$ serve as the initial samples of the latent MCMC as shown in Eq. 6. The training algorithm follows:

```
Algorithm 1: Latent Energy-based Model Training
  Input: source domain $P_{\mathcal{X}}$, target domain $P_{\mathcal{Y}}$
  Output: latent energy function $E_{x\to y}$
  while not converged do
      # Draw source and target domain image $x$ and $y$
      $x \sim P_{\mathcal{X}},\ y \sim P_{\mathcal{Y}}$
      # Encode sample $\tilde{z}_y^0$ and target $z_y$
      $\tilde{z}_y^0 = z_x = Enc(x)\ , z_y = Enc(y)$
      # MCMC to sample $\tilde{z}_y^T$
      for $t = 1 : T$ do
          Update $\tilde{z}_y^t$ according to Eq. 6
      end
      # Update $E_{x\to y}$ based on $\tilde{z}_y^T$ and $z_y$
      Update $E_{x\to y}$ according to Eq.2
  end
```

Translation Given an input image, the translation process is simple.

```
Algorithm 2: Latent Energy Transport for Translation
  Input: $x$
  Output: $y$
  $z_y^0 = z_x = Enc(x)$
  for $t = 1 : T$ do
      Update $z_y^t$ according to Eq. 6
  end
  $y = Dec(z_y^T)$
```

B. β-VAE

We adopt the open-source code in https://github.com/1Konny/Beta-VAE. We keep all the settings the same but set the latent dimension at 32. We construct the latent EBM as an one-hidden-layer MLP (32-64-1) and use LeakyReLU for activation. We use SGD for optimization with learning rate 0.1. The MCMC sampler is ran for 10 steps and the step size is 0.1. More results are given in Figure 10 and Figure 11.

C. ALAE

We adopt the open-source code in https://github.com/podgorskiy/ALAE and keep all the settings the same. Implementation details have been given in the main part. More results are given in Figure 12.

Evaluation protocol: For FID evaluations, we follow the protocol in Appendix C of StarGAN v2, and the public code can be found at github.com/clovaai/stargan-v2. Specifically, FID is calculated between translated test images and training images. We report the average FID of each pair of domains. For KID evaluation, we adopt the source code from github.com/taki0112/GAN_Metrics-Tensorflow, which has also been used in CF-EBM.

D. VQ-VAE-2

We adopt the open-source code in https://github.com/rosinality/vq-vae-2-pytorch. We keep all the settings the same but set the codebook dimesnion at 32 and codebook size at 256. In Table 8, we evaluate the reconstruction error when the codebook design varies. Figure 13 demonstrates the high reconstruction quality on AFHQ. The latent EBM resembles the discriminator of BigGAN [3]. We use Adam for optimization where the learning rate is set at 0.001. We run the latent transport for 40 steps with a step size 1.0. We pretrain the VQ-VAE-2 on the whole AFHQ dataset including all the three domains cat, dog and wildlife. Therefore, if we want to obtain a model translating any two domains, the overall efficiency will be even higher than CUT as seen in Table 6.

More comparisons with CF-EBM: We present more results in Table 7.

Datasets	*cat* → *dog*	*dog* → *cat*	*vangogh* → *photo*
CF-EBM	6.20	9.21	**4.49**
Ours	**6.01**	**7.45**	4.61

Table 7. More KID comparisons with CF-EBM.

More results In Figure 14, we compare the translation results under various pretraining settings and a baseline model CUT [31]. We observe although the autoencoder is pretrained with a totally irrelevant dataset, we still can generate reasonable translations. Compared with our standard setting (a) and the baseline CUT, our model demonstrates better style controllability and content preservation ability. Figure 15 gives extended comparisons on AFHQ cat → dog. Figure 16 provides additional translation results on AFHQ dog → cat, cat → wild, dog → wild and wild → cat.

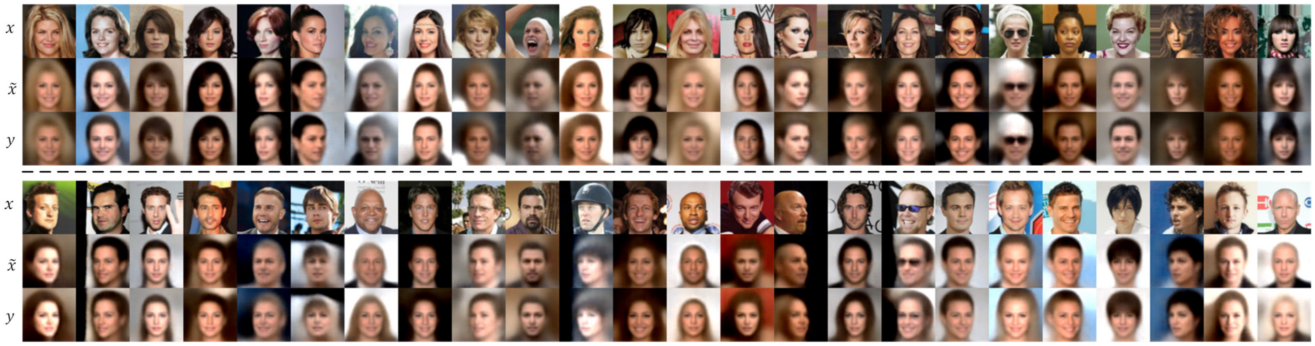

Figure 10. More uncurated results based on β-VAE. (*Top*) Male to Female; (*Bottom*) Female to Male. x: the input, $\tilde{x}$: the reconstruction, y: the translated output.

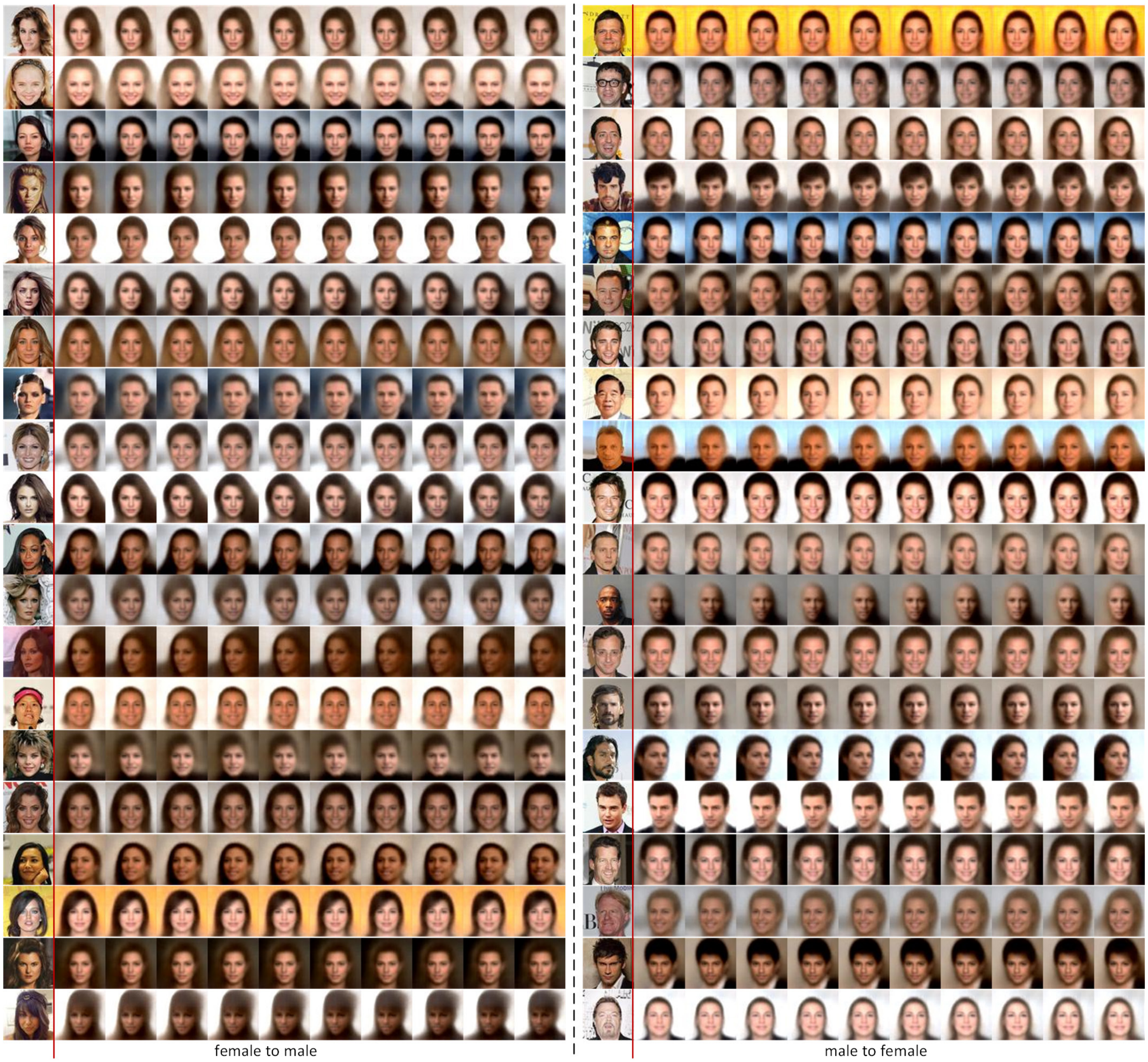

Figure 11. Smooth unpaired image-to-image translation dynamics via MCMC. The leftmost column is the input.

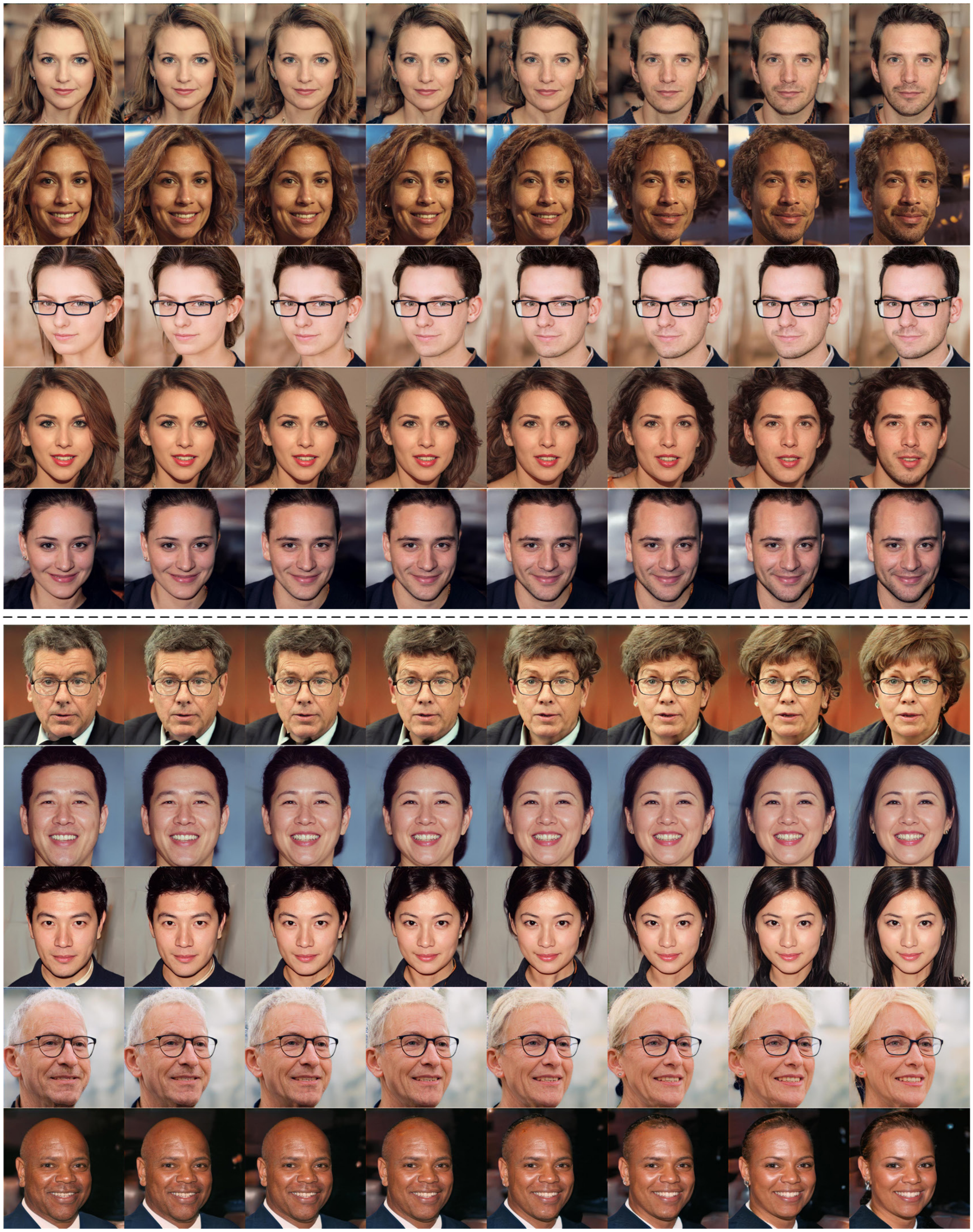

Figure 12. More 1024^2-pixel image translation dynamics based on ALAE. (*Top*) Female to Male, (*Bottom*) Male to Female.

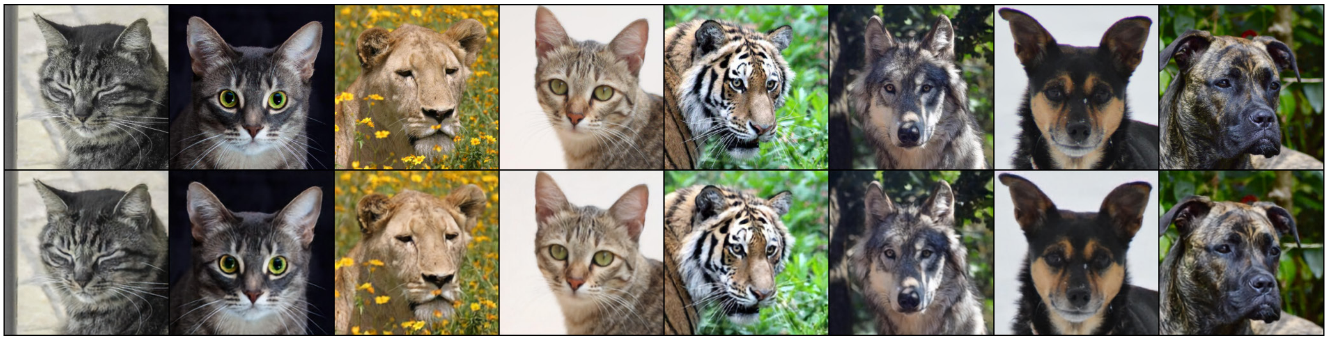

Figure 13. VQ-VAE-2 reconstructions on AFHQ. (*Top*) Inputs, (*Bottom*) Reconstructions.

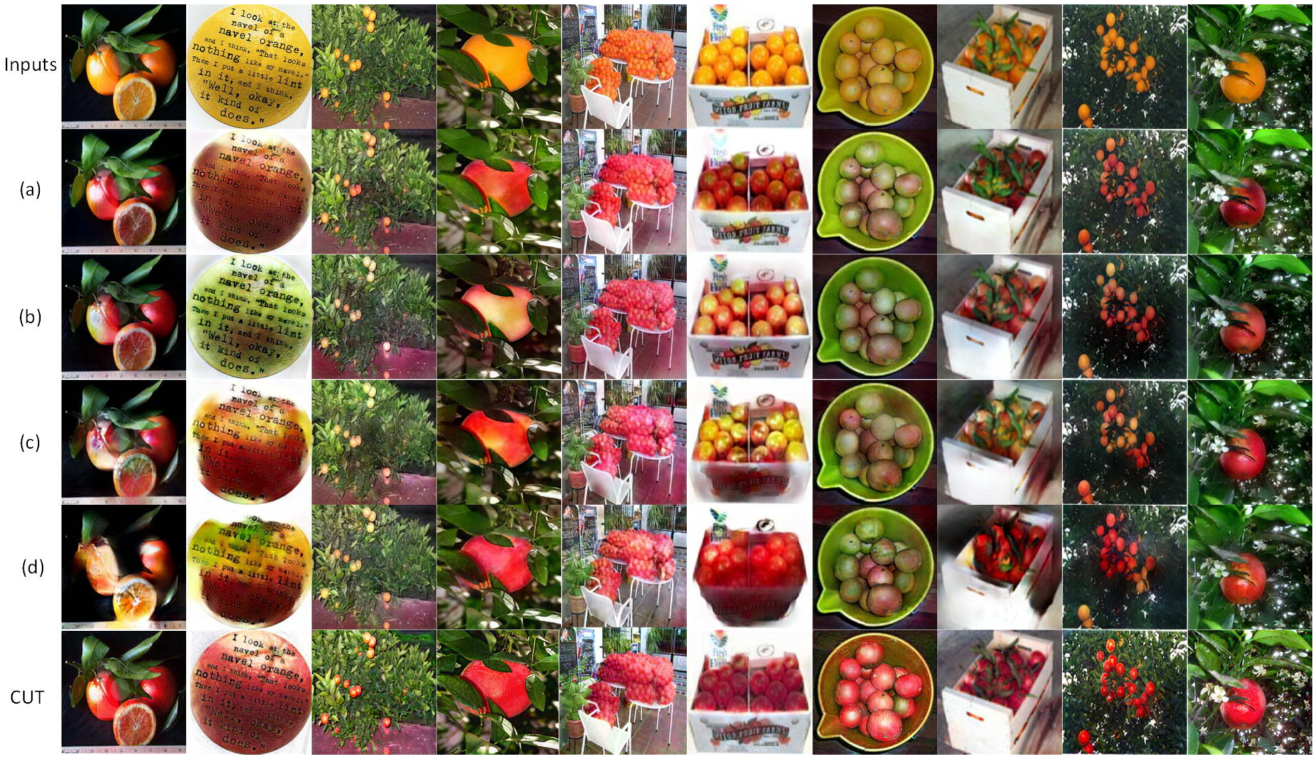

Figure 14. Uncurated translation results on orange → apple. (a)-(d) denote different pretraining settings. (a) Pretrain on apple2orange. (b) Pretrain on ImageNet. (c) Pretrain on CelebA-HQ. (d) Pretrain on AFHQ. The last row shows the results from CUT [31].

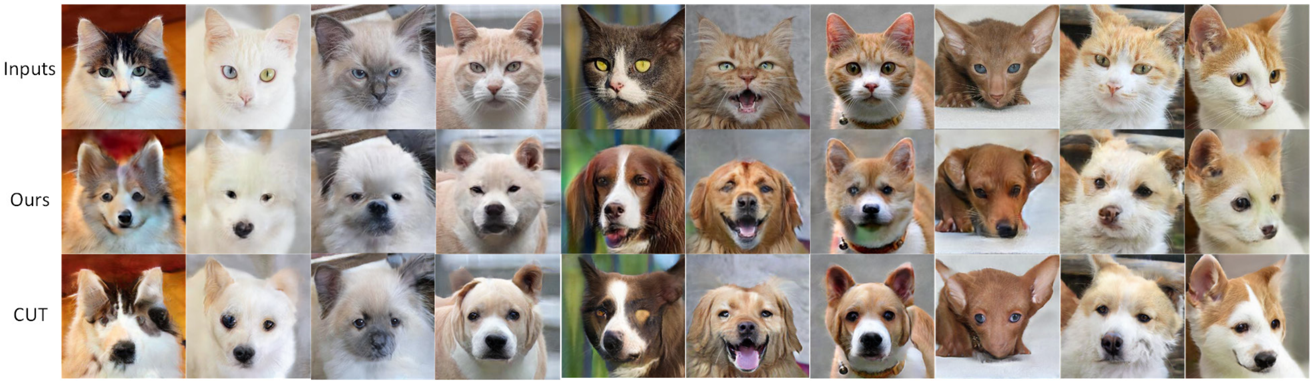

Figure 15. Extended translation results on AFHQ cat → dog based on VQ-VAE-2.

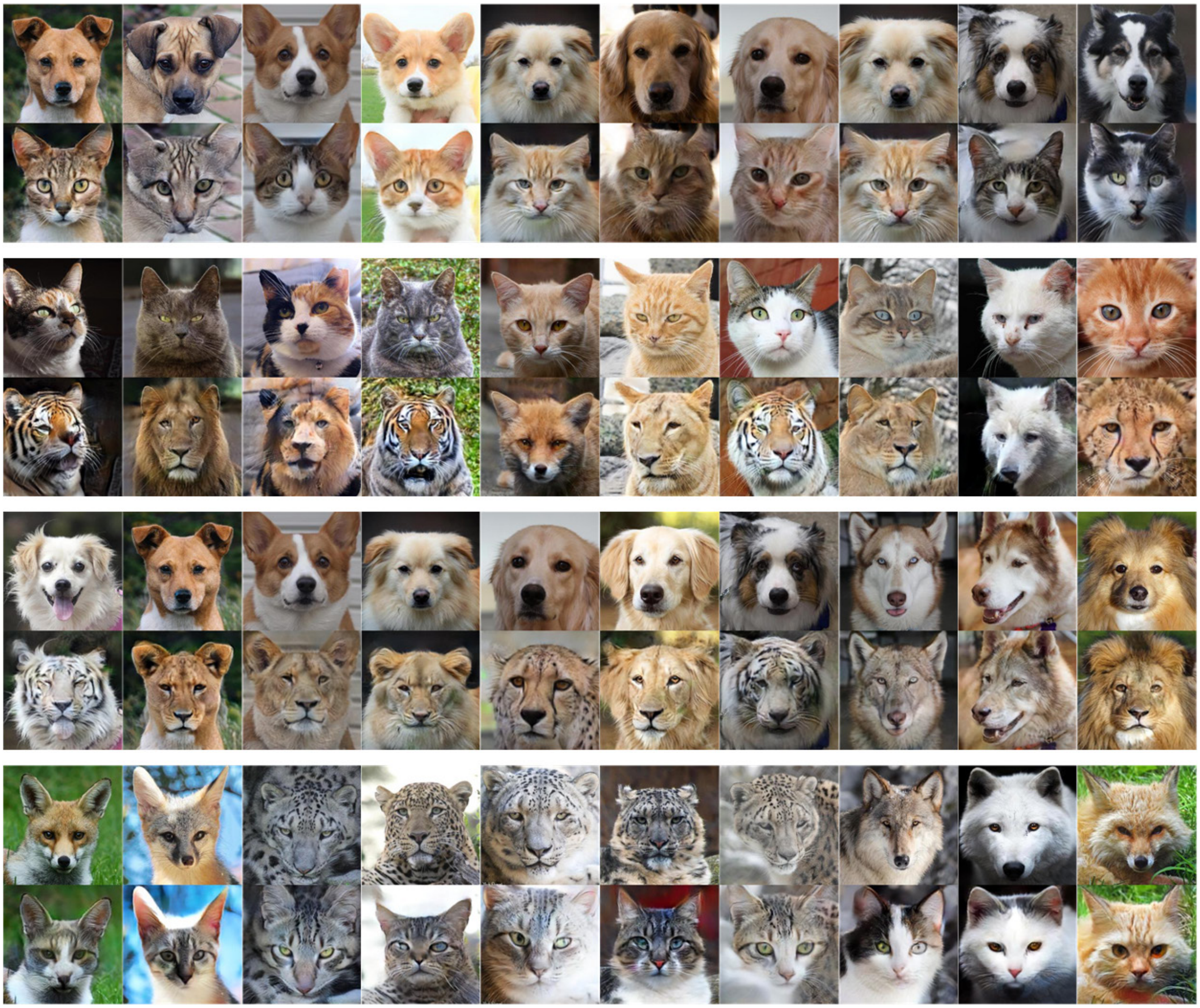

Figure 16. Additional translation results on AFHQ based on VQ-VAE-2. From Top to Bottom: dog → cat, cat → wild, dog → wild, wild → cat.

S D	128	256	512
1	-	4.73	4.59
2	-	2.84	2.62
4	-	2.67	2.53
8	-	2.40	2.38
32	2.62	2.38	2.29
64	-	2.31	2.07

Table 8. VQ-VAE-2 reconstruction quality (MSE:10^{-3}) under various codebook configurations in AFHQ. Each column varies the codebook dimension (D) and each row varies the codebook size (S).