A. Appendix

A.1. Implementation details

Architectures We borrow the generator and discriminator architectures from StyleGAN2 with two modifications; (i) The latent code is adapted to our formulation and forms a concatenation of three separate latent codes: y, u^{corr}, u^{uncorr}. (ii) We do not apply any noise injection during training. A brief summary of the architectures is presented in Tab. 4 and 5 for completeness. The architecture of the feed-forward encoders trained in the synthesis stage is influenced by StarGAN-v2 [5] and presented in Tab. 6. Note that we do not use any domain-specific layers and our model is able to generalize to unseen domains (e.g. labeled attribute) at inference time (e.g. new face identities in CelebA).

Optimization In the disentanglement stage, we optimize over a single u^{uncorr} embedding per image (dim = 64), a single y embedding per known attribute (dim = 512) and the parameters of E_c (dim = 64) and G. We set the learning rate of the latent codes to 0.01, the learning rate of the generator to 0.001 and the learning rate of the encoder to 0.0001. The penalty of the uncorrelated bottleneck is set to $\lambda_b = 0.001$. We train the disentanglement stage for 200 epochs. For each mini-batch, we update the parameters of the models and the latent codes with a single gradient step each. In the synthesis stage, we add two encoders to infer the latent codes learned in the disentanglement stage directly from an image. We also optimize a discriminator in an end-to-end manner to increase the perceptual fidelity of the images. This stage is trained for 100 epochs and the learning rate for all the parameters is set to 0.0001.

Baseline models For the evaluation of competing methods, we use the following official publicly available pretrained models: Lifespan [28] and SAM [1] (for age editing on FFHQ), StarGAN-v2 (for AFHQ and CelebA-HQ) and StyleGAN (for mGANprior on CelebA-HQ). We train the rest of the baselines using the official repositories of their authors and make an effort to select the best configurations available for the target resolution (for example, FUNIT trained by us for AFHQ achieves similar results to the public StarGAN-v2 which was known as the SOTA on this benchmark).

Evaluation Protocol We assess the disentanglement at two levels: the learned representations and the generated images. At the representation level, we follow the protocol in LORD [10] and train a two-layer multi-layer perceptron to classify the labeled attributes from the learned uncorrelated codes (lower accuracy indicates better disentanglement). In CelebA, where annotations of part of the uncorrelated attributes are available (for evaluation only) such as 68-facial landmark locations, we train a linear regression model to locate the landmarks given the learned identity codes (higher error indicates better disentanglement). At the image level, we follow StarGAN-v2 and translate all images in the test set to each of the other domains multiple times, borrowing correlated attribute codes from random reference images in the target domain. We then train a classifier to classify the domain of the source image from the translated image. A lower accuracy indicates better disentanglement as the source domain does not leak into the translated image. We also compute FID [13] in a conditional manner to measure the discrepancy between the distribution of images in each target domain and the corresponding translations generated by the models. A lower FID score indicates that the translations are more reliable and better fit to the target domain. FID between real train and test images of the same domain forms the optimal score for this metric. In order to assess the diversity of translation results, we measure the perceptual pairwise distances using LPIPS [37] between all translations of the same input image. Higher average distances indicate greater diversity in image translation. In cases where external annotation methods are available (for evaluation only), such as face recognition [3] and head pose [30] and landmark detection for CelebA, we further measure the similarity of the identity of the generated face and the reference, as well as expression (represented by landmarks) and head pose errors. To validate that the head pose and expression are distributed evenly across identities, we use landmark annotations together with pose-related attributes from CelebA (Open Mouth and Smiling) and train a classifier to infer the identity. The accuracy of this classifier (0.001) forms the optimal result for the representation metric. For translating males to females on CelebA-HQ, we measure the accuracy of fooling a target classifier trained on real images, as well as FID to evaluate how the images fit the target domain.

A.2. Datasets

FFHQ [19] $70,000$ high-quality images containing considerable variation in terms of age, ethnicity and image background. We use the images at 256×256 resolution. FFHQ-Aging [28] provides age labels for these images.

AFHQ [5] $15,000$ high quality images categorized into three domains: cat, dog and wildlife. We follow the protocol used in StarGAN-v2 and use the images at 256×256 resolution, holding out 500 images from each domain for testing.

CelebA [24] 202,599 images of 10,177 celebrities. We designate the person identity as class. We crop the images to 128×128 and use 9,177 classes for training and 1,000 for testing.

CelebA-HQ [18] 30,000 high quality images from CelebA. We set the gender as class. We resize the images

to 256×256 and leave 1,000 images from each class for testing. The masks provided in CelebAMask-HQ [21] are used to disentangle the correlated hairstyle.

Edges2Shoes [35] A collection of 50,000 shoe images and their edge maps.

Training resources Training each of the models presented in this paper takes approximately 3 days for 256x256 resolution on a single NVIDIA RTX-2080 TI.

A.3. Additional results

We provide additional qualitative results on facial age editing (Fig. 8, 9, 10), identity transfer (Fig. 15), pose-appearance translation (Fig. 12, 13), Male-to-Female translation (Fig. 14) and Shape-Texture transfer (Fig. 18).

A.4. Latent optimization

In this work, we opt for learning the representation of the unlabeled uncorrelated attributes (u^{uncorr}) using latent optimization, similarly to LORD [10]. Autoencoders assume a parametric model, usually referred to as the encoder, to compute a latent code from an image. On the other hand, we jointly optimize the latent codes and the generator (decoder) parameters. Since the latent codes are learned directly and are unconstrained by a parametric encoder function, our model can recover all the solutions that could be found by an autoencoder, and reach some others. In order to justify this design choice, we validate the observation presented in [10] stating that latent optimization improves disentanglement and train our disentanglement stage in an amortized fashion using E_u. As can be seen in Fig. 19, amortized training fails to reduce the correlation between the labeled and unlabeled representations. We have experimented with several decay factors (λ_b in Eq. 6). Although the disentanglement improves as λ_b increases, the reconstruction gets worse and the model fails to converge with $\lambda_b > 0.1$.

A.5. Visualization of ablation analysis

Examples from the ablation analysis are provided in Fig. 16. Visualization of the three sets of attributes modeled by our method is provided in Fig. 17.

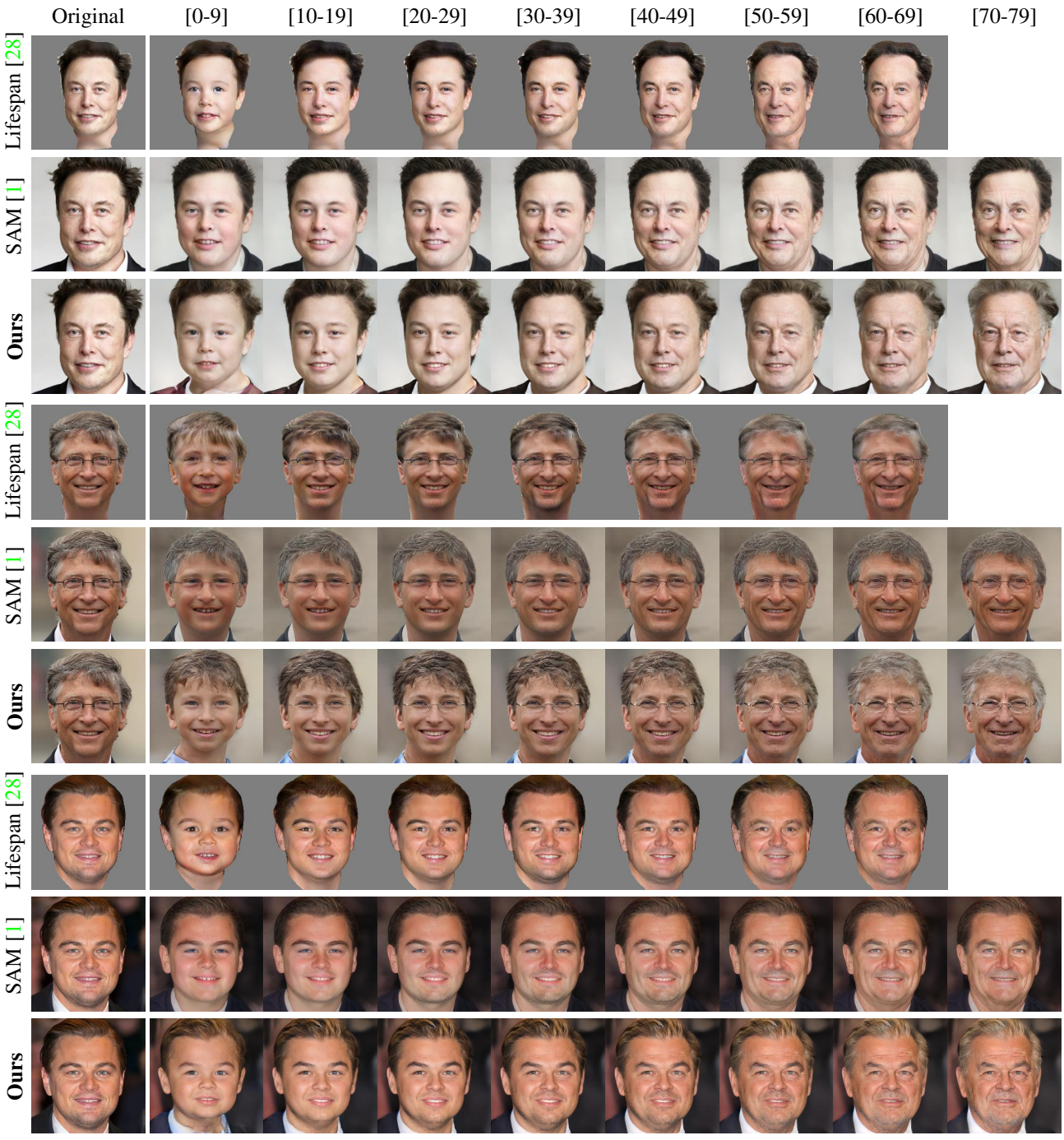


Figure 8: An extended comparison of disentangling age and unlabeled attributes (e.g. identity). Our general method performs better on age-editing than the two task-specific baselines which rely on a supervised identity loss.

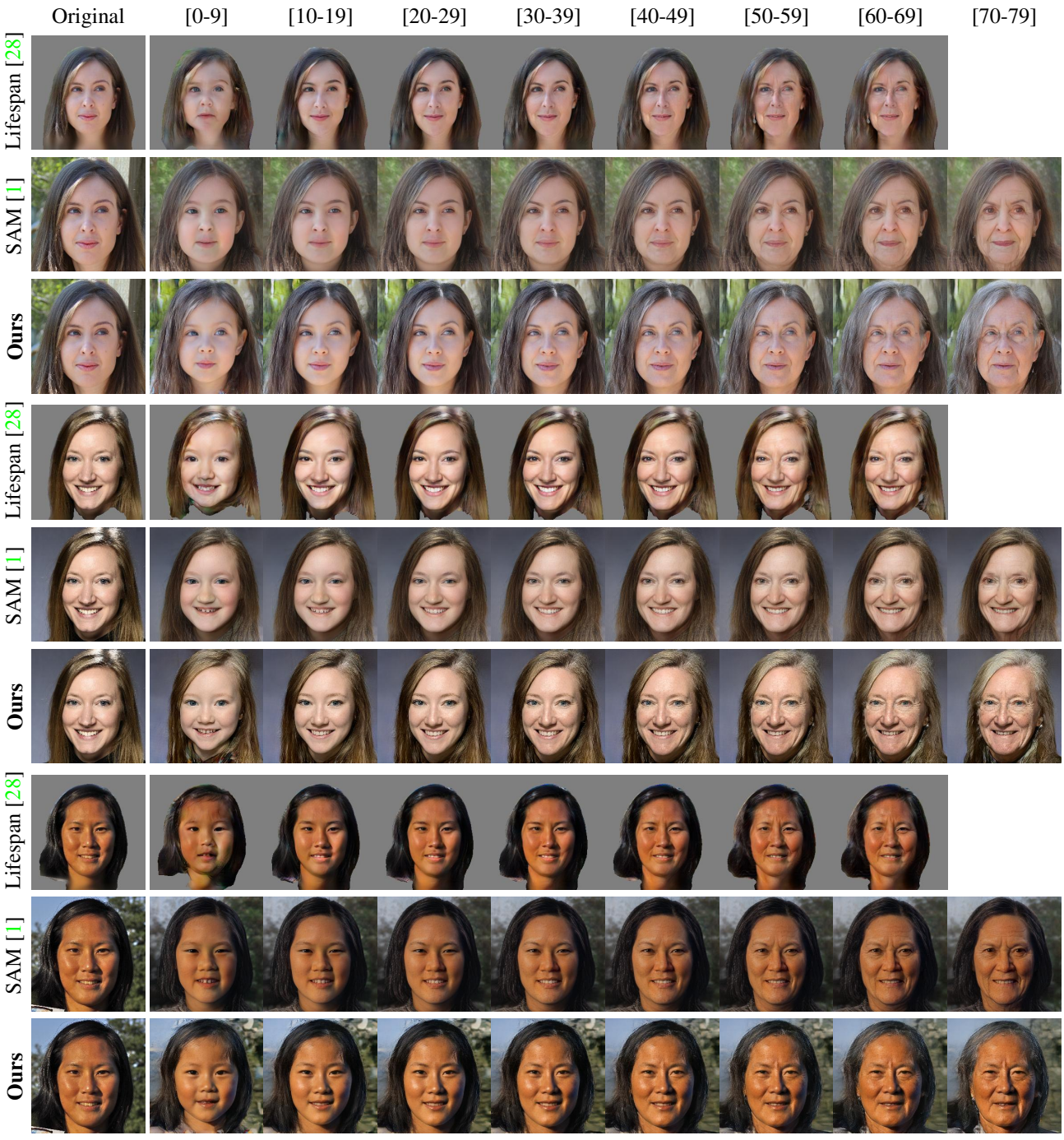


Figure 9: More qualitative results of facial age editing. Our model makes more significant changes (e.g. hair color) while preserving the identity better than the baselines.

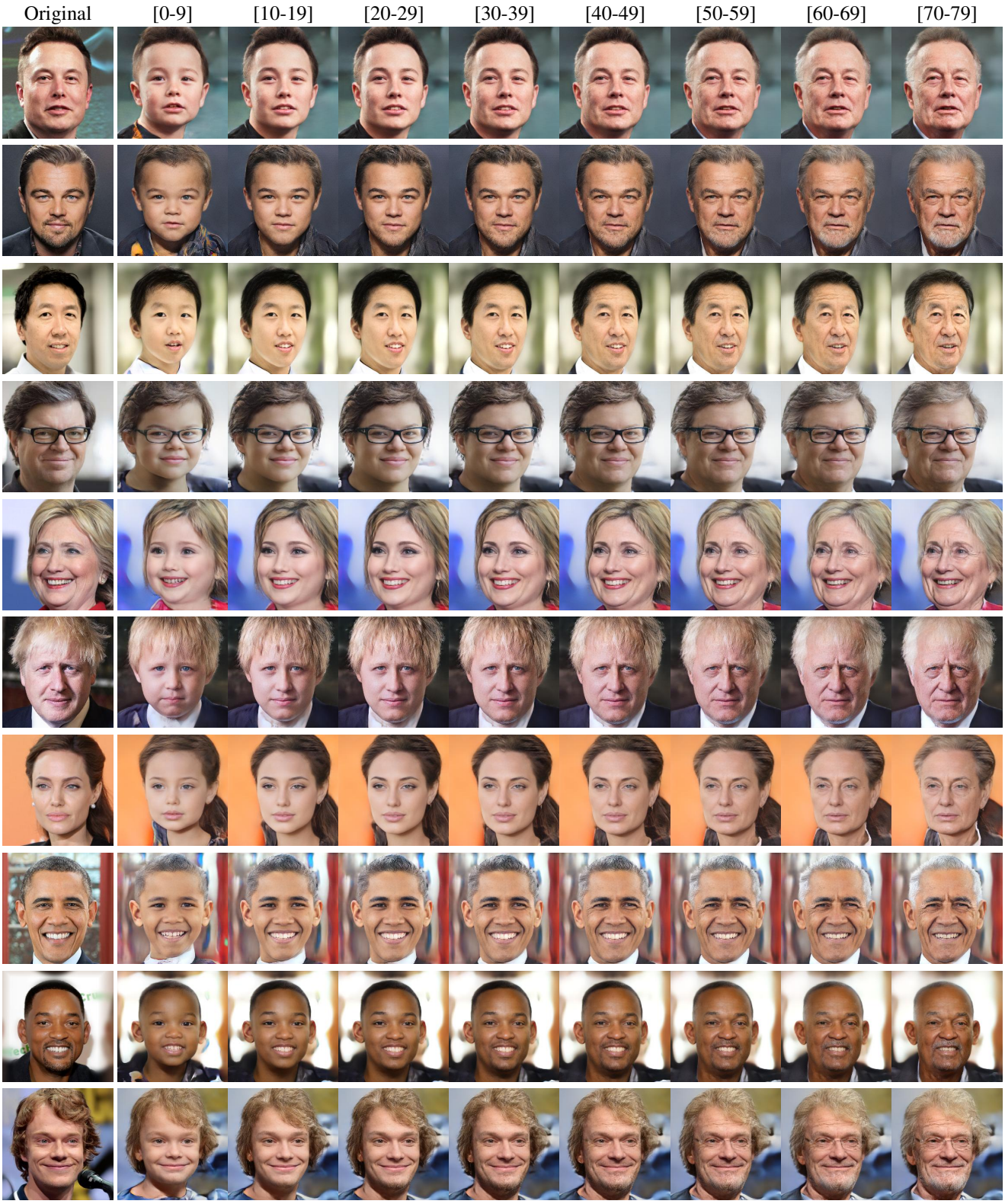


Figure 10: More qualitative results of facial age editing with our model.

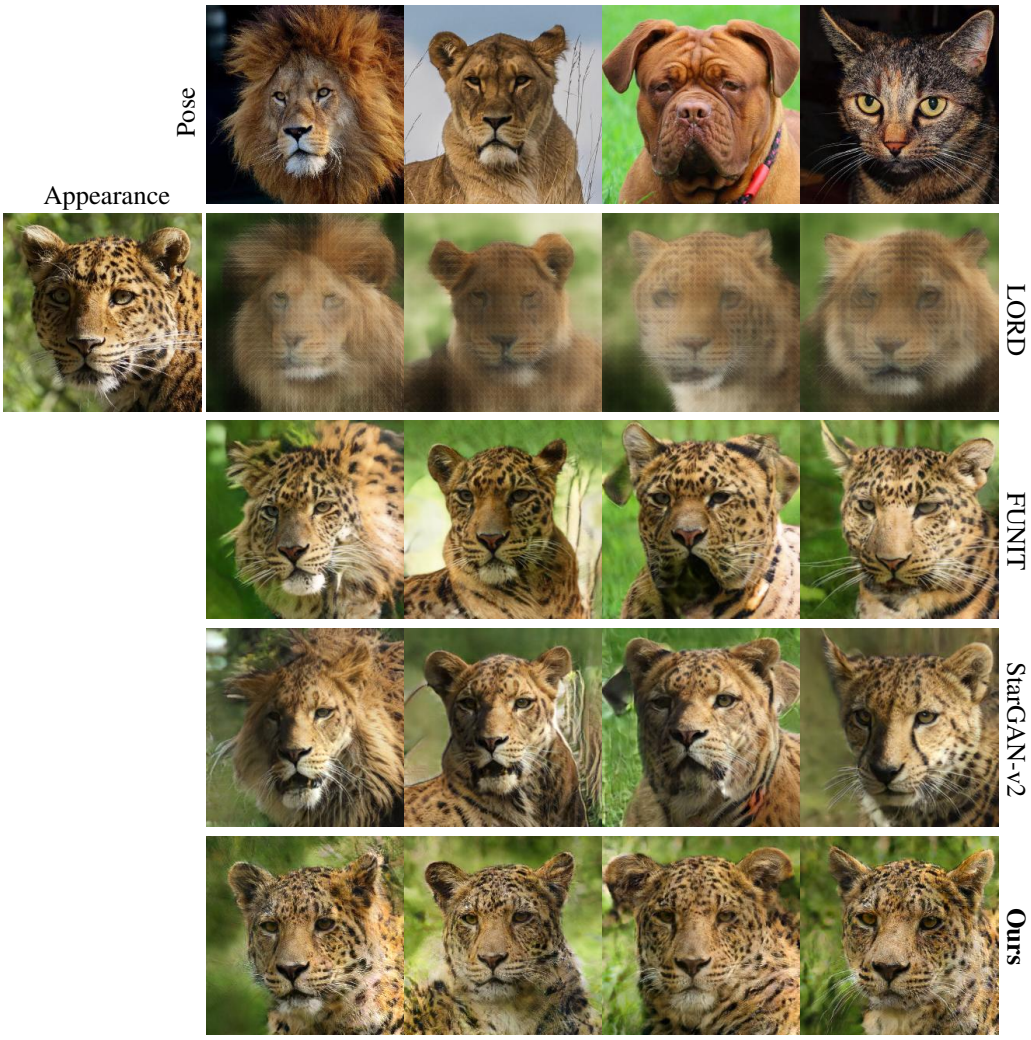


Figure 11: Comparison against LORD in the presence of correlated attributes (AFHQ). LORD does not distinct between correlated and uncorrelated attributes and can not utilize a reference image e.g. translating the cat into a wild animal is poorly specified and results in a tiger instead of a leopard. Moreover, the generated images exhibit low visual quality which is further improved by our method. FUNIT and StarGAN-v2 leak some of the correlated attributes such as the lion's mane and the dog's facial shape, leading to unreliable translation between species.

Figure 12: More qualitative results on AFHQ.

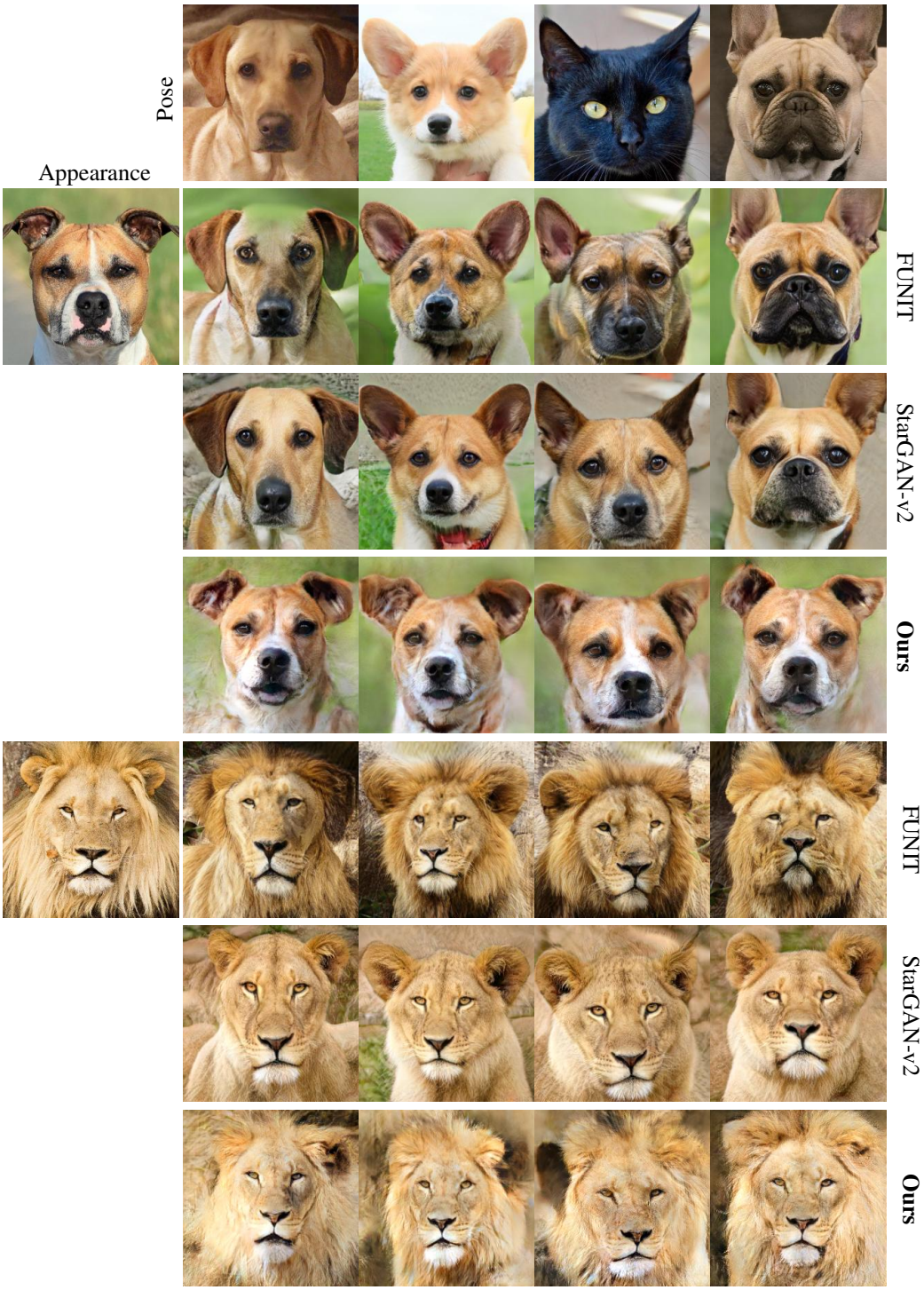


Figure 13: More qualitative results on AFHQ.

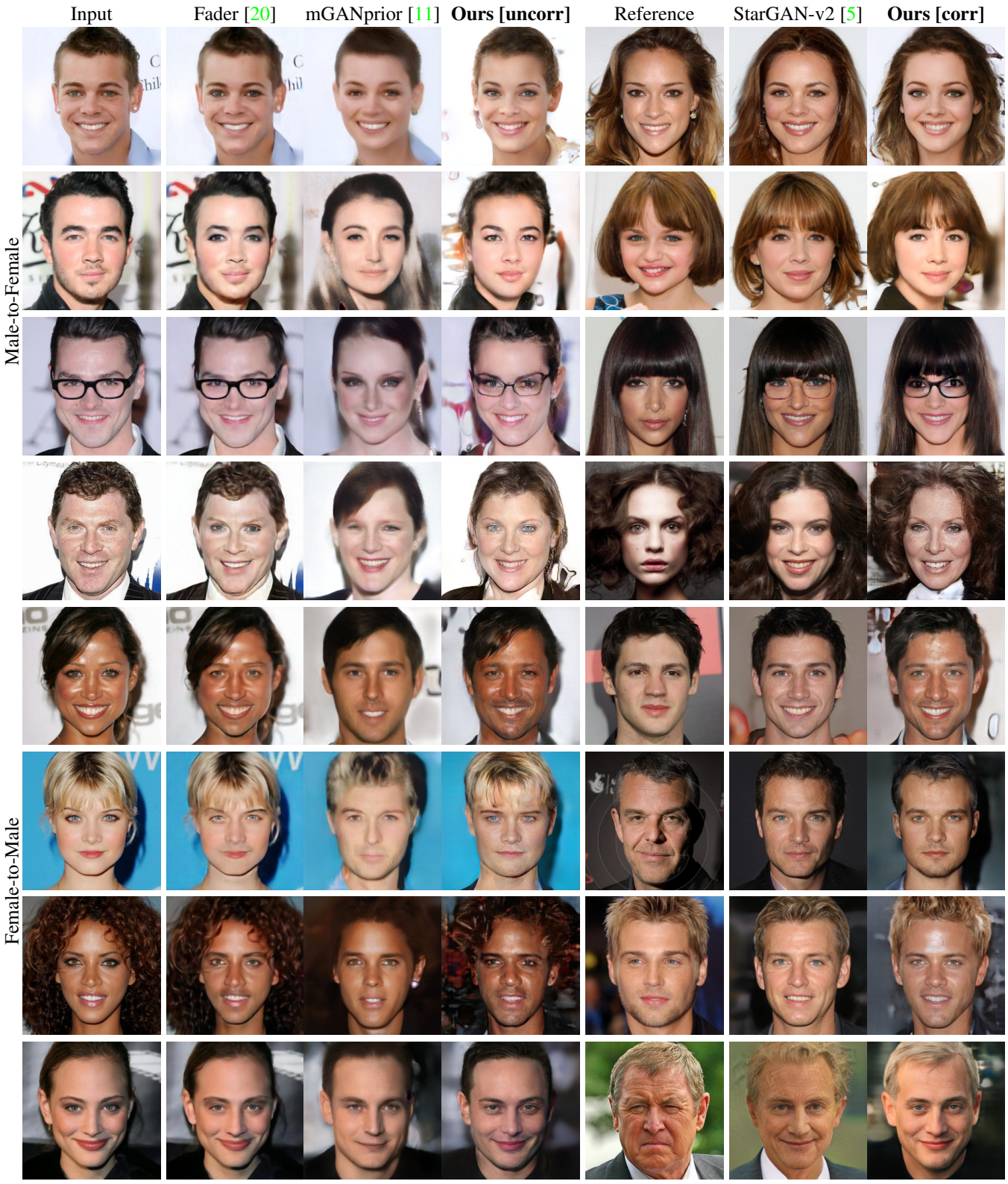
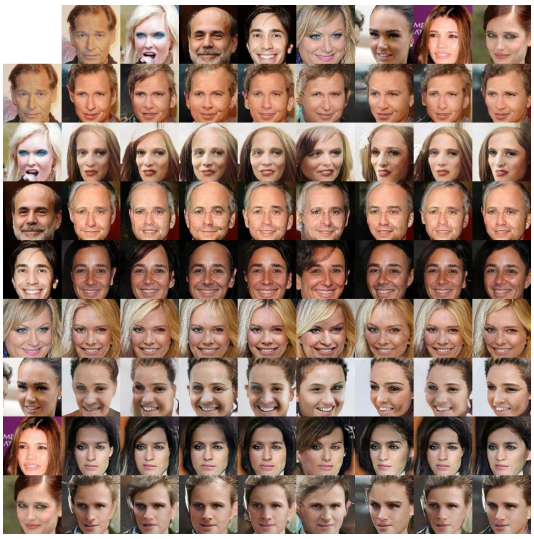


Figure 14: More qualitative results of Male-to-Female translation in two settings: (i) When the attributes are assumed to be *uncorrelated*. (ii) When we model the hair style as the *correlated* attribute and utilize a reference image specifying its target. Our method preserves the uncorrelated attributes including identity, age and illumination better than StarGAN-v2.



(a) FUNIT

(b) LORD

(c) Ours

Figure 15: More qualitative results on CelebA in the task of translating facial identities (left column) across different unlabeled head poses, expressions and illumination conditions (top row).

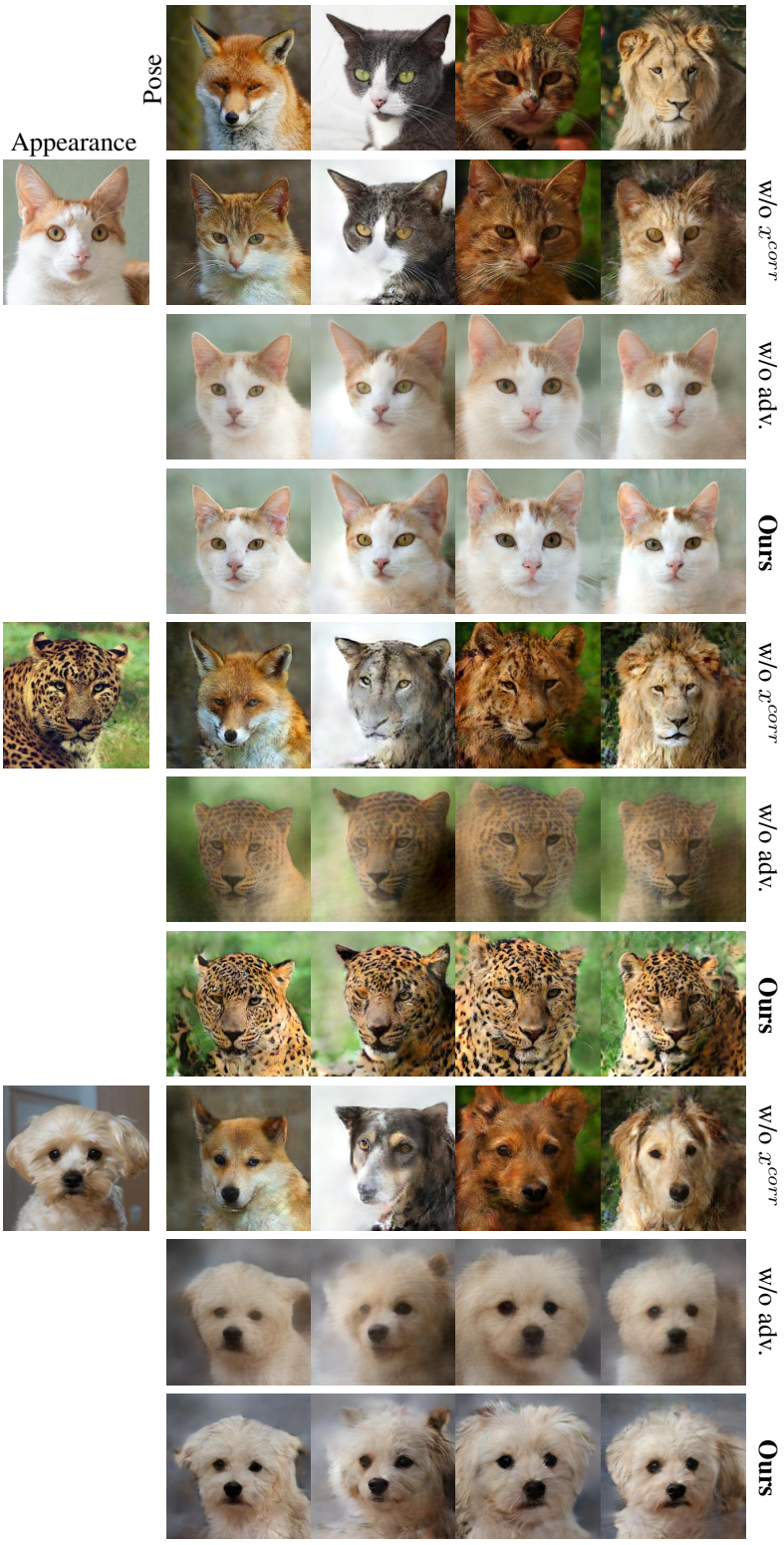


Figure 16: Qualitative examples from the ablation analysis; (i) w/o x^{corr}: Leaving the correlated attributes intact leads to unreliable and entangled translations. (ii) w/o adv.: Disentanglement is achieved without the adversarial loss, which mostly contributes to the visual fidelity.

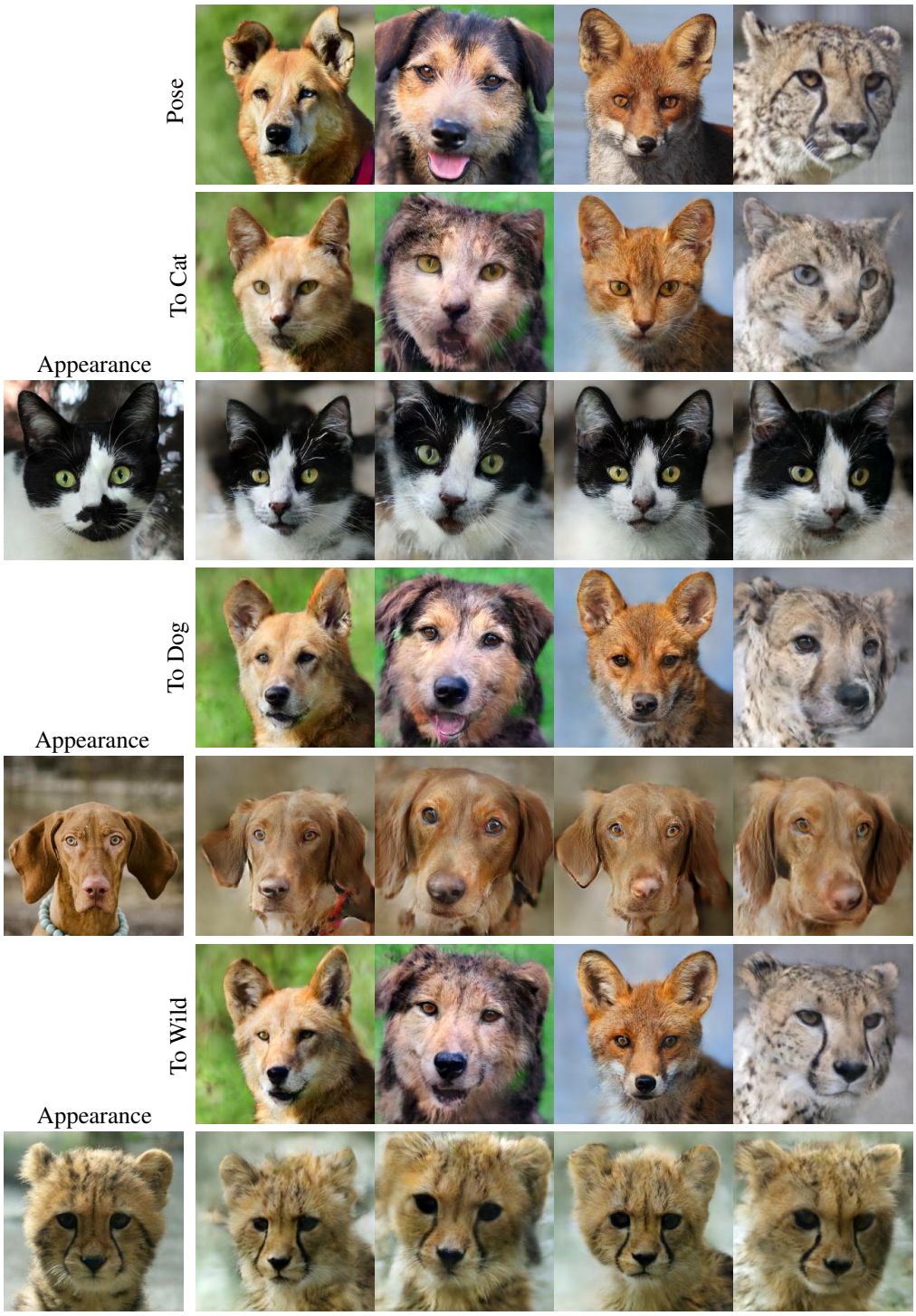


Figure 17: Visualization of the three sets of attributes modeled by our framework. Changing the labeled attribute (e.g. To cat, dog, wild) while leaving the unlabeled correlated attributes intact affects high level semantics of the presented animal, although can generate unreliable translations (e.g. rightmost image in "To Dog"). Guiding the correlated attributes by a reference image allows for specification of the exact target appearance (e.g. breed). The remaining unlabeled and uncorrelated attributes mainly encode the pose of the animal.

Figure 18: More qualitative results on Edges2Shoes.

Table 4: Generator architecture based on StyleGAN2. StyleConv and ModulatedConv use the injected latent code composed of y, u^{corr}, u^{uncorr}.

Layer	Kernel Size	Activation	Resample	Output Shape
Constant Input	-	-	-	$4 \times 4 \times 512$
StyledConv	3×3	FusedLeakyReLU	-	$4 \times 4 \times 512$
StyledConv	3×3	FusedLeakyReLU	UpFirDn2d	$8 \times 8 \times 512$
StyledConv	3×3	FusedLeakyReLU	-	$8 \times 8 \times 512$
StyledConv	3×3	FusedLeakyReLU	UpFirDn2d	$16 \times 16 \times 512$
StyledConv	3×3	FusedLeakyReLU	-	$16 \times 16 \times 512$
StyledConv	3×3	FusedLeakyReLU	UpFirDn2d	$32 \times 32 \times 512$
StyledConv	3×3	FusedLeakyReLU	-	$32 \times 32 \times 512$
StyledConv	3×3	FusedLeakyReLU	UpFirDn2d	$64 \times 64 \times 512$
StyledConv	3×3	FusedLeakyReLU	-	$64 \times 64 \times 512$
StyledConv	3×3	FusedLeakyReLU	UpFirDn2d	$128 \times 128 \times 256$
StyledConv	3×3	FusedLeakyReLU	-	$128 \times 128 \times 256$
StyledConv	3×3	FusedLeakyReLU	UpFirDn2d	$256 \times 256 \times 128$
StyledConv	3×3	FusedLeakyReLU	-	$256 \times 256 \times 128$
ModulatedConv	1×1	-	-	$256 \times 256 \times 3$

Table 5: Discriminator architecture based on StyleGAN2.

Layer	Kernel Size	Activation	Resample	Output Shape
Input	-	-	-	$256 \times 256 \times 3$
Conv	3×3	FusedLeakyReLU	-	$256 \times 256 \times 128$
ResBlock	3×3	FusedLeakyReLU	UpFirDn2d	$128 \times 128 \times 256$
ResBlock	3×3	FusedLeakyReLU	UpFirDn2d	$64 \times 64 \times 512$
ResBlock	3×3	FusedLeakyReLU	UpFirDn2d	$32 \times 32 \times 512$
ResBlock	3×3	FusedLeakyReLU	UpFirDn2d	$16 \times 16 \times 512$
ResBlock	3×3	FusedLeakyReLU	UpFirDn2d	$8 \times 8 \times 512$
ResBlock	3×3	FusedLeakyReLU	UpFirDn2d	$4 \times 4 \times 512$
Concat stddev	3×3	FusedLeakyReLU	UpFirDn2d	$4 \times 4 \times 513$
Conv	3×3	FusedLeakyReLU	-	$4 \times 4 \times 512$
Reshape	-	-	-	8192
FC	-	FusedLeakyReLU	-	512
FC	-	-	-	1

Table 6: Encoder architecture based on StarGAN-v2. Note that we do not use any domain-specific layers. D is the dimension of y, u^{corr}, u^{uncorr} respectively.

Layer	Kernel Size	Activation	Resample	Output Shape
Input	-	-	-	$256 \times 256 \times 3$
Conv	3×3	-	-	$256 \times 256 \times 64$
ResBlock	3×3	LeakyReLU ($\alpha = 0.2$)	Avg Pool	$128 \times 128 \times 128$
ResBlock	3×3	LeakyReLU ($\alpha = 0.2$)	Avg Pool	$64 \times 64 \times 256$
ResBlock	3×3	LeakyReLU ($\alpha = 0.2$)	Avg Pool	$32 \times 32 \times 256$
ResBlock	3×3	LeakyReLU ($\alpha = 0.2$)	Avg Pool	$16 \times 16 \times 256$
ResBlock	3×3	LeakyReLU ($\alpha = 0.2$)	Avg Pool	$8 \times 8 \times 256$
ResBlock	3×3	LeakyReLU ($\alpha = 0.2$)	Avg Pool	$4 \times 4 \times 256$
Conv	4×4	LeakyReLU ($\alpha = 0.2$)	-	$1 \times 1 \times 256$
Reshape	-	-	-	256
FC	-	-	-	D

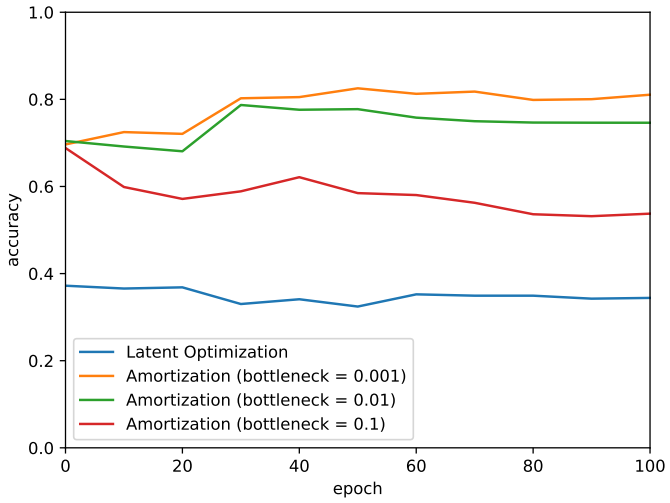


Figure 19: Evidence for the inductive bias conferred by latent optimization on AFHQ (a validation of the discovery presented in [10]). We plot the accuracy of an auxiliary classifier predicting the labeled attributes from the learned representations of the unlabeled attributes. Latent optimization starts with randomly initialized latent codes and preserves the disentanglement of the labeled and unlabeled representations along the entire training (the accuracy matches a random guess). In contrast, a randomly initialized encoder (amortization) outputs entangled codes. In order to reach disentanglement, the encoder should distillate the information of the labeled attributes during the optimization, which is shown to be unsuccessful in practice.