Mod	Acc	r_{attr}	r_{rest}	V_{rest}
original	0.94	0.4	0.11	0.01
two enc	0.15	0.32	0.24	0.07
one dec	0.49	0.71	0.04	0.02
k=16	0.14	0.8	0.5	0.25
d=64	0.75	0.61	0.42	0.01

Table 3: Ablation study for digit rotation: two encoders instead of single shared encoder, non-shared decoder, smaller rotation embedding (same overall embedding size) and two times reduced dimensionality of both embeddings.

Model	Attribute	$J^{\text{syn}}_{\text{attr}}$	$J^{\text{syn}}_{\text{rest}}$	$J^{\text{gen}}_{\text{attr}}$	$J^{\text{gen}}_{\text{rest}}$
PuppetGAN	Rot	0.05	2.2	0.03	0.32
CycleGAN				0.05	0.28
PuppetGAN†		0.06	108	$+\infty$	$+\infty$
PuppetGAN	Size	0.27	0.78	0.24	0.04
CycleGAN				0.20	0.07
PuppetGAN†		0.90	0.92	0.27	0.05

Table 4: For moderate discrepancies in attribute distributions, AoI in generated images followed the distribution of AoI in the real domain.

6. Supplementary

Please take a look at our video demonstration available at http://bit.ly/iccv19_pupgan and in the attached MP4 file (x264 codec). More images with more detailed experiments on all datasets are given below.

6.1. Additional Motivating Examples

A single "emerging" attribute of a complex simulation often can be randomly perturbed (or fixed) without explicitly measuring it. As an example, in an algorithm for synthesizing vascular tree-like structures proposed by Hamarneh and Jassi [2010], global parameters are specified ahead of time, but local topology is determined pseudo-randomly using a user-provided seed. A label-based method would require an explicit numeric representation of the tree topology, whereas our approach would only need to know which samples used the same random seeds. In general, our approach seems to better fit procedural pseudo-random data models, often used to model natural phenomena. Or if the scene contains multiple objects, representing their attributes (e.g. positions and colors of a variable number of circles) in a way that is invariant to their permutations is still an open problem [Zaheer et al 2017, Wagstaff et al 2019] - this issue does not arise in a triplet-based approach.

6.2. Error Case Analysis

Two major sources of errors in attribute manipulation are naturally coupled attributes in real data and mode collapse in attributes with multimodal distributions.

1. Since MNIST digit classes have different distributions of inclination, keeping digit's class label constant while rotating it is difficult (8s rotated too counter-clockwise sometimes turn into 3s); the same reasoning applies to keeping subject's mouth open while it moves away from the microphone.

2. If an attribute's distribution is multimodal with one mode dominating the other, GAN losses induce mode collapse in generators, i.e. make them handle more frequent attribute combinations at the cost of rarer ones.

3. Our experiments suggest that the proposed regularizers help considerably in combating both issues, but over-regularization leads to excessive "unification" of model outputs, e.g. more subtle attributes of digits (stroke, handwriting style) are less preserved during manipulation than more vivid attributes (size, rotation, position). For example, removal of the cycle attribute losses led to more distinct outputs for real inputs with different styles, but hurt robustness to degenerate solutions. Practitioners are encouraged to trade-off output diversity and manipulation precision on case-by-case basis by varying regularization parameters.

6.3. Implementation details.

Architecture. We used the "CycleGAN resnet" encoder (padded 7x7 conv followed by two 3x3 conv with stride 2 all with relus), followed by six residual conv blocks (two 3x3 convs with relus) a fully-connected bottleneck of size 128 and a pix2pix decoder (two bi-linear up-sampling followed by a convolution). We used LS-GAN objective in all GAN losses. It generally follows the architecture of CycleGAN implementation provided in the tfgan package[1].

Training. We optimized the entire loss jointly with respect to all encoder-decoder weights and then all discriminator losses in two consecutive iterations of the Adam optimizer with $\alpha =$ (2e-4, 5e-5) learning rates with polynomial decay and $\beta = 0.5$. A model trained by updating different losses wrt different weights independently in alternating fashion did not converge, so all generator and discriminator losses must be updated together in two large steps. We also added Gaussian instance noise to each image used in disentanglement and attribute cycle losses to improve stability during training. We added stop gradient op after the application of C_B in the second attribute cycle loss and instance

[1] https://www.tensorflow.org/api_docs/python/tf/contrib/gan/CycleGANModel

noise to all intermediate images to avoid the "embedding" behaviour.

We purposefully avoid constraining embeddings themselves, *e.g*. penalizing Euclidean distances between embedding components of images that are known to share a particular attribute, as such penalties often cause embedding magnitudes to vanishing.

Hyperparameters. The reasonable choice of loss weights we used is given below. We did not perform any large-scale hyperparameter optimization, just tried a couple of combinations, the $\mathcal{L}_{rest}$ weight required few (2-3) manual tuning attempts to balance $\mathcal{L}_{attr}$, (*i.e*. tried 1 then 5 then 3).

$$\mathcal{L}_{total} = 10 \cdot \mathcal{L}_{rec} + 10 \cdot \mathcal{L}_{cyc} + 5 \cdot \mathcal{L}_{attr} + 3 \cdot \mathcal{L}_{rest} + \\ + \sum_{K_1, K_2, K_3} \sum_{x \in K_1, y \in K_2} \mathcal{L}_{GAN}^{(K_3)}(C_{K_3}(x, y))$$

Metrics. The quality of attribute isolation can also be evaluated by estimating mutual information between attribute values and parts of embeddings that should or should not encode it [8]; we do not explicitly penalize our model for embedding extra information as long as decoder learns to ignore it, so this metric was not useful in our case.

Related methods. In related experiments we used a modified DiDA implementation from https://github.com/yangyanli/DiDA/ in both last and best training modes, MUNIT from https://github.com/NVlabs/MUNIT and cycle-consistent VAE https://github.com/ananyahjha93/cycle-consistent-vae.

6.4. Extended Results

"Saturated" inputs. To clarify, by "model saturates" we mean that if we pass synthetic inputs with the AoI value beyond what we used during training, model outputs reasonable "highest" or "lowest" output for respective domains instead of breaking (it could, since inputs are not typical).

Digits. You can find results for USPS in Figures 7 and 8, model did not manage to disentangle rotation in USPS probably due to the lack of thereof.

300-VW. In addition to the attached video, static examples of manipulated faces can be found in Figures 9, 10, 11 and 12. As pointed in the main paper, model properly preserves orientation and expression of the real input, and mouth expression of the synthetic input, and completely discards everything else.

YaleB. One can find more examples in Figures 13, 14, 15, 16 and 17. Identities of real inputs are preserved most of the time (as a reminder, all YaleB images were combined into a large single domain with no identity labels, so the model had to learn to disentangle and preserve identity). In cases when too little light is available in the real scene, the model "hallusinates" an "average" identity details. When model is asked to "imagine" lighting conditions that were not present in YaleB, but present in the synthetic dataset, some identity details are corrupted.

Synthetic faces. In Figure 18 we present more examples of outputs of a model trained to disentangle lighting across synthetic identities. "Dot artifacts" disappear if model is trained long enough.

Color blind and print friendly. An alternative version of Figure 4 is given in Figure 19.

Figure 7: Rotation was not disentangled, probably due to the lack thereof in USPS naturally.

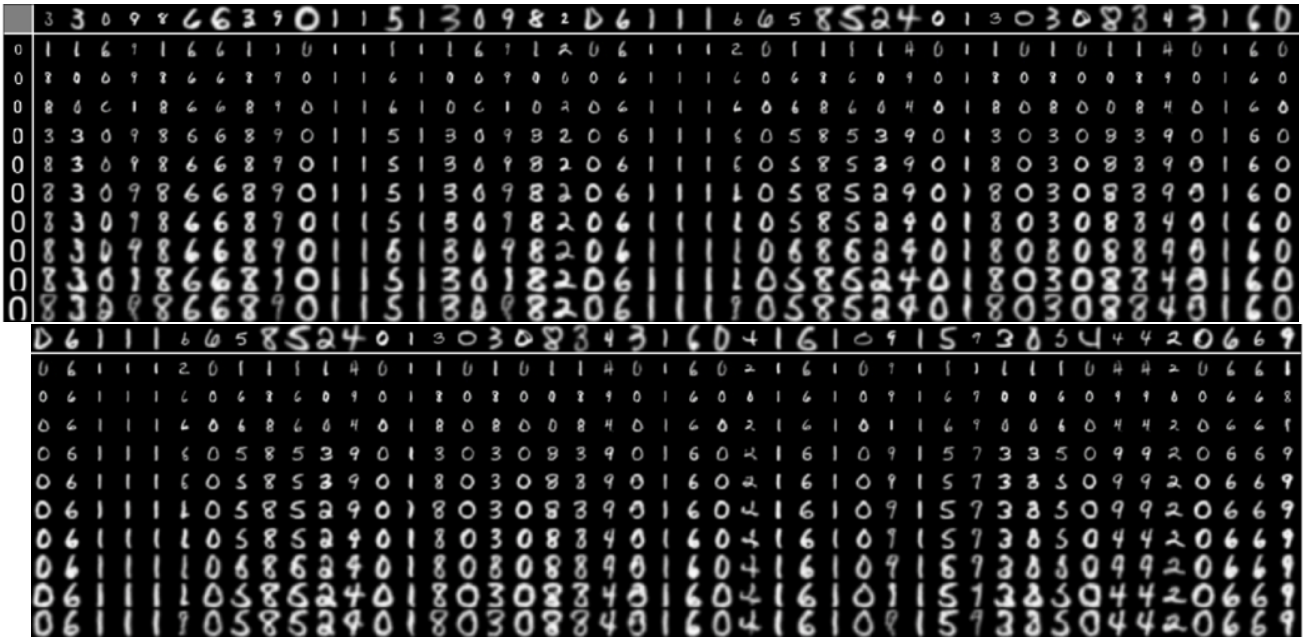


Figure 8: Size disentangled in USPS digit using synthetic renders.

Figure 9: More random examples for an identity from 300-VW dataset with mouth expression manipulated using our model. Two first and two last rows are “saturated” examples.

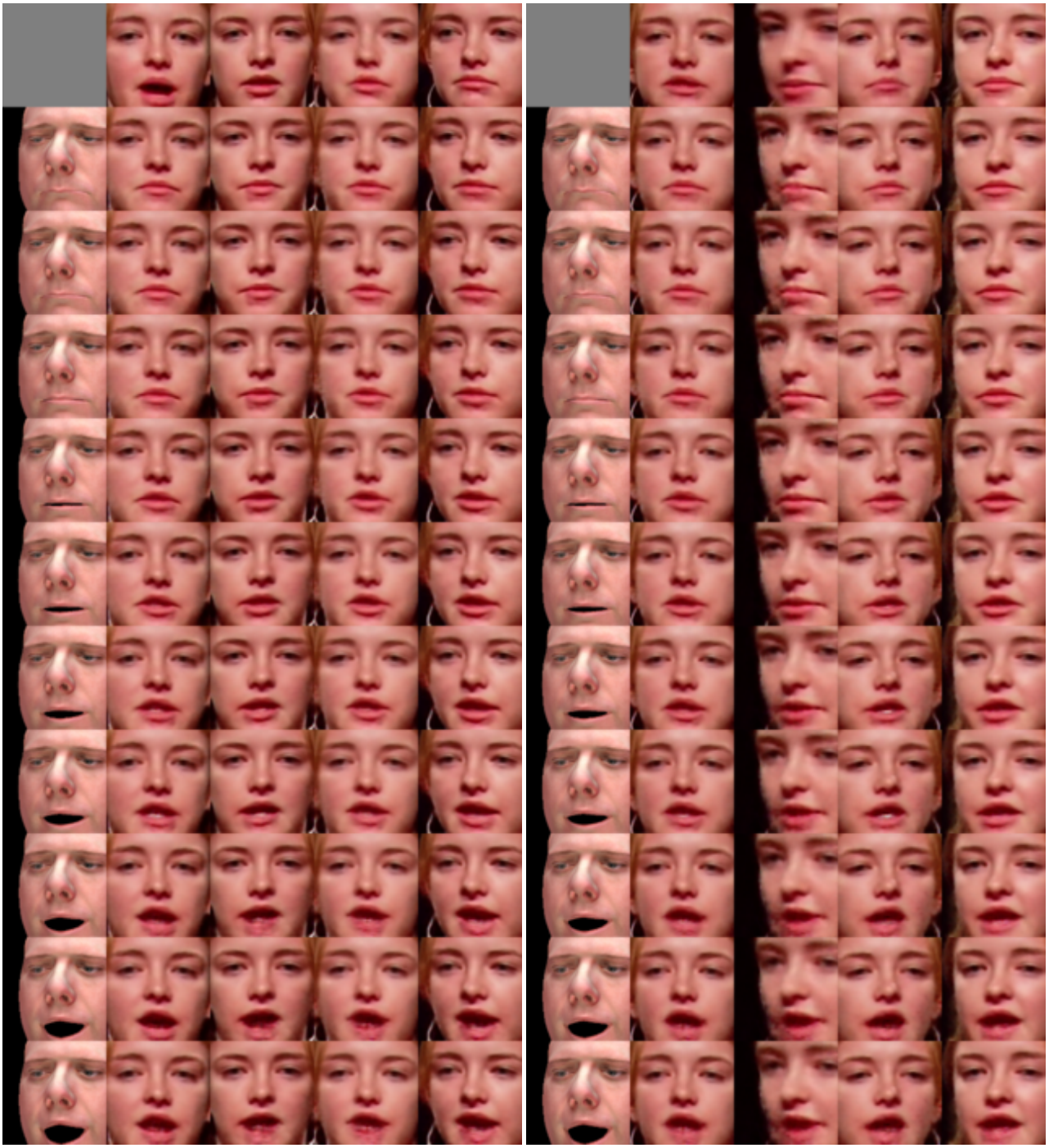


Figure 10: More random examples for an identity from the 300-VW dataset with mouth expression manipulated using our model. Two first and two last rows are “saturated” examples.

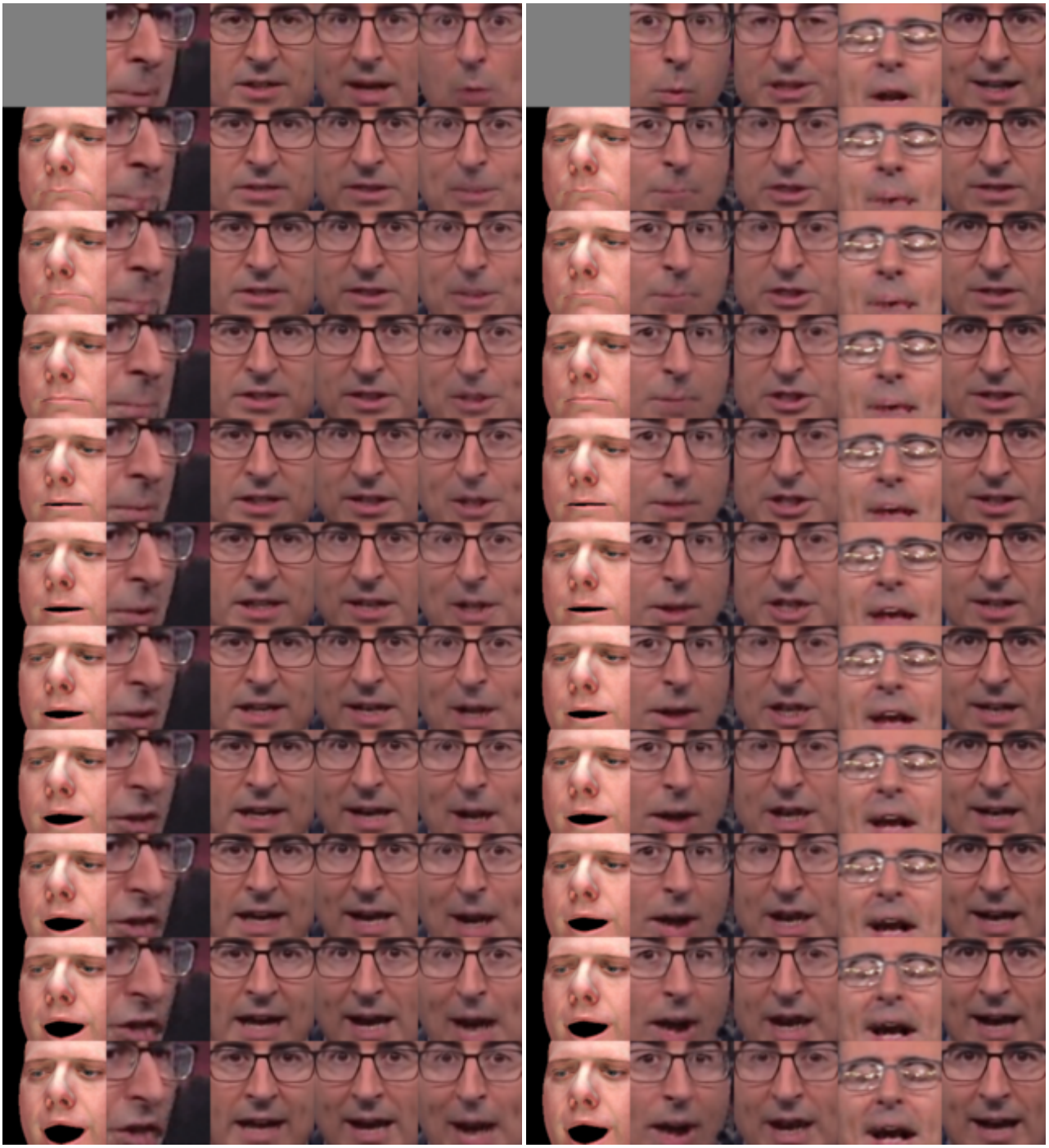


Figure 11: More random examples for an identity from the 300-VW dataset with mouth expression manipulated using our model. Two first and two last rows are “saturated” examples.

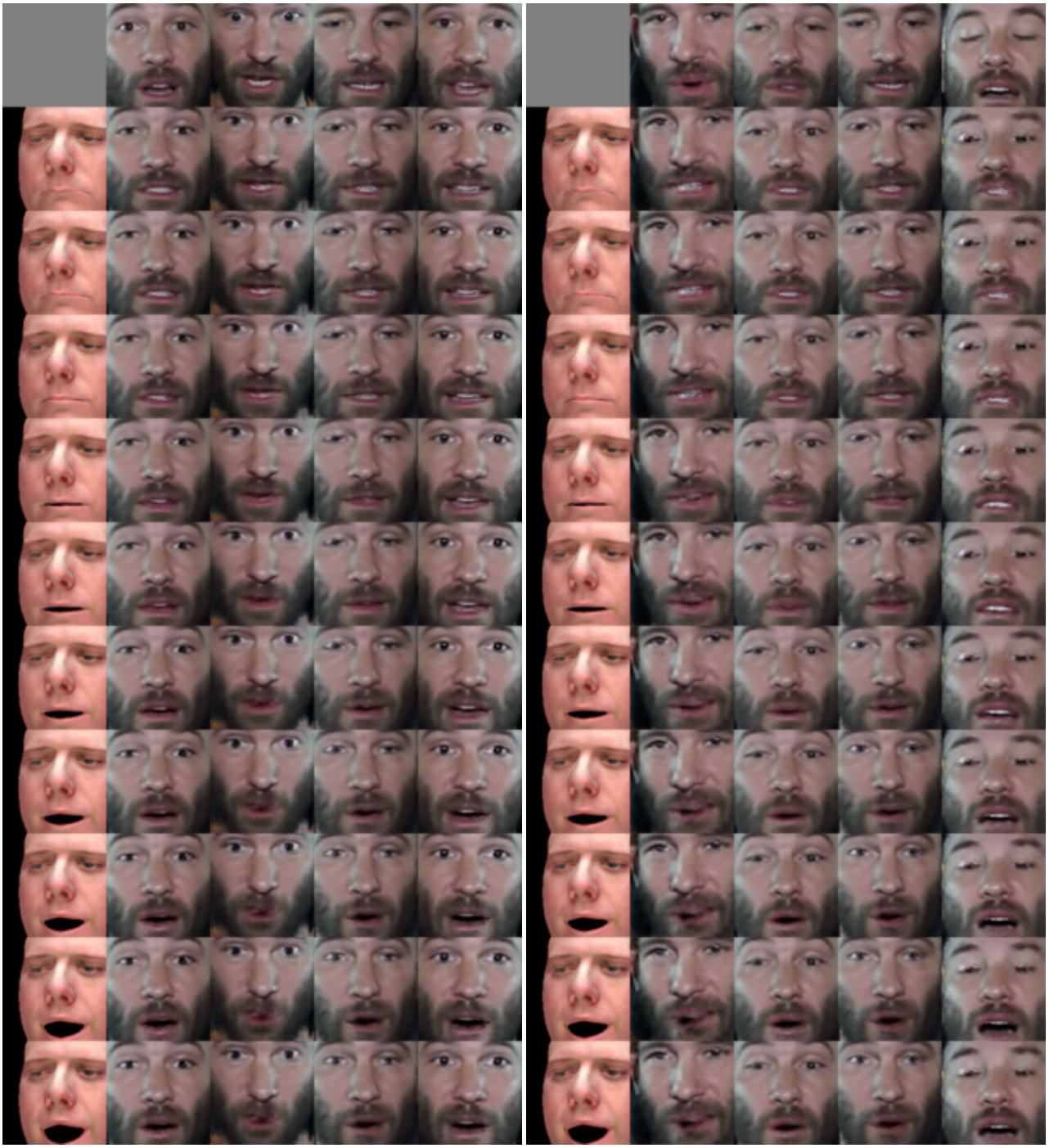


Figure 12: More random examples for an identity from the 300-VW dataset with mouth expression manipulated using our model. Two first and two last rows are “saturated” examples.

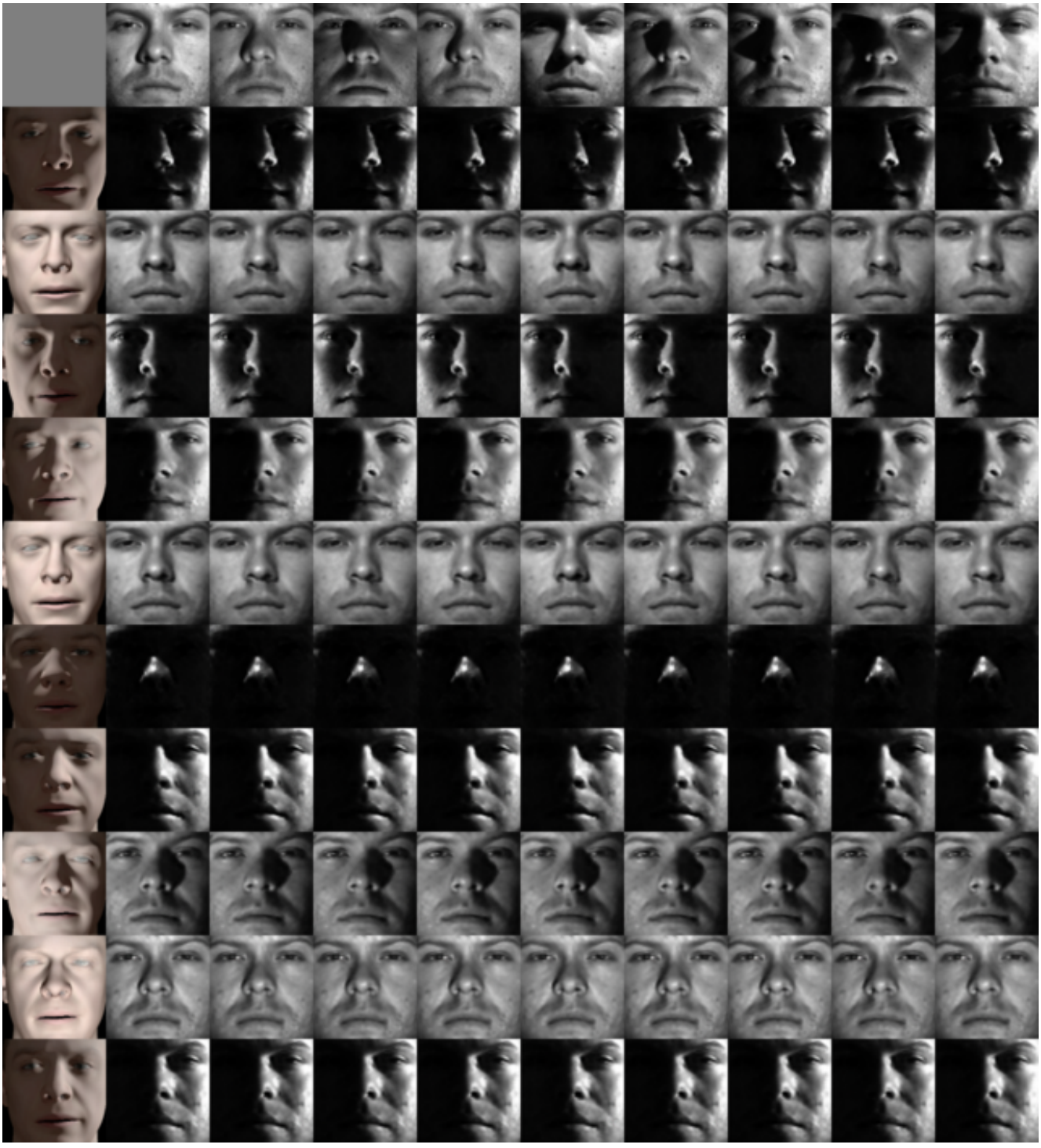


Figure 13: More random examples for a single identity from the YaleB dataset with lighting expression manipulated using our model.

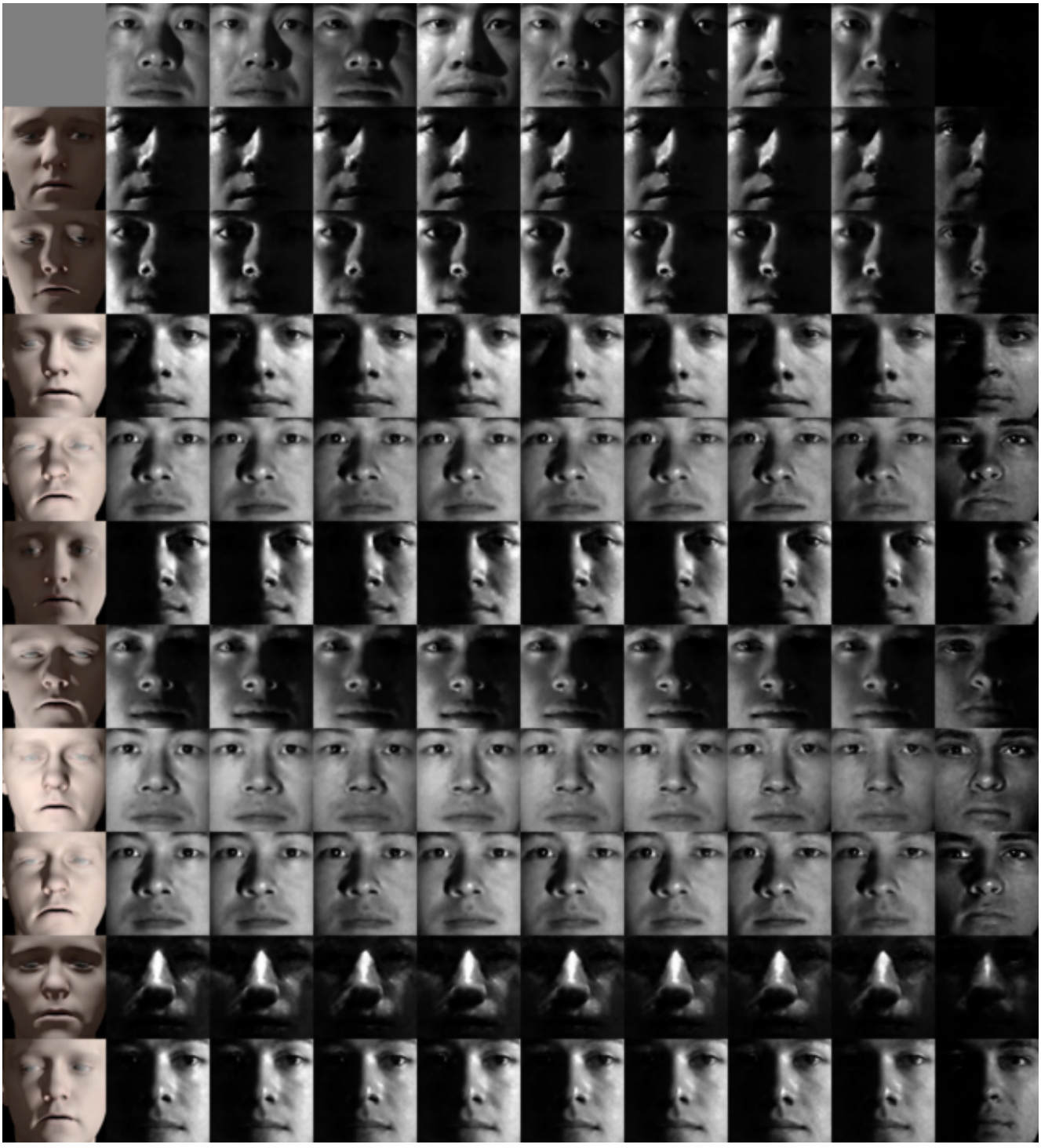


Figure 14: More random examples for a single identity from the YaleB dataset with lighting expression manipulated using our model.

Figure 15: More random examples for a single identity from the YaleB dataset with lighting expression manipulated using our model.

Figure 16: More random examples for a single identity from the YaleB dataset with lighting expression manipulated using our model.

Figure 17: More random examples for a single identity from the YaleB dataset with lighting expression manipulated using our model.

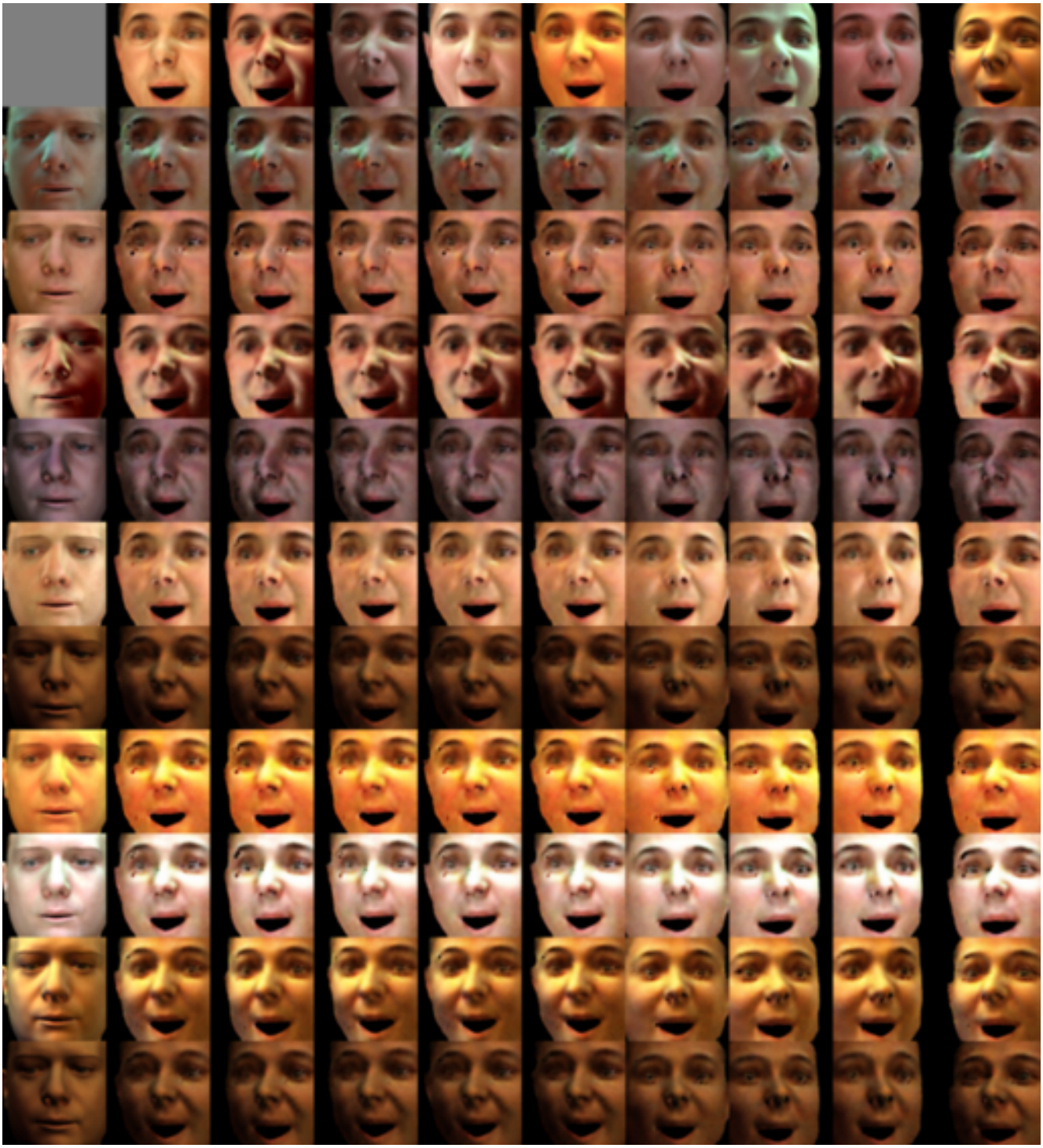
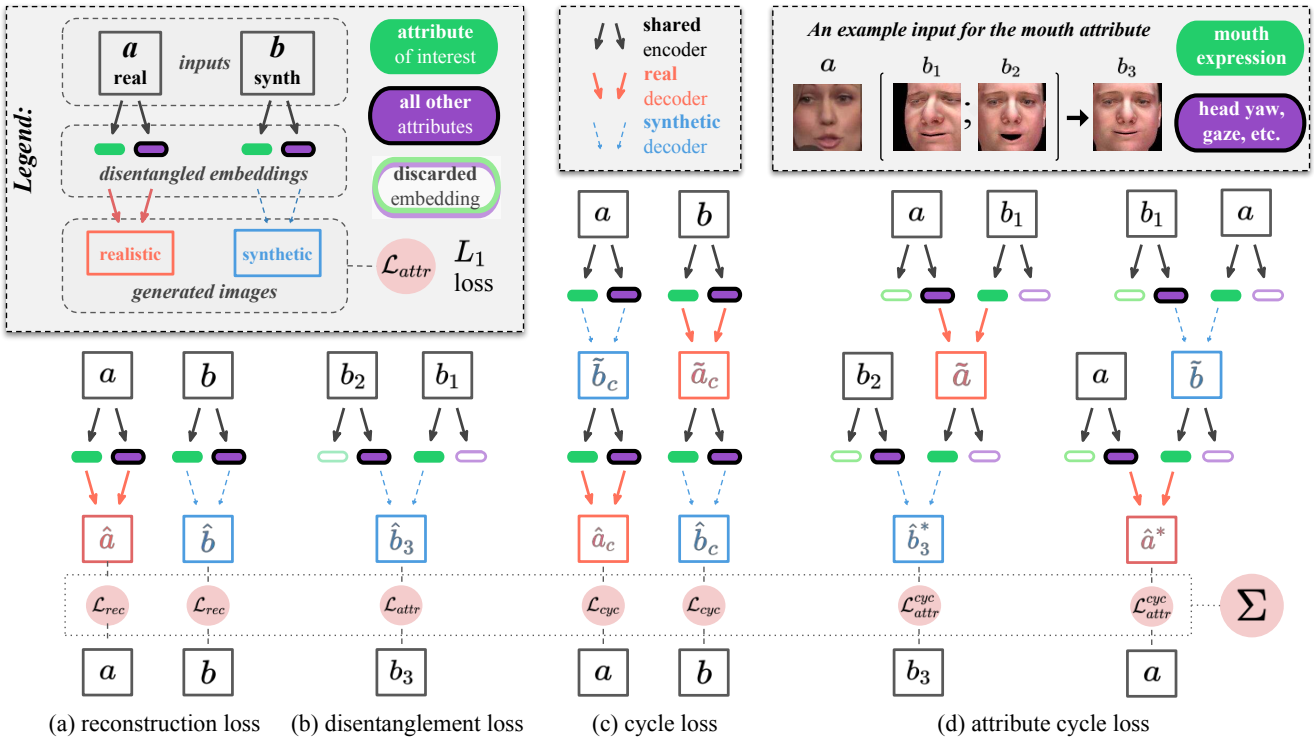


Figure 18: More random examples for disentanglement of spherical harmonics across synthetic identities.



(a) reconstruction loss (b) disentanglement loss (c) cycle loss (d) attribute cycle loss

Figure 19: *(A color blind and print friendly version).* Supervised losses jointly optimized during training of the PuppetGAN. When combined, these losses ensure that the "attribute embedding" (green capsule without a border) affects only the attribute of interest (AoI) in generated images, and that the "rest embedding" (purple capsule with a bold border) does not affect the AoI in generated images. When trained, manipulation of AoI in real images can be performed by replacing their attribute embedding components. Unsupervised (GAN) losses are not shown in this picture. An example at the top right corner illustrates sample images fed into the network to disentangle mouth expression (AoI) from other face attributes in real faces. Section 3 provides more details on the intuition behind of these losses.