A. Model details

A.1. Code and pretrained models

Our code and models used for experiments are available at this url: github.

A.2. GANs

In our experiments we used `https://github.com/rosinality/stylegan2-pytorch` for training GANs for *MPI3D* and *Isaac3D*. See the list of training parameters below.

Parameter	Value
`iter`	800000 for *Isaac3D* and 300000 for *MPI3D*
`batch`	32
`n_sample`	64
`size`	128 for *Isaac3D* and 64 for other *MPI3D*
`r1`	10
`path_regularize`	2
`path_batch_shrink`	2
`d_reg_every`	16
`g_reg_every`	4
`mixing`	0.9
`lr`	0.002
`augment`	False
`augment_p`	0
`ada_target`	0.6
`ada_length`	500000
`latent_dim`	512
`n_mlp`	3
`width`	512
`truncation`	1.0 for *Isaac3D* and 0.7 for *MPI3D*
`mean_latent`	4096
`input_is_latent`	`True`
`randomize_noize`	`False` (for evaluation)

For *FFHQ* we took the pretrained checkpoint `ffhq-res256-mirror-paper256-noaug` provided at `https://github.com/NVlabs/stylegan2-ada`. For *CUB-200-2011* and *Places365* we trained StyleGAN 2 with ADA using the config `auto` from the above repository. We only changed the number of layers in the style network to 2 for the *CUB-200-2011* dataset and to 8 for *Places365*. All the `tensorflow` checkpoints were converted to the `Pytorch` format for the analysis. For these models we used truncation 0.7 when generating samples.

A.3. Attribute regressors

All our attribute regressors have the same structure: a convolutional backbone followed by a multiclass classification head (represented as an MLP of depth 2). We used the following backbones:

- A simple 4-layer CNN for *MPI3D*,
- Randomly initialized ResNet18 for *Isaac3D*,
- ResNet18 pretrained on ImageNet for *FFHQ*, *CUB-200-2011*, *Places365*.

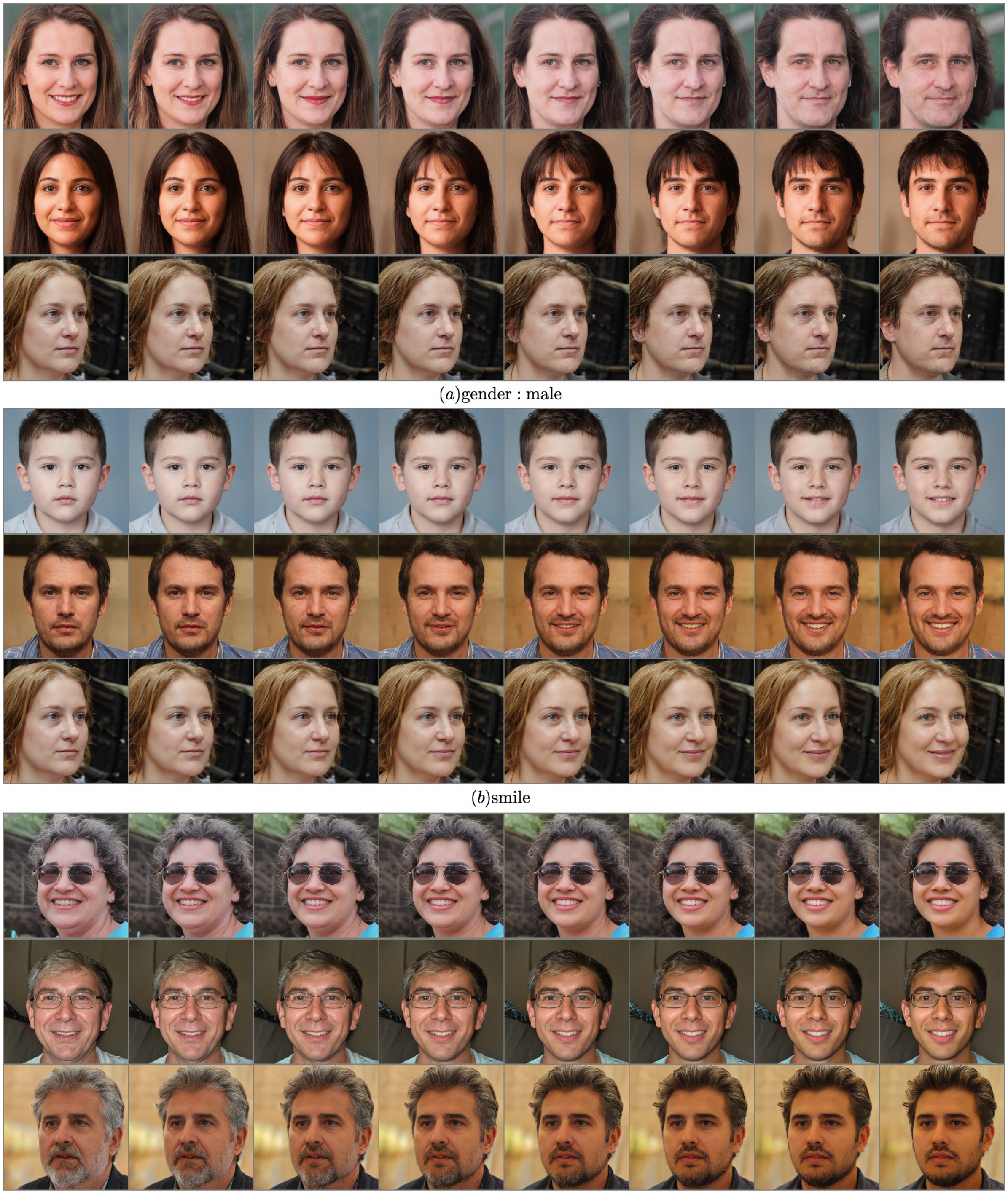
A.4. Disentanglement via pretrained encoders

As described in the main text, we consider an alternative loss formulation to enforce disentanglement. Namely, we take the pretrained FaceNet $\mathcal{F}$ and utilize

$$\mathcal{L}_2(\mathbf{w};\theta) = -\cos\Big(\mathcal{F}[G(\mathbf{w}(T,\theta)], \mathcal{F}[G(\mathbf{w})]\Big),$$

i.e., we simply want to keep the identity of a person after editing. The final loss function takes the form $\mathcal{L} = \mathcal{L}_1 + \alpha\mathcal{L}_2$ where $\alpha = 0.5$ was chosen (we found it not to significantly affect the results). We trained the model using exactly the same experimental setup for the same 13 attributes as in the human evaluation study in Section 5. We utilized the InceptionResnetV1 model pretrained on the VGGFace2 [6] dataset available at https://github.com/timesler/facenet-pytorch. To evaluate the results, we performed human evaluation asking “Which edited image better preserved identity of the original image?”. Interestingly, we found that the FaceNet disentanglement approach provided better results (60% vs 40%), while correctly performing the edits.

B. Additional examples

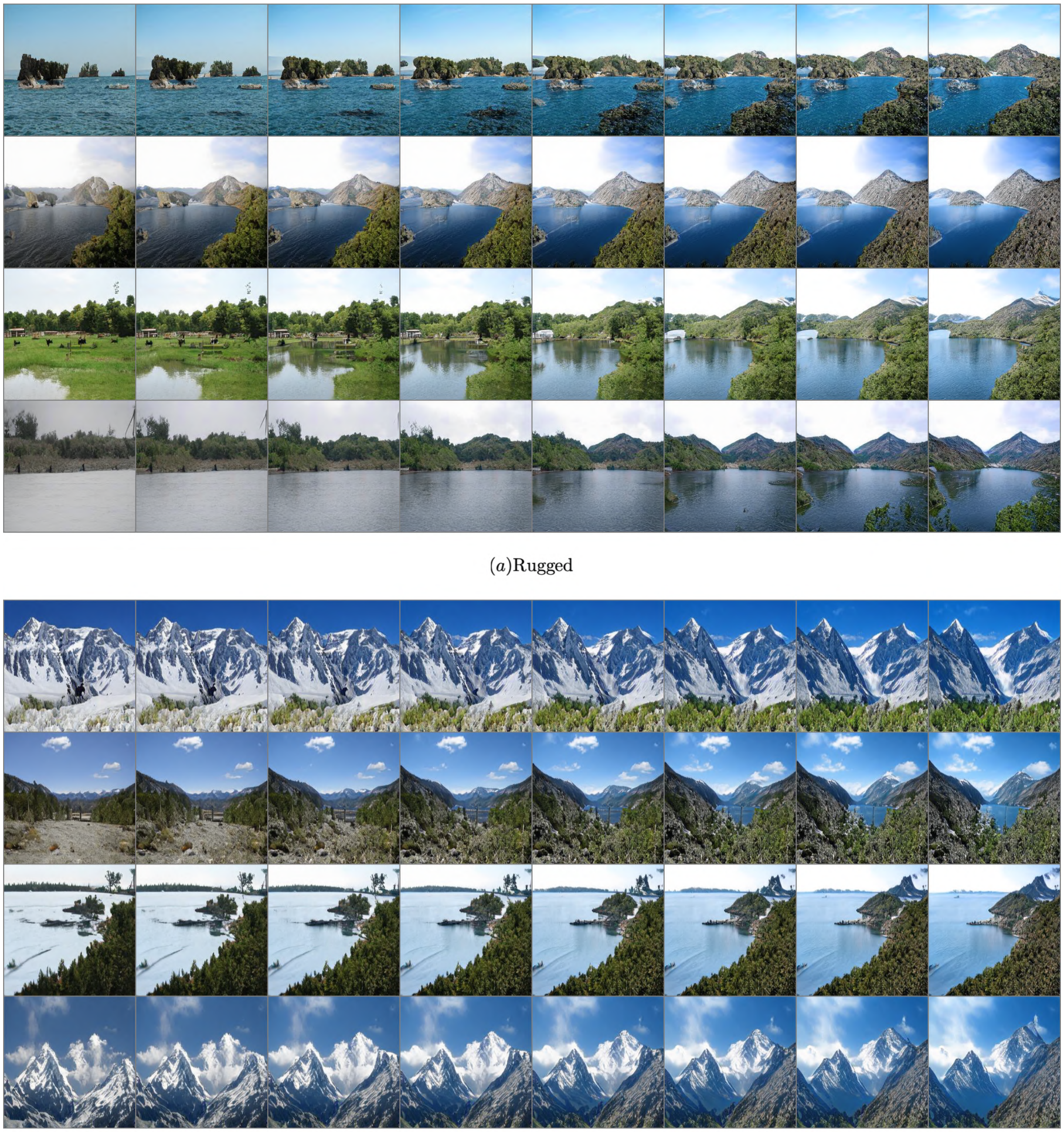


(a)gender : male

(b)smile

(c)age : young

Figure 11. Additional examples of factor manipulations on *FFHQ* made by our Neural ODE method (nonlinear).



(a)Rugged

(b)Lush vegetation

Figure 12. Additional examples of factor manipulations on *Places365* made by our Neural ODE method (nonlinear).

B.1. CUB-200-2011

CUB-200-2011 consists of $11,788$ images of birds with given attribute information covering 312 binary factors of variation [32]. These attributes represent visual features focusing mostly on color, patterns, shapes, *etc*. For *CUB-200-2011* we utilized the existent annotations. We selected the attributes with at least 10% of positive examples for both values, resulting in 109 attributes total. Results for this dataset are available in the supplementary material.

This dataset was particularly challenging to experiment with since the main bulk of its attributes was connected to texture-based features. In our experimental setup, we chose the following factors: the bill shape, the bird size, and one texture attribute describing the primary color; our findings are summarized in Figure 13. We may see a significant difference in bill shape and body size attributes and less visually perceptive difference for feather color. However, we may notice that in several cases, both methods produce entangled modification, *e.g*., changing the bird color simultaneously with size manipulation.

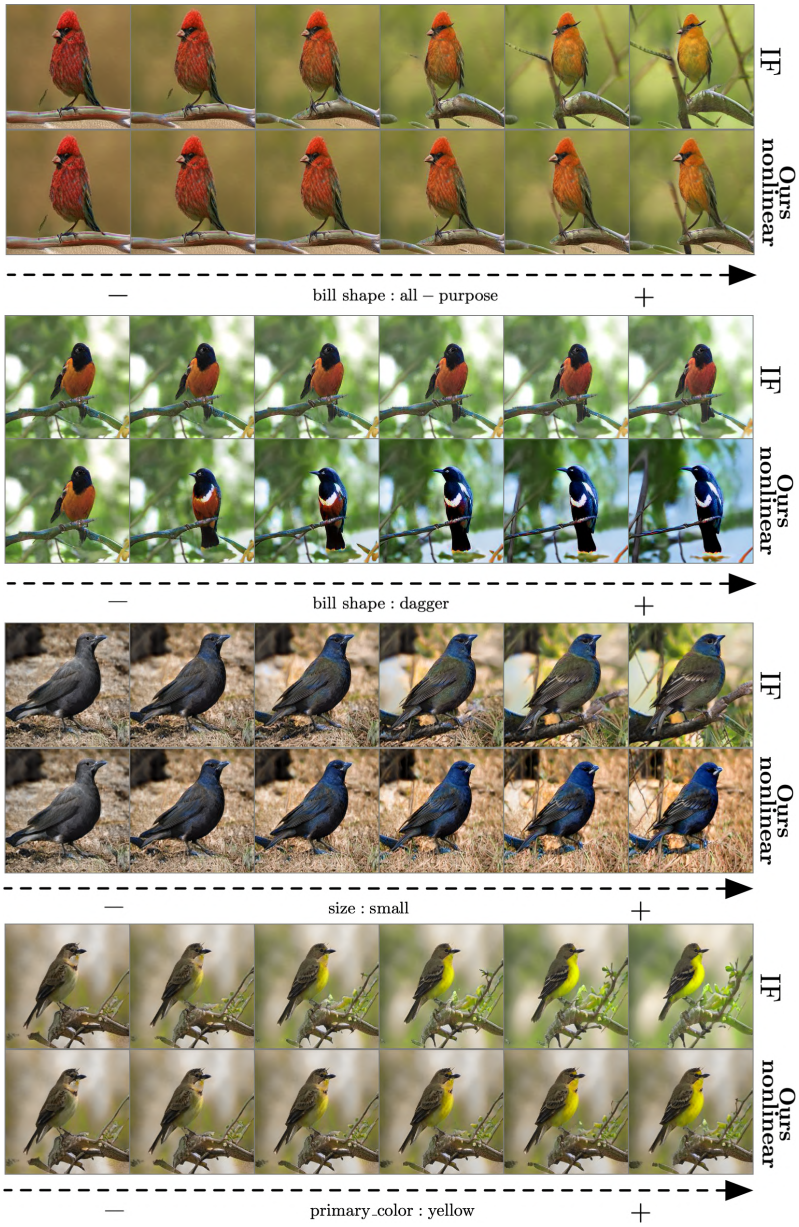
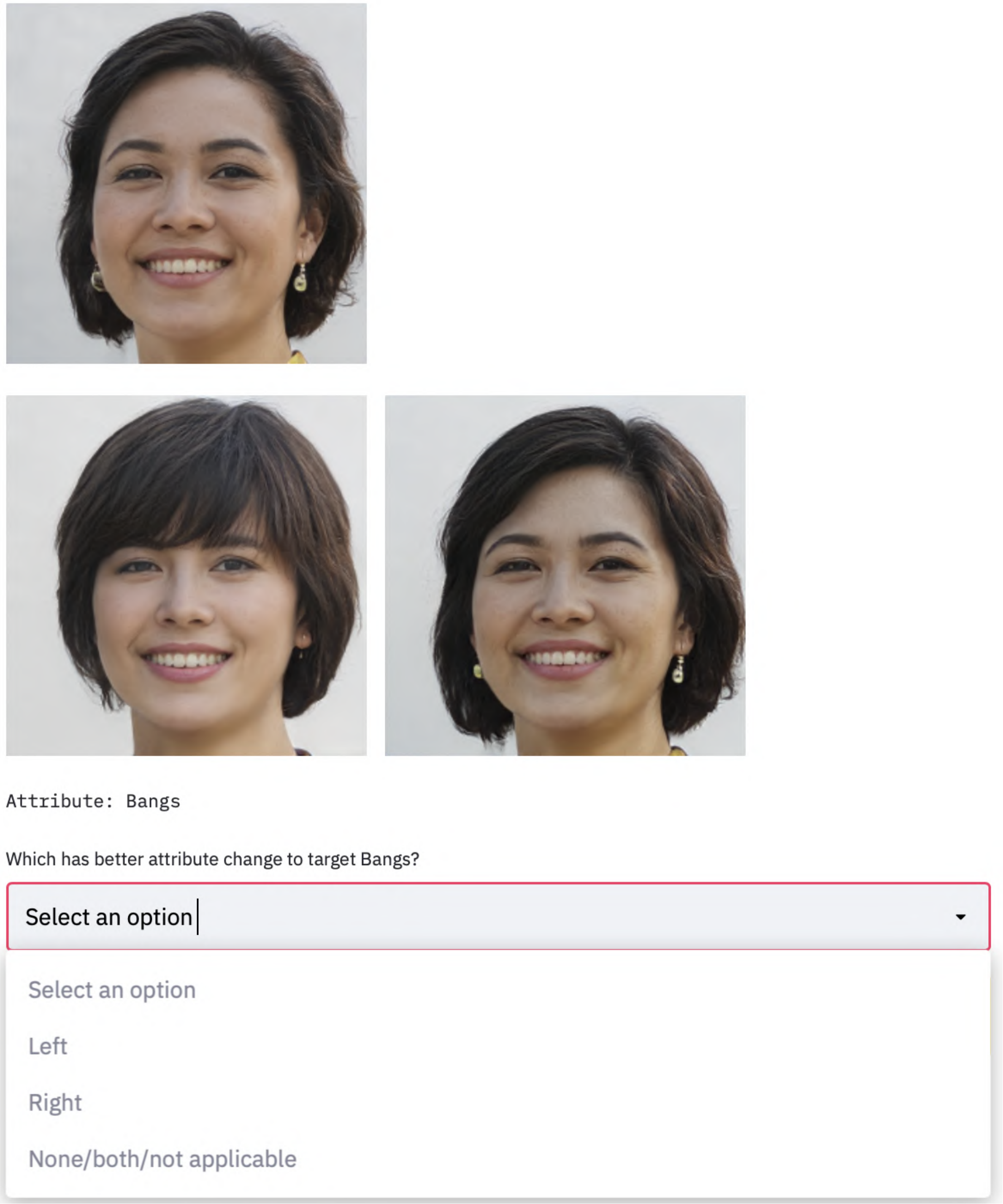


Figure 13. Manipulating attributes corresponding to the birds bill shape, size, and color on the *CUB-200-2011* dataset. While in both cases, we observe some degree of entanglement, linear shifts produced by IF do not change the attribute of choice (*e.g*., for the first three attributes) or change the obtained factor in an unnatural manner (manipulated `primary color:yellow` looks slightly greenish).

C. Human evaluation



```
Attribute: Bangs
```

Which has better attribute change to target Bangs?

Figure 14. The interface of human evaluation questionnaire.

C.1. Attribute breakdown

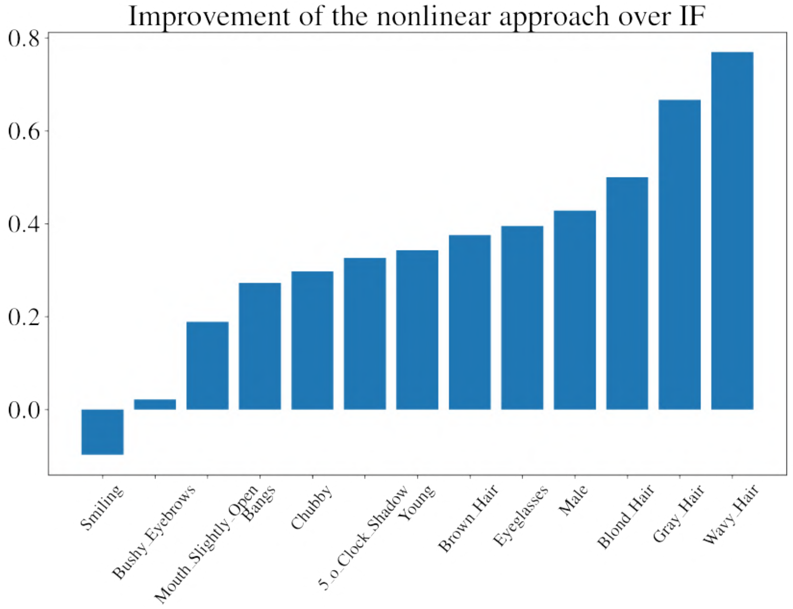


Figure 15. The breakdown of the improvement of the nonlinear model over IF on the *FFHQ* dataset.

D. $\mathcal{H}_{SVD}$ for *Places365*

Here we provide the obtained values of $\mathcal{H}_{SVD}$ for all the attributes on the *Places365* dataset. For this dataset, we do not compute the correlation between the human evaluation breakdown and entropy scores since, in most of the cases, our nonlinear method significantly outperformed IF. We note that the easiest attributes according to Figure 16, such as `cold`, `winter`, `ice`, correspond to simple color based transformations. On the other hand, the most difficult ones such `lush` are mostly 'content' based.

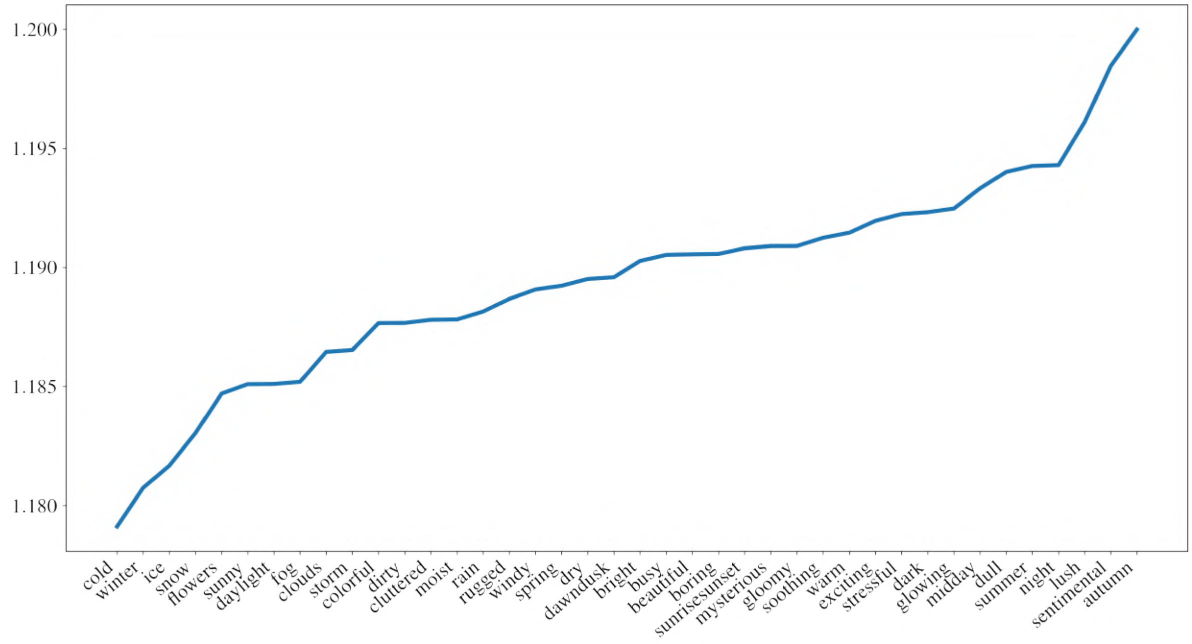


Figure 16. The values of $\mathcal{H}_{SVD}$ for various attributes on *Places365*