

Supplementary Material: MAGECally invert images for realistic editing

Asya Grechka¹²

asya.grechka@lip6.fr

Jean-Francois Goudou¹

jean-francois.g@meero.com

Matthieu Cord²

matthieu.cord@lip6.fr

¹ Meero

Paris, France

² Sorbonne Université

Paris, France

A Details of our LinkNet model

To train our LinkNet model, we first create a dataset of 50,000 (latent, descriptor) training pairs with our pre-trained GAN. Note that with StyleGAN2, we generate the latent vectors using the extended $\mathcal{W}+$ space, meaning that we generate a distinct latent vector for each style block of StyleGAN2. Generating data in this way allows for better predictive capacities when we effectuate our inversion optimization, since this is also optimized in $\mathcal{W}+$. The descriptors are obtained by evaluating our pre-trained classifier F on the generated images. We use the predicted binary labels for the experiments on faces, and the predicted probability vectors for the experiments on cars.

Our LinkNet model is a 1-layer linear model mapping the flattened latent vector to the image descriptor vector with a sigmoid activation function. We used a learning rate of $5e^{-3}$ and the Adam [1] optimizer with the default parameters. We trained the model for 10 epochs using the binary cross-entropy loss and evaluated on a separate validation set, achieving a 89% accuracy for face attributes, and a 95% accuracy for car models. It should be noted however, that the generated cars were often classified the same due to high discrepancy between real and generated cars (generated cars generally don't have distinctive logos).

B Supplementary Visual Results

In terms of assessing visual quality, human evaluation is still the gold standard [2, 3]. We have thus provided abundant uncensored visual results using our method and comparing it to Image2StyleGAN++ [4] as well as to our ablated method without the MAGEC loss.

B.1 Supplementary Facial Edits

Fig. 2, 3, 4, and 5 show examples with the respective four editing methods: InterfaceGAN [5], StyleFlow [6], GANSpace [7], and random interpolations [8]. When viewing the results,



(a) Average latent vector from the pre-trained StyleGAN2 on LSUN cars.



(b) Average latent vector from the pre-trained StyleGAN2 on FFHQ.

Figure 1: Average latent vectors from two different pre-trained StyleGANs

take extra care to notice the reconstructions (compared to the original images) as well as the result of the intended edit operation (with respect to the original image). Figures should ideally be viewed zoomed and in color. Note that ambiguous edit operations like *gender*, *expression* and *age* should flip the attribute in question (for example, *age* edit means young turns to old, and vice versa). The following general observations can be made:

- Image2StyleGAN++ [10] produces very accurate reconstructions, but edits are often of abysmal quality.
- Ablating the MAGEC loss leads to worse reconstructions.
- Ablating the MAGEC loss produces edits that are of good-quality, but often don't respect the edit intention (for example, the *glasses* edit may not make any noticeable change, despite producing a high-quality image) nor fidelity to the input image.
- Our MAGEC loss gives accurate reconstructions, but also produces the intended edits that are sharper, less noisy, and of higher-quality.

B.2 Experiments on Cars Dataset

We applied our method onto images of real cars. Visual results can be seen in Fig. 6.

Configurations For our feature extractor F , we use a pre-trained cars classifier [9] which classifies the image into one of 196 car models. We use GANSpace [5] as our editor e . It's worth noting that GANSpace does not allow edits of the car model, so our MAGEC loss supervises our training in a weaker fashion than before. Here, the modified “ground-truth” feature vector is simply the image descriptor vector, since the car model should not change with GANSpace edits. Finally, since the “average” latent vector from the pre-trained *cars* StyleGAN2 is a poor representation of a car (see Fig. 1), we train an encoder E which predicts a latent vector from an image. This was used to initialize the latent vector for further optimization. We used a pre-trained ResNet-50 model [2] as our backbone, and modified the last layers to output a latent vector. Generated data was used for training.

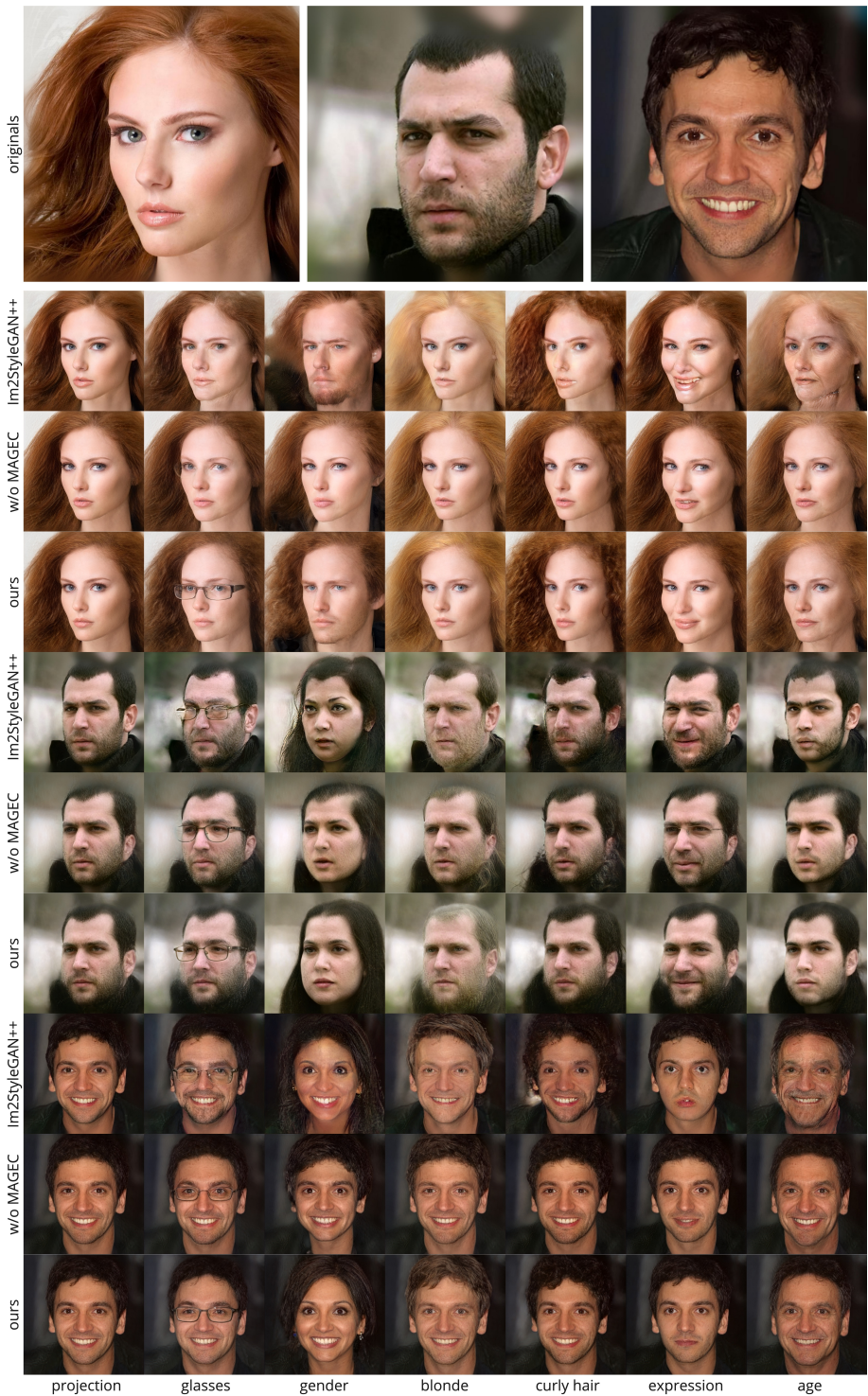


Figure 2: InterfaceGAN [9] edits using various inversion methods. Our method gives the intended edits with high-quality results. Best viewed zoomed and in color.

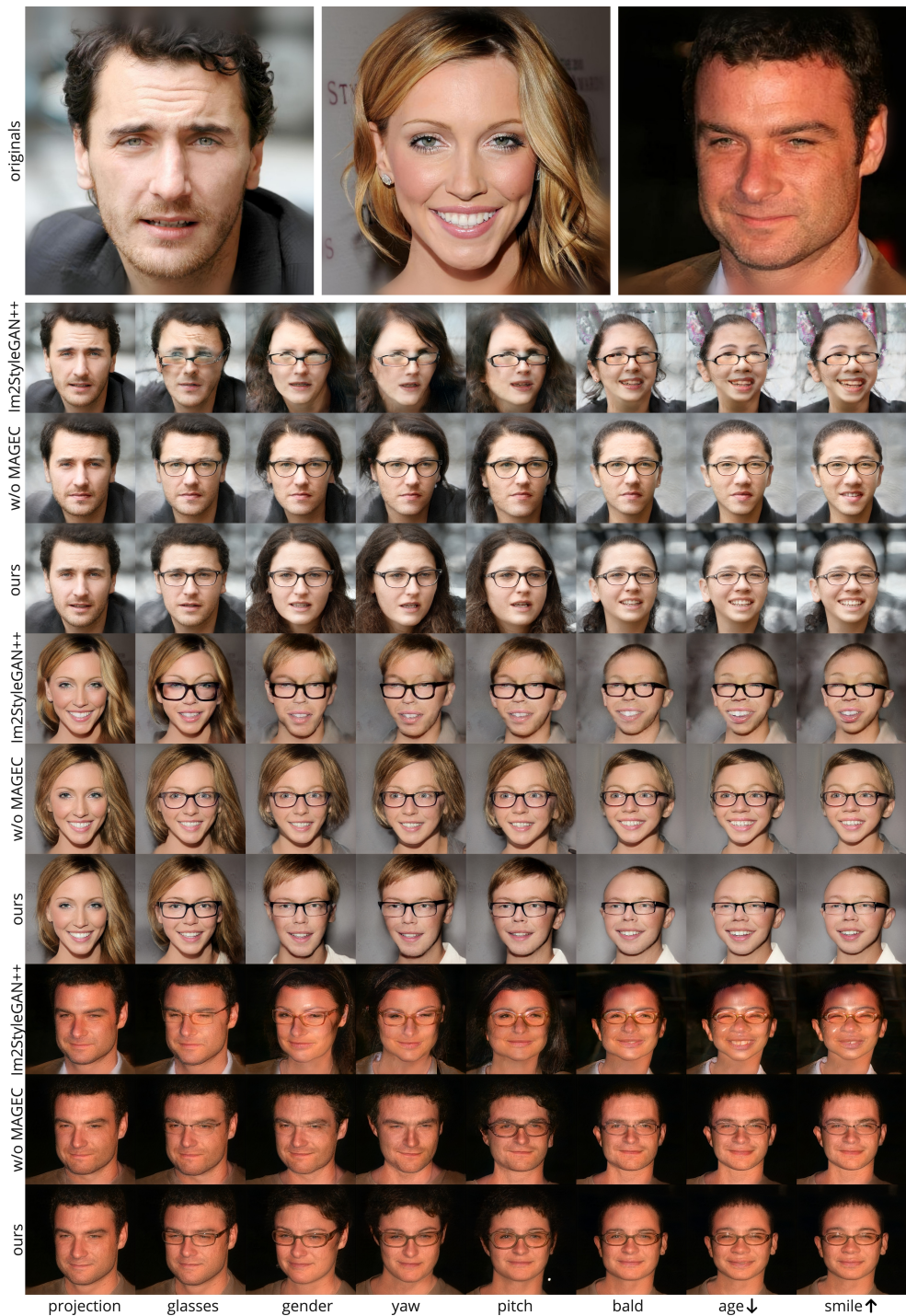


Figure 3: StyleFlow [14] edits using various inversion methods. Remark that here, the edits should be **cumulative**. Our MAGEC loss helps to produce accurate reconstructions as well as the intended edits with high-quality results.



Figure 4: GANSpace [1] edits using various inversion methods. Image2StyleGAN++’s inversion method produces accurate reconstructions, but distorted and low-quality edits. Using our MAGEC loss greatly helps with reconstruction, but also helps to produce the intended change (notice *male / female* edits in particular). Best viewed zoomed and in color.

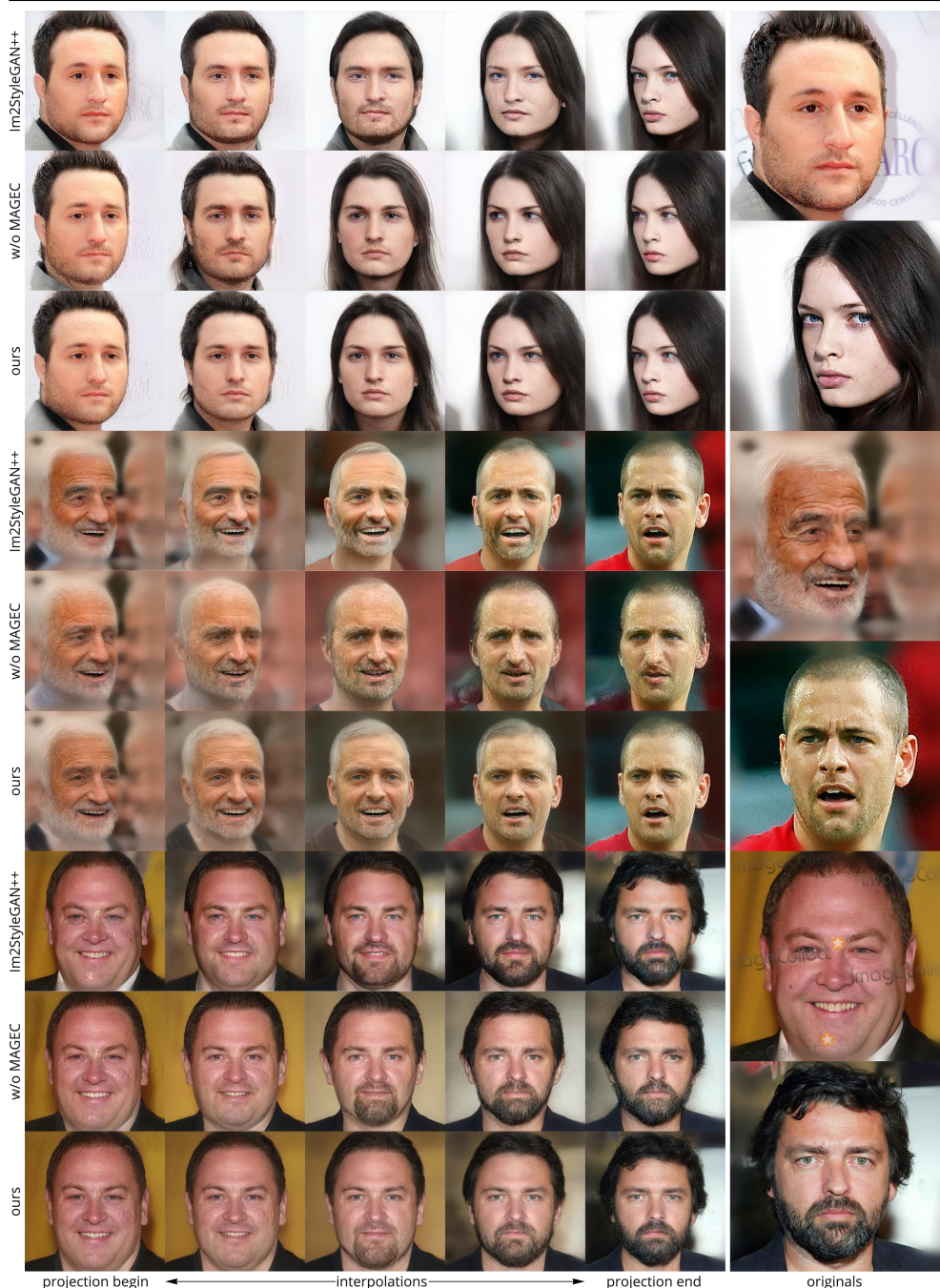


Figure 5: Interpolations [8] using various inversions. Take extra notice of the reconstructions, where our MAGEC loss clearly helps. In the first edit, only our edit gives a beard which clearly progressively disappears (rather than increasing then decreasing), suggesting stronger semantic information in the inverted latent. In the last case, notice the gradual background changes in the third set with our method, compared to the harsher change of Im2StyleGAN++’s edit. Best viewed zoomed and in color.

Training protocol We initialize z with $z_0 = E(x_{real})$. Training is again effectuated in two parts. First, we optimize for 50 steps using $\lambda_{MAGEC} = 5e^{-2}$ and $\lambda_{MSE} = \lambda_{LPIPS} = 5e^{-1}$ with a learning rate of 0.08. Then, we decrease $\lambda_{MAGEC} = 5e^{-8}$ with a learning rate of 0.01 and train for 50 more steps. We use the Adam optimizer for all the training[2].

Datasets and Editor We evaluate using random images from the Stanford Cars test set [8], not used to train F . We use the GANSpace [9] editing method to evaluate editability.

Results and Interpretation As expected, Image2StyleGAN++’s projection leads to distorted and inaccurate edits. When comparing our method to the ablated method, we can see that MAGEC helps editing and reconstruction. Notice the rotation operations for the first and second cars in Fig. 6. The red car preserved the “sports car” look while the white car similarly preserved the *Audi* logo. Finally, the last rows show that we were correctly able to reconstruct the *BMW* model as well as preserving it during edits.

We used a very general pre-trained classifier F which was rather unrelated to the edits of the GANSpace editor that supervised our training. Moreover, our method assumes that StyleGAN’s latent code can predict the specific car model, a strong assumption, especially considering that purely generated car images rarely have a clear logo. It is more likely that the latent code encodes some sort of “shape” which roughly predicts the car model with LinkNet. Despite these limits, we can see that adding this simple MAGEC loss using an arbitrary auxiliary classifier does indeed improve editing and reconstruction capacity for many cases, giving high promise to the capacity and flexibility of our method.

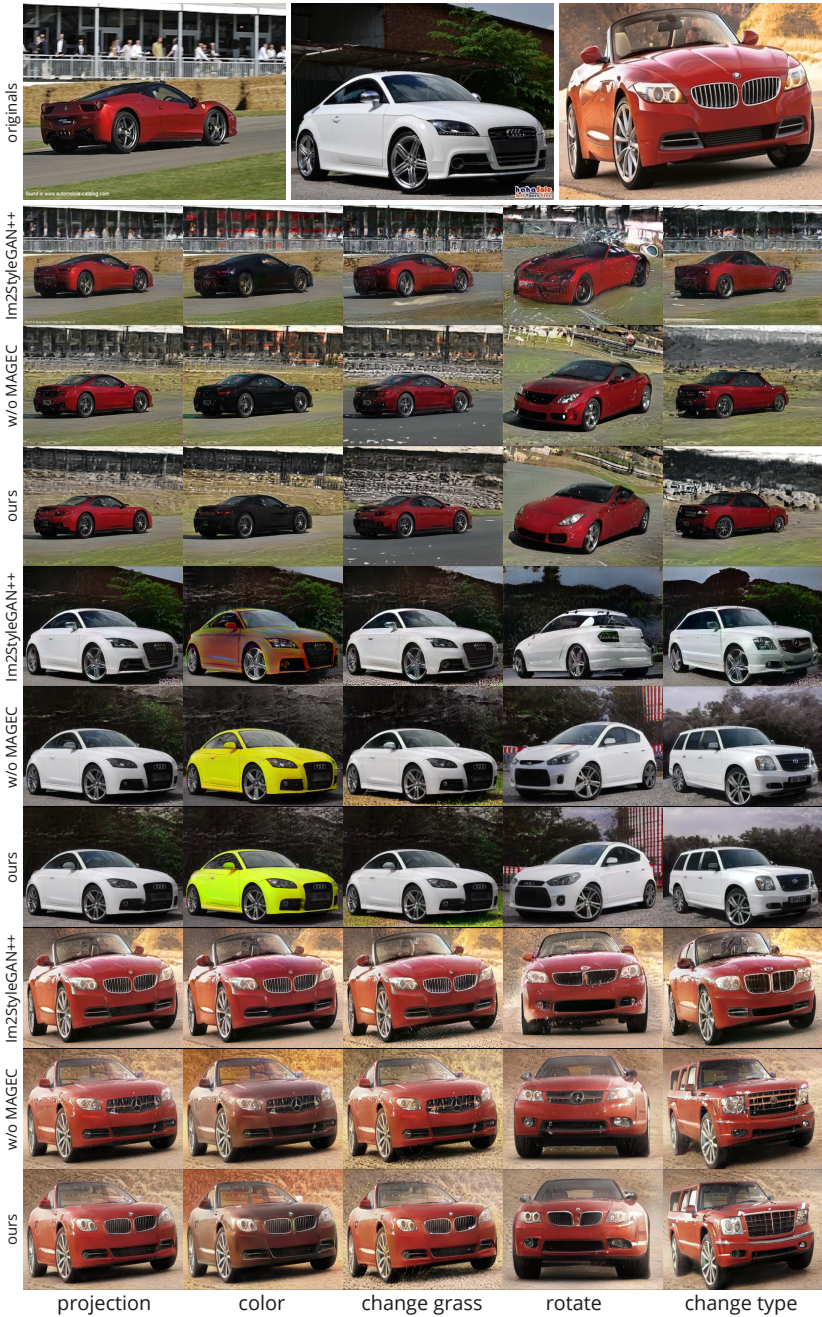


Figure 6: GANSpace [9] edits with various inversions. Image2StyleGAN++’s method produces good reconstructions but distorted edits. Our method helps in preserving the car model during reconstruction and edits. Remark the top red car when rotated: our method preserves the *sports car* style. Remark that the *Audi* logo of the white car is also conserved when rotating. Finally, the bottom red car reveals that our method consistently maintains the correct car model (*BMW*) during reconstruction and edits. Best viewed zoomed and in color.

References

- [1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *CVPR*, 2020.
- [2] Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Trans. Graph.*, 2021.
- [3] fishman2008. fishman2008/stanford-cars-classification, Jun 2019. URL <https://github.com/fishman2008/stanford-cars-classification>.
- [4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [5] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. In *NeurIPS*, 2020.
- [6] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.
- [7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [8] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV Workshops*, 2013.
- [9] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, 2020.
- [10] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *arXiv preprint arXiv:2102.02766*, 2021.
- [11] Sharon Zhou, Mitchell Gordon, Ranjay Krishna, Austin Narcomey, Li F Fei-Fei, and Michael Bernstein. Hype: A benchmark for human eye perceptual evaluation of generative models. In *NeurIPS*, 2019.