
Reproducibility study for "Explaining in Style: Training a GAN to explain a classifier in StyleSpace"

Anonymous Author(s)

Affiliation

Address

email

Reproducibility Summary

1

2 Scope of Reproducibility

3 This work aims to reproduce Lang et al.'s StyleEx [9] which proposes a novel approach to explain how a classifier makes
4 its decision. They claim that StyleEx creates a post-hoc counterfactual explanation whose principal attributes correspond
5 to properties that are intuitive to humans. The paper boasts a large range of real-world practicality. However, StyleEx
6 proves difficult to reproduce due to its time complexity and holes in the information provided. This paper tries to fill in
7 these holes by: i) re-implementation of StyleEx in a different framework, ii) creating a low resource training benchmark.

8 Methodology

9 We use their provided python notebook to confirm their *AttFind* algorithm. However, to test the authors' claims, we
10 reverse engineer their architecture and completely re-implement their train algorithm. Due to the computational cost of
11 training, we use their pre-trained weights to test our reconstruction. To expedite training, a smaller resolution dataset is
12 used. The training took 9 hours for 50,000 iterations on a Google Colab Nvidia K80 GPU. The hyperparameters are
13 listed in the proceedings.

14 Results

15 We reproduce the StyleEx model in a different framework and test the *AttFind* algorithm, verifying the original paper's
16 results for the perceived age classifier. However, we could not reproduce the results for the other classifiers used, due
17 to time limitations in training and the absence of their pre-trained models. In addition, we verify the paper's claim of
18 providing human-interpretable explanations, by reproducing the two user studies outlined in the original paper.

19 What was easy

20 The notebook supplied by the authors loads their pre-trained models and reproduces part of the results in the paper.
21 Furthermore, their algorithm for discovering classifier-related attributes, *AttFind*, is well outlined in their paper making
22 the notebook easy to follow. Lastly, the authors were responsive to our inquiries.

23 What was difficult

24 A major difficulty was that the authors provide only a single pre-trained model, which makes most of the main claims
25 require training code to verify. Moreover, the paper leaves out information about their design choices and experimental
26 setup. In addition, the authors do not provide an implementation of the models' architecture or training. Finally, the
27 practical audience is limited by the resource requirements.

28 Communication with original authors

29 We had modest communication with the original author, Oran Lang. Our discussion was limited to inquiries about
30 design choices not mentioned in the paper. They were able to clarify the encoder architecture and some of their
31 experimental setup. However, their training code could not be made available due to internal dependencies.

32 1 Introduction

33 As the field of machine learning (ML) develops and its algorithms become more prevalent in society, concerns on
34 the explainability of black-box models become pivotal. For problems that have a high societal impact, there is
35 understandable apprehension towards trusting models that do not provide justification. For applications such as medical
36 imaging and autonomous driving, there is a need for some level of human supervision. Even if a model has high
37 performance, such as neural networks, without the ability for human interpretation, its use will be limited.

38 In order to gain trust in systems powered by ML models, the models need to be interpretable and explainable. The
39 two concepts are regularly used interchangeably, yet have subtle differences. Interpretability is the degree to which
40 humans can understand the cause of a decision [10]. Deep neural networks, such as classifiers are often perceived as
41 “black boxes” whose decisions are opaque and hard for humans to understand. Explaining the decision of classifiers
42 can reveal model biases[8] and also provide support to downstream human decision-makers. On the other hand,
43 explainability is linked to the internal logic of a model. It focuses on explaining the data representation within that
44 network. Explainability implies interpretability, however, the implication is not bidirectional.

45 In recent years, there has been increasing attention to the field of explainability of deep network classifiers. Among the
46 various ways of explanations, counterfactual explanations are gaining increasing attention [11, 2, 3]. To discover and
47 visualize, the attributes used to generate counterfactual explanations, a natural candidate is generative models. In [13]
48 they observed that StyleGAN2 [7], tends to contain a disentangled latent space (i.e., the “StyleSpace”) which can be
49 used to extract individual attributes. The authors based their proposed methodology [9] on this observation. Though
50 [12] propose a similar architecture, Lang et al. assert that by integrating the classifier into the training of StyleEx they
51 can obtain principal attributes that are specific for the classification task. Additionally, they suggest that StyleEx can be
52 applied to a large variety of complex, real-world tasks, which makes its replicability especially intriguing.

53 Our work aims to reproduce the claims made by Lang et al. and confirm their results. Their paper reports in detail many
54 experiments to justify their claims, but does not dive into their experimental setups for architecture and training. Since
55 not all the information needed is available without contacting the authors, we argue that this paper cannot be considered
56 *fully* reproducible.

57 To remedy the holes in reproducibility and aid future work that builds on or applies StyleEx, we build their proposed
58 architecture and training algorithm, after correspondence with the authors.

59 2 Scope of reproducibility

60 To determine the scope of reproduction, we quote Lang et al.’s main claims:

61 Claim 1 : [They] propose the StyleEx model for classifier-based training of a StyleGAN2, thus driving its StyleSpace to
62 capture classifier-specific attributes

63 Claim 2 : A method to discover classifier-related attributes in StyleSpace coordinates, and use these for counterfactual
64 explanations.

65 Claim 3 : StyleEx is applicable for explaining a large variety of classifiers and real-world complex domains. [They]
66 show it provides explanations understood by human users.

67 To reproduce Claim 2, a trained model and the *AttFind* algorithm are sufficient; both of which are contained in the
68 authors’ notebook. Claim 1 requires a network trained conditioned on a classifier and a network trained without, while
69 Claim 3 requires multiple networks trained on multiple domains. However, to train these models, the architecture and
70 training code is necessary; which, as stated previously, are not open source or thoroughly documented. In addition, the
71 computational cost to train the models is expensive. Thus, to verify these claims our goals will be to:

- 72 • Reconstruct their architecture and port the pre-trained weights in PyTorch
- 73 • Evaluate whether the principal attributes we obtain correspond to the same features using their pre-trained
74 weights
- 75 • Retrain on datasets of smaller images and analyze the scalability of their method using fewer training steps
76 and smaller architecture
- 77 • Conduct two user studies on visual coherence and distinctness to prove that attributes extracted are interpretable
78 by humans

79 To ease reproduction for future work, we built the StyleEx architecture into a different framework, to get a deeper
80 understanding of the model, and become more equipped to tackle training. As an addition, this contribution allows
81 StyleEx to be more accessible for classifiers trained in PyTorch.

82 3 Background

83 There have been many attempts to extract explanations from classifiers most of which utilize heatmaps of important
84 features. However, heatmaps struggle to visualize features that are not spatially localized such as color or shape. Rather
85 than identifying areas of interest, one can provide an explanation through a "what-if" example where the features are
86 slightly altered. These forms of justification have been found to be more interpretable for non-localized features, and
87 are known as *counterfactual* examples. However, it often requires domain knowledge and handcrafting examples to be
88 appropriate. Lang et al. automate this and utilize machine learning to generate realistic counterfactual examples. This
89 section will outline how they claim to achieve this with their two major contributions, StyleEx and AttFind.

90 3.1 StyleEx

91 The way Lang et al. generate examples is through a neural generative model they dubbed StyleEx. StyleEx expands on
92 the popular generative adversarial network StyleGAN v2, which generates realistic images by creating competition
93 between two networks.

94 One of these two networks, referred to as the Generator, G , attempts to generate a realistic image. To this end, the
95 generator samples from a latent space, $z \in R^n$, with a simple probability distribution such as $z_i \sim \mathcal{N}(0, 1)$. The
96 sampled vector is pushed through a series of linear layers called *mapping network* to create a new latent vector, w ,
97 with a more complex probability distribution. This vector is used as input to a number of *StyleBlocks* based on the
98 logarithmic resolution of the image. StyleBlocks consist of an affine transform and an upsampling layer. The affine
99 transform, A_r , maps w to yet another vector s_r , where r denotes the block number or resolution of the block. This
100 concatenation of all s_r is known as the style, or *attribute*, vector, and the space that it spans is known as the StyleSpace.
101 The attribute space is emphasized due to recent observations that it is less entangled than the latent space. The second
102 network is the discriminator, D . This network is trained to differentiate between fake and real images. This forces the
103 generator to slowly improve its creation of fake images. In this way, the discriminator can be seen as an adaptive loss
104 function.

105 The flaw with the direct application of StyleGAN is that it generates from a random latent space. To explain a
106 classification, we would like to condition it on a particular image of interest, but StyleGAN has no mechanism for
107 extracting the attributes of an image. To fix this, Lang et al. added a third, encoding network to StyleEx, E . Rather than
108 using a randomly sampled z and the mapping network to obtain w , StyleEx uses the output of the encoder, $z = E(x)$,
109 where x is an input image. StyleEx adds an extra loss condition that the reconstructed image, $x' = G(E(x))$, should be
110 approximately x . Thus, the encoder combined with the affine transformations allows us to extract the attributes of an
111 input image.

112 StyleEx is not unique in adding an encoder to the StyleGAN to explain a classifier. However, other methods do not
113 include the classifier in the training of the network. StyleGAN incorporates the classifier into training by appending its
114 output to the encoded z vector. This results in another loss condition $C(x) \approx C(x')$.

115 3.2 AttFind

116 Once the attributes of an image have been extracted, a counterfactual explanation can be achieved from the attributes
117 with the most affect on a classifier's decision. Lang et al. propose attribute find (AttFind) to discover the most influential
118 attributes. The algorithm adjusts all the attributes one at a time by a fixed amount d and observes their effect on the
119 classification Δc_s . The k attributes with the highest Δc create a *local* explanation for an image's classification. To
120 approximate a *global* explanation, the principal attributes are determined by the mean Δc across images in a set.

121 4 Reproduction approach

122 Reimplementing StyleEx has been split into two main tasks to ease resource requirements. The first task consists
123 of rebuilding StyleEx in a different framework; the second is training the model from scratch. In this section, we
124 discuss how we rebuilt the model architecture and training process. Additionally, we include details obtained through
125 correspondence missing from the original paper.

126 4.1 Model descriptions

127 To test [Claim 1](#) and [Claim 3](#), at least two models are necessary. Because only one pre-trained model is available, a
128 new model needs to be trained. However, this is computationally expensive as it builds on StyleGAN¹. This led us to
129 evaluate reproducibility in two ways. Firstly, we recreate their architecture in PyTorch, using their pre-trained weights
130 to bypass the training limitation. Secondly, we attempt to train a model from scratch using less complex datasets with
131 smaller resolutions to verify claims requiring multiple models. In the following sections, we explain how we reconstruct
132 the StyleEx architecture and training process.

133 4.1.1 Rebuilding StyleEx

134 The author’s notebook includes a TensorFlow StyleEx pre-trained on the FFHQ[6] dataset to find the attributes most
135 influential in age classification.

136 Taking advantage of the pre-trained model’s raw parameters, we reverse engineer the architecture of each component of
137 StyleEx and implement it in PyTorch. Subsequently, the pre-trained weights are transferred into the reconstructed StyleEx
138 to confirm the correct implementation of the structure. Transferring the pre-trained parameters from a TensorFlow
139 model to a PyTorch model turned out to be challenging and non-trivial.

140 We start by building the architecture of the MobileNetV1 [5] classifier, as described in the summary of their model,
141 in both TensorFlow and PyTorch. We follow this approach so that we can compare how the results of each layer
142 differ, depending on the framework. We notice that for the 2D convolutional layers PyTorch and TensorFlow pad
143 the images differently, leading to different results. To address this, we add a [ConstantPad2D](#) layer in our PyTorch
144 architecture before each convolution with a stride of 2. In addition, we change the default hyperparameters of PyTorch’s
145 [BatchNorm2D](#) to match the corresponding TensorFlow defaults.

146 The next step is to follow the same procedure for the encoder and the StyleGAN components. We use the official
147 StyleGAN2 implementation in PyTorch by NVlabs[7] and modify the initial architecture to align with the StyleEx model.
148 In particular, instead of only using the encoding of an image X as input to the generator, we also concatenate the
149 classifier’s output logits. Additionally, their generator returns the StyleSpace which contains classifier-specific attributes.
150 For the encoder, we use the same architecture as StyleGAN2’s discriminator. Finally, we transfer the pre-trained
151 weights, to our components.

152 The last step is to load the rebuilt StyleEx model in the provided notebook to confirm that the conversion of the models is
153 successful and reproduce the results provided in the notebook.

154 4.1.2 Training the model

155 Lang et al. asserted that StyleEx works for a wide range of classifiers and datasets. The results they show in their paper
156 are all with high-resolution images. The high resolution comes with a high computational cost as StyleEx is built on top
157 of a StyleGAN. High-resolution StyleGANs can take over a month to train on a single GPU system. To tackle this, we
158 train our model on a low-resolution MNIST dataset. In this way, we investigate whether their model works well on
159 low-resolution datasets and relieve computational requirements.

The training is as outlined in their paper. The loss function for the StyleEx model is broken into seven parts: \mathcal{L}_x , \mathcal{L}_w ,
 \mathcal{L}_{LPIPS} , \mathcal{L}_{adv} , \mathcal{L}_{PLR} , \mathcal{L}_{KL} , and the \mathcal{L}_{GP} . \mathcal{L}_x is the L1 loss between the real image, x , and the reconstruction of that
image, $G(E(x))$. \mathcal{L}_{LPIPS} is the Learned Perceptual Image Patch Similarity (LPIPS) of the two images. This loss is a
metric other than raw pixel value error for the similarity between two images. \mathcal{L}_w is the L1 loss between the encoding
of the original image, $w = E(x)$, and the encoding of the reconstructed image $w' = E(G(E(x)))$. Collectively, these
three losses make up the reconstruction loss, \mathcal{L}_{rec} , ie,

$$\mathcal{L}_{rec} = \mathcal{L}_w + \mathcal{L}_x + \mathcal{L}_{LPIPS}.$$

160 In the implementation, each loss term in \mathcal{L}_{rec} had a weighting coefficient to even out the magnitude of their contributions.
161 The weights are detailed further in [Section 5.2](#).

162 \mathcal{L}_{KL} is the KL divergence loss between the classification probabilities of the original image and its reconstructed
163 classification probabilities. \mathcal{L}_{GP} and \mathcal{L}_{PLR} are the *gradient penalty* and *path length regularization* losses described
164 in the WGAN-GP[4] and StyleGAN2 paper[7] respectively. \mathcal{L}_{adv} is the Wasserstein adversarial generator loss of x' .
165 Finally, the discriminator’s loss is the Wasserstein adversarial discriminator loss.

¹StyleGAN can take on the order of 40 days on one GPU for high resolutions [6]

166 5 Experimental setup

167 5.1 Datasets

168 The pre-trained models the authors offer are trained on the Flickr-Faces-HQ Dataset [6]². The dataset contains
169 70,000 high-quality PNG images at 1024×1024 resolution with large variations in terms of age, ethnicity, and image
170 background. They use it to find the top attributes which contribute to perceiving a person’s age (young or old) or
171 gender (male or female). They also preprocess the images by lowering the resolution to 256x256. The official dataset is
172 unlabeled. It is not clear whether the authors’ dataset is an internal, labeled Google version or an unofficially labeled
173 dataset.

174 For training, the MNIST [1] dataset is used due to its simplicity. Only the examples with labels 8 or 9 are kept and
175 the resolution is increased to 32x32. MNIST was chosen because images compressed to 16x16 or even 8x8 tend to be
176 recognizable for humans. Unfortunately, LPIPS relies on neural networks that have a fixed number of pooling layers.
177 Without editing reimplementing of LPIPS, the lowest resolution possible is 32.

178 5.2 Hyperparameters

179 A complete list of hyperparameters can be found in Table 2 (see Appendix C). A hyperparameter search was not
180 performed for two reasons. First, the training time is long – even for very low resolutions, this is constraining. Second,
181 the criteria for evaluating success is based on a human user, making automated hyperparameter tuning unintuitive.

182 5.3 Computational requirements

183 Most of our experiments were conducted on Google Colab along with our systems. For training our models we use
184 Colab’s NVIDIA Tesla K80 GPU. Our code is provided in the following GitHub repository: [MLRC_2021_FALL-E358](https://github.com/mlrc2021/fall-e358).

185 The basic architecture of the StyleGAN2 was adapted from NVlabs’ [GitHub repository](#). As previously mentioned, we
186 modify the basic architecture, to align with StyleEx’s generator and load Lang et al.’s pre-trained weights. The training
187 code was adapted from [labml.ai Annotated Paper Implementations](#)’ StyleGAN implementation.

188 Training the model on MNIST for 50,000 iterations takes on the order of nine hours to train on Colab. The time required
189 for AttFind is dependent on the resolution, latent dimension, and the number of images in the dataset. Finding the
190 attribute of a single image took approximately one minute for an image with resolution 32 and a latent space of 514.

191 6 Results

192 6.1 Rebuilding StyleEx results

193 To support Claim 1, we recreate their pre-trained models to PyTorch and test if our results agree. In Figure 3 (see
194 Appendix A), we compare the results from our PyTorch StyleEx to their TensorFlow implementation. There are minor
195 differences in the probabilities from the PyTorch classifier which are likely caused by differences in default values or
196 module implementations in the two frameworks.

197 6.2 AttFind results

198 We are now equipped to test our PyTorch models on the AttFind method and inspect the principal attributes of the age
199 classifier; meaning the attributes with the highest contribution to young or old classification. To this end, we compute
200 the AttFind algorithm – with our classifier and generator as inputs – using the 250 latent variables of the FFHQ dataset.
201 As can be seen in Figures 1 and 5 (see Appendix B), our model obtains the same attributes as in the original paper.

202 In addition, we implement the **Independent** selection strategy, to generate image-specific explanations as described in
203 the original paper. This method is a *local* explanation that returns the top-k attributes affecting a classifier’s decision for
204 a single image rather than the entire dataset. The results are shown in Figure 2.

205 These results support the author’s Claim 2, that AttFind discovers significant attributes for a classifier’s decision.
206 Notably, in 1c the reported probability of the top left image is 17% in the paper, while the probability we find with our
207 and their notebook classifier is 39%.

²<https://github.com/NVlabs/ffhq-dataset>

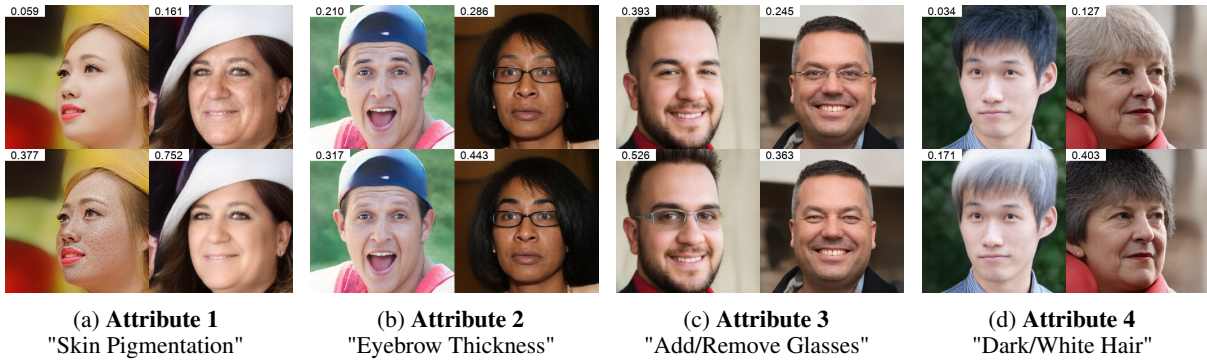


Figure 1: **Top 4 attributes for the perceived age classifier detected by our model.** These images show how the probability of classifying a person as young or old changes based on each attribute. On the first column of each image we display the probability of the person being classified as old and on the second column the probability of them being classified as young.



Figure 2: **Independent selection strategy.** Top-5 detected attributes for explaining a perceived-age classifier for a *specific* image. The attributes obtained are different from those presented in Figure 1 which are computed based on the largest average effect over 250 images. The probabilities displayed correspond to the person being classifier as old.

208 6.3 Quantitative evaluation results

209 To validate the authors' [Claim 3](#) that attributes obtained are identifiable by humans, we conduct the two user studies
 210 explained in the paper. Both studies ([Classification](#) and [Verbal description](#)) aim to prove that the top extracted attributes
 211 are distinct, visually coherent, and can be used as counterfactual explanations.

212 The material used for the classification study was obtained by our PyTorch StyleEx model on the perceived gender
 213 classifier (top 6 attributes), and by the authors' [supplementary material](#) for the perceived age classifier (top 4 attributes).
 214 The verbal description study combines a mixture of attributes from our and the authors' models, explaining Face and
 215 Cats/Dogs classifiers. Results for both studies were provided by 30 users (different per study).

216 Table 1 shows that the results we obtain are within a standard deviation of their results; verifying their contribution that
 217 StyleEx provides attributes that are easily distinguishable by humans.

218 Table 3 depicts the three most common words used, to describe the most prominent attribute that changes in the images
 219 (see Appendix D). By inspecting the results, we draw two main conclusions. First, for all coordinates except skin color
 220 (i.e. 5th row in Face(age/gender) classifiers), the majority of the users use the same word in their descriptions. Second,
 221 the most common word used is different per attribute, proving that each attribute is unique. Our results agree with the
 222 results provided in the original paper.

223 6.4 Reconstruction Generalization

224 To further investigate the proposed model, we create new latent variables using images from the FFHQ dataset on our
 225 architectures with their pre-trained weights. Then, we use the obtained latent variables to reconstruct the images using
 226 our pre-trained generator. Finally, we follow the same process using their architecture and compare the resulting images.
 227 Our StyleEx reconstructs a clearer image, compared to their model which is more blurred. This may occur because of
 228 some differences in the formatting between the frameworks.

	Theirs	Ours
Perceived Gender	0.96(± 0.047)	0.94(± 0.031)
Perceived Age	0.983(± 0.037)	0.978(± 0.025)

Table 1: **Classification study results.** Correct identification of the top-6 attributes.

229 6.5 Training

230 The training proved quite volatile. The \mathcal{L}_{rec} would get stuck in local minima during training. Examples of the images
 231 reconstructed by the fully trained model (see Appendix E).

232 Lang et al. experimented with two training regimens. The first regimen was trained using only $E(x)$ as w , the inputs
 233 to the generator, and the above loss. The second regimen alternated between using $E(x)$ and a randomly generated
 234 encoding, \bar{w} . This \bar{w} is created by applying a mapping network to z , where $z \sim \mathcal{N}(0^n, 1^n)$ and n is the dimensionality
 235 of w . For this randomly generated $\bar{x}' = G(\bar{w})$, only the adversarial loss is calculated. Training using \bar{w} can be viewed
 236 as the same as training a vanilla StyleGAN. Because we are unsure which method was used for the results in their paper
 237 and notebook, we experimented with both. However, the first regimen was the only one that converged.

238 Though we were able to train a model, due to time constraints, we were unable to fully investigate Claim 1.

239 Again due to time constraints, we were unable to run AttFind on the trained model to fully test Claim 3.

240 7 Discussion

241 Using the definition of reproducibility³ by the U.S. National Science Foundation (NSF) subcommittee on replicability
 242 in science, it is difficult to determine Lang et al.’s reproducibility. All details regarding the experimental setup, such
 243 as the hyperparameters, the hours of training, the number of steps, the labels of the datasets, etc. are omitted, thus
 244 recreating the exact materials of the original investigators is difficult. Since our definition is an implication and we
 245 cannot satisfy the first condition, we cannot determine the reproducibility.

246 Instead, we will use a looser definition of reproducibility. We will refer to reproducibility as the ability for another
 247 researcher to test their claims. We found that, given enough time, the StyleEx is seemingly reproducible. However, given
 248 a limited time budget such as our own, the paper is not fully reproducible. We, therefore, can only provide unit tests of
 249 their claims. The following sections will discuss information from the results section 6 and to what degree they confirm
 250 reproducibility claim by claim.

251 7.1 Claim 1

252 The most difficult claim to investigate, given a limited time budget, is the effect of classifier-based training on the
 253 StyleSpace. The original paper trains three models, the StyleEx with and without integration of the classifier in training
 254 and the StyleGAN v2. We found, once the training algorithm is implemented correctly, just training all three models
 255 will take at least 24 hours for 50,000 epochs on one GPU even for the simple MNIST dataset. The authors stated that it
 256 took approximately a week to train StyleEx with 8GPUs. Over two weeks of training time is beyond our time constraints.

257 In addition, we observed that training is volatile.⁴ The reconstruction error stagnates in a local minimum before suddenly
 258 dipping. However, the model was not always able to escape the local minima within 50,000 iterations. This suggests
 259 that, though their results are likely replicable, their replicability may be stochastic. This again hinders reproducibility
 260 when time is limited.

261 7.2 Claim 2

262 The claim that the authors document the most was Claim 2, their AttFind method. Because the method was implemented
 263 in the notebook provided, testing reproducibility was easy.

³“reproducibility refers to the ability of a researcher to duplicate the results of a prior study using the same materials as were used by the original investigator”

⁴An example of successful training can be found [here](#) and one where the model failed to converge [here](#)

264 We were able to verify that for the perceived age classifier, our model obtains the same top attributes. We conclude that
265 their method can discover the most influential classifier-related attributes.

266 In addition to their notebook, we modified the AttFind method to find the principal attributes of a single image as shown
267 in Figure 2. This validated the sub-claim of AttFind that StyleEx can provide image-specific explanations. Rather than
268 finding the *globally* important attributes, the model can find the *locally* important attributes for a particular image.

269 7.3 Claim 3

270 The authors claim that StyleEx is *applicable* to a variety of real-world problems. Applicability can be interpreted in two
271 different ways. One can interpret it as being *possible* to apply StyleEx to a variety of domains, or as *practical* to apply
272 StyleEx to a variety of domains. From what we have seen in Figures 1, 2, it is possible to use StyleEx for explaining an
273 age classifier, thus it can explain a real-world problem. From Figure 6 (see Appendix E), we found that the StyleEx can
274 be trained to, at minimum, reconstruct MNIST data, thus multiple domains.

275 Though we have found that it is *possible*, we have also found that it is seemingly impractical. Every domain requires
276 the model to be retrained, meaning every domain requires days or weeks of training.

277 7.4 What was easy

278 The open-source notebook is very well structured, which combined with the pseudo-code outlined in Algorithm 1 of
279 their paper, made the AttFind method easy to replicate. In addition, the provided pre-trained models helped to derive
280 some of the vague components of StyleEx model.

281 7.5 What was difficult

282 As we already emphasized, there are many difficulties in reproducing this paper. StyleEx is built on top of several
283 previous papers making the knowledge needed for implementation substantial. Lang et al. proposed a model without
284 providing code, that is computationally expensive, and with volatile training behavior. In addition, that is sensitive to
285 hyperparameters, which in our case were unknown. Even when scaling down the complexity of the model using smaller
286 resolutions, the time cost of training exceeds what was feasible with our time constraints.

287 Taking shortcuts to subvert these difficulties had a multitude of challenges. We found loading weights from TensorFlow
288 to PyTorch deceptively complex and far from trivial due to differences between the frameworks. Even evaluating their
289 notebook came with difficulties as the dataset they trained on FFHQ does not officially have labels, so the details of
290 their dataset were unknown.

291 7.6 Future Work

292 The primary goal of this paper was to reproduce the work of Lang et al., however, through reimplementing their
293 code, we found two open avenues for future research. Firstly, the paper focused on general image explanations but
294 did not show examples of misclassified data. It would be interesting to see what insights can be obtained through
295 StyleEx. Secondly, the paper compared StyleEx only with StyleGAN v2 models. AttFind seems applicable to general
296 autoencoders, and not specific to GANs. Viewing StyleEx as an autoencoder, rather than a GAN seems like a promising
297 angle for scalability to a similar counterfactual generator.

298 **References**

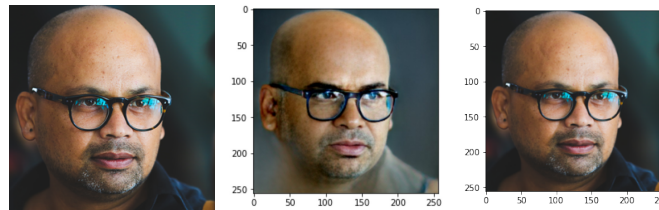
- 299 [1] L. Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing*
300 *Magazine*, 29(6):141–142, 2012.
- 301 [2] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, and S. Lee. Counterfactual visual explanations. In *International*
302 *Conference on Machine Learning*, pages 2376–2384. PMLR, 2019.
- 303 [3] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, and S. Lee. Counterfactual visual explanations. In *International*
304 *Conference on Machine Learning*, pages 2376–2384. PMLR, 2019.
- 305 [4] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans.
306 In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances*
307 *in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- 308 [5] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets:
309 Efficient convolutional neural networks for mobile vision applications, 2017.
- 310 [6] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks, 2019.
- 311 [7] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality
312 of stylegan. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8107–8116,
313 2020.
- 314 [8] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, et al. Interpretability beyond feature attribution:
315 Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages
316 2668–2677. PMLR, 2018.
- 317 [9] O. Lang, Y. Gandselman, M. Yarom, Y. Wald, G. Elidan, A. Hassidim, W. T. Freeman, P. Isola, A. Globerson,
318 M. Irani, and I. Mosseri. Explaining in style: Training a gan to explain a classifier in stylespace. *ArXiv*,
319 abs/2104.13369, 2021.
- 320 [10] T. Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38,
321 2019.
- 322 [11] R. K. Mothilal, A. Sharma, and C. Tan. Explaining machine learning classifiers through diverse counterfactual
323 explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages
324 607–617, 2020.
- 325 [12] K. Schutte, O. Moindrot, P. Hérent, J.-B. Schiratti, and S. Jégou. Using stylegan for visual interpretability of deep
326 learning models on medical images. *arXiv preprint arXiv:2101.07563*, 2021.
- 327 [13] Z. Wu, D. Lischinski, and E. Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation.
328 In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12863–12872,
329 2021.

330 **A Our StyleEx vs Lang et al.'s**



(a) TensorFlow (theirs) (b) PyTorch (ours)

Figure 3: **Comparison of StyleEx models results.** The probabilities shown correspond to being classifier as young.



(a) Original image (b) TensorFlow (theirs) (c) PyTorch (ours)

Figure 4: **Comparison of StyleEx models encoding and then reconstructing an image.** Both models use their encoder and classifier to produce the latent variable. Then using their generator the image is reconstructed from the latent variable.

331 **B AttFind Lang et al.'s top attributes**



(a) Attribute 1 "Skin Pigmentation" (b) Attribute 2 "Eyebrow Thickness" (c) Attribute 3 "Add/Remove Glasses" (d) Attribute 4 "Dark/White Hair"

Figure 5: **Top 4 attributes for the perceived age classifier detected by Lang et al.'s pre-trained model.** These images show how the probability of classifying a person as young or old changes based on each attribute. On the first column of each image, we display the probability of the person being classified as old and on the second column the probability of them being classified as young.

332 **C Hyperparameters**

	Our StylEx	Lang et al’s StylEx
Step Size	1e-3	2e-4
Number of Steps	50,000	250,000
Total Loss Weights ($\mathcal{L}_{rec}, \mathcal{L}_{adv}, \mathcal{L}_c, \mathcal{L}_{PL}$)	1,1,1,1	1,1,1,?
Reconstruction Loss Weights ($\mathcal{L}_w, \mathcal{L}_x, \mathcal{L}_{LIPS}$)	.1, 1, .1	.1, 1, .1
Latent Dimension	32	512
Number of Classes	2	2 (depending on data)
Image Resolution	32	256
Classifier Structure	DenseNet121	MobileNet
Optimizer	Adam	?

Table 2: Training hyperparameters

333 **D Verbal Description Study**





















Cats/Dogs					Face				
		eye: 0.73	pupil: 0.16	shape: 0.1			eyebrow: 0.90	thick: 0.17	brow: 0.07
		mouth: 0.73	open: 0.3	tongue: 0.16			tooth: 0.30	lip: 0.10	disappear: 0.07
		ear: 0.90	right: 0.06	become: 0.06			glass: 0.90	size: 0.13	bigger: 0.10
							mouth: 0.70	open: 0.40	lip: 0.10
							bright: 0.37	skin: 0.30	light: 0.27
							mustache: 0.93	facial: 0.07	hair: 0.07
							eye: 0.77	color: 0.47	eyelash: 0.13

Table 3: **Verbal description study results.** The 3 most common words used in user descriptions for the Cat/Dogs (a) and Face (age/gender) (b) classifiers. This user study proves the distinctness of each attribute since the most common word used to describe each attribute change is different per classifier.

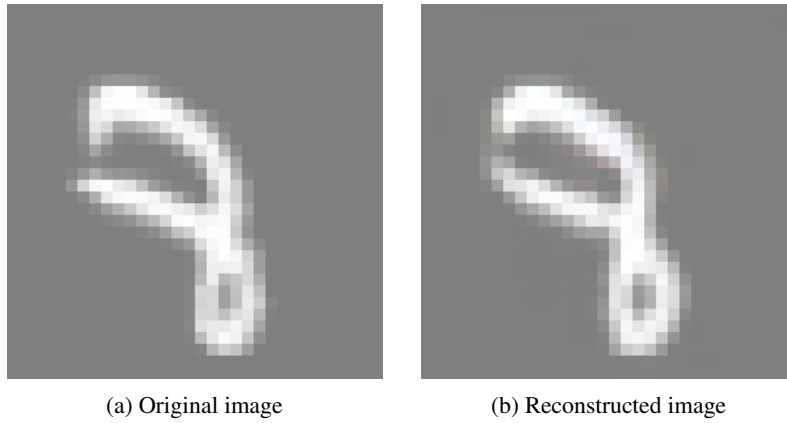


Figure 6: An example of image reconstruction on the MNIST dataset. The StyleEx had converged however, it was trained conditioned on a classifier that always predicted 8, thus was effectively trained without a classifier. It's loss curves can be found [here](#).