Zero-shot Model Diagnosis

Jinqi Luo* Zhaoning Wang* Chen Henry Wu Dong Huang Fernando De la Torre
Carnegie Mellon University
{jinqil, zhaoning, chenwu2, dghuang, ftorre}@cs.cmu.edu

Abstract

When it comes to deploying deep vision models, the behavior of these systems must be explicable to ensure confidence in their reliability and fairness. A common approach to evaluate deep learning models is to build a labeled test set with attributes of interest and assess how well it performs. However, creating a balanced test set (i.e., one that is uniformly sampled over all the important traits) is often time-consuming, expensive, and prone to mistakes. The question we try to address is: can we evaluate the sensitivity of deep learning models to arbitrary visual attributes ***without an annotated test set****?*

This paper argues the case that ***Z****ero-sh****o****t* ***M****odel Diagnosis (ZOOM) is possible without the need for a test set nor labeling. To avoid the need for test sets, our system relies on a generative model and CLIP. The key idea is enabling the user to select a set of prompts (relevant to the problem) and our system will automatically search for semantic counterfactual images (i.e., synthesized images that flip the prediction in the case of a binary classifier) using the generative model. We evaluate several visual tasks (classification, key-point detection, and segmentation) in multiple visual domains to demonstrate the viability of our methodology. Extensive experiments demonstrate that our method is capable of producing counterfactual images and offering sensitivity analysis for model diagnosis without the need for a test set.*

1. Introduction

Deep learning models inherit data biases, which can be accentuated or downplayed depending on the model's architecture and optimization strategy. Deploying a computer vision deep learning model requires extensive testing and evaluation, with a particular focus on features with potentially dire social consequences (e.g., non-uniform behavior across gender or ethnicity). Given the importance of the problem, it is common to collect and label large-scale datasets to evaluate the behavior of these models across attributes of interest. Unfortunately, collecting these test datasets is extremely time-consuming, error-prone, and expensive. Moreover, a balanced dataset, that is uniformly distributed across all attributes of interest, is also typically impractical to acquire due to its combinatorial nature. Even with careful metric analysis in this test set, no robustness nor fairness can be guaranteed since there can be a mismatch between the real and test distributions [25]. This research will explore model diagnosis without relying on a test set in an effort to *democratize* model diagnosis and lower the associated cost.

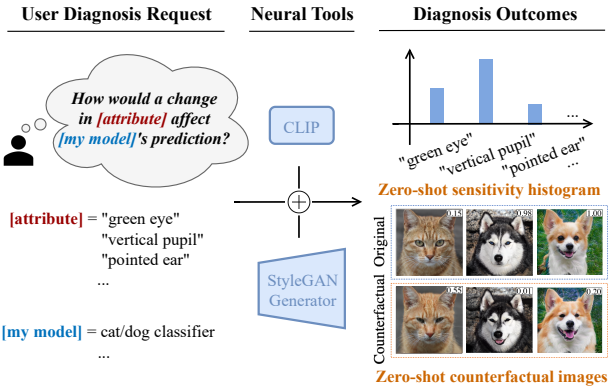


Figure 1. Given a differentiable deep learning model (e.g., a cat/dog classifier) and user-defined text attributes, how can we determine the model's sensitivity to specific attributes without using labeled test data? Our system generates counterfactual images (bottom right) based on the textual directions provided by the user, while also computing the sensitivity histogram (top right).

Counterfactual explainability as a means of model diagnosis is drawing the community's attention [5,20]. Counterfactual images visualize the sensitive factors of an input image that can influence the model's outputs. In other words, counterfactuals answer the question: *"How can we modify the input image* $\mathbf{x}$ *(while fixing the ground truth) so that the model prediction would diverge from* $\mathbf{y}$ *to* $\hat{\mathbf{y}}$*?"*. The parameterization of such counterfactuals will provide insights into identifying key factors of where the model fails. Unlike existing image-space adversary techniques [4,18], counterfactuals provide semantic perturbations that are interpretable by humans. However, existing counterfactual studies re-

*Equal contribution.

quire the user to either collect uniform test sets [10], annotate discovered bias [15], or train a model-specific explanation every time the user wants to diagnose a new model [13].

On the other hand, recent advances in Contrastive Language-Image Pretraining (CLIP) [24] can help to overcome the above challenges. CLIP enables text-driven applications that map user text representations to visual manifolds for downstream tasks such as avatar generation [7], motion generation [37] or neural rendering [22, 30]. In the domain of image synthesis, StyleCLIP [21] reveals that text-conditioned optimization in the StyleGAN [12] latent space can decompose latent directions for image editing, allowing for the mutation of a specific attribute without disturbing others. With such capability, users can freely edit semantic attributes conditioned on text inputs. This paper further explores its use in the scope of model diagnosis.

The central concept of the paper is depicted in Fig. 1. Consider a user interested in evaluating which factors contribute to the lack of robustness in a cat/dog classifier (target model). By selecting a list of keyword attributes, the user is able to (1) see counterfactual images where semantic variations flip the target model predictions (see the classifier score in the top-right corner of the counterfactual images) and (2) quantify the sensitivity of each attribute for the target model (see sensitivity histogram on the top). Instead of using a test set, we propose using a StyleGAN generator as the picture engine for sampling counterfactual images. CLIP transforms user's text input, and enables model diagnosis in an open-vocabulary setting. This is a major advantage since there is no need for collecting and annotating images and minimal user expert knowledge. In addition, we are not tied to a particular annotation from datasets (e.g., specific attributes in CelebA [16]).

To summarize, our proposed work offers three major improvements over earlier efforts:

- The user requires neither a labeled, balanced test dataset, and minimal expert knowledge in order to evaluate where a model fails (i.e., model diagnosis). In addition, the method provides a sensitivity histogram across the attributes of interest.
- When a different target model or a new user-defined attribute space is introduced, it is not necessary to retrain our system, allowing for practical use.
- The target model fine-tuned with counterfactual images not only slightly improves the classification performance, but also greatly increases the distributional robustness against counterfactual images.

2. Related Work

This section reviews prior work on attribute editing with generative models and recent efforts on model diagnosis.

2.1. Attribute Editing with Generative Models

With recent progress in generative models, GANs supports high-quality image synthesis, as well as semantic attributes editing [35]. [1, 6] edit the images by perturbing the intermediate latent space encoded from the original images. These methods rely on images to be encoded to latent vectors to perform attribute editing. On the contrary, StyleGAN [12] can produce images by sampling the latent space. Many works have explored ways to edit attributes in the latent space of StyleGAN, either by relying on image annotations [27] or in an unsupervised manner [8, 28]. StyleSpace [34] further disentangles the latent space of StyleGAN and can perform specific attribute edits by disentangled style vectors. Based upon StyleSpace, StyleCLIP [21] builds the connection between the CLIP language space and StyleGAN latent space to enable arbitrary edits specified by the text. Our work adopts this concept for fine-grained attribute editing.

2.2. Model Diagnosis

To the best of our knowledge, model diagnosis without a test set is a relatively unexplored problem. In the adversarial learning literature, it is common to find methods that show how image-space perturbations [4, 18] flip the model prediction; however, such perturbations lack visual interpretability. [36] pioneers in synthesizing adversaries by GANs. More recently, [9, 23, 26] propose generative methods to synthesize semantically perturbed images to visualize where the target model fails. However, their attribute editing is limited within the dataset's annotated labels. Instead, our framework allows users to easily customize their own attribute space, in which we visualize and quantify the biased factors that affect the model prediction. On the bias detection track, [13] co-trains a model-specific StyleGAN with each target model, and requires human annotators to name attribute coordinates in the Stylespace. [3, 14, 15] synthesize counterfactual images by either optimally traversing the latent space or learning an attribute hyperplane, after which the user will inspect the represented bias. Unlike previous work, we propose to diagnose a deep learning model without any model-specific re-training, new test sets, or manual annotations/inspections.

3. Method

This section firstly describes our method to generate counterfactual images guided by CLIP in a zero-shot manner. We then introduce how we perform the sensitivity analysis across attributes of interest. Fig. 2 shows the overview of our framework.

3.1. Notation and Problem Definition

Let f_θ, parameterized by θ, be the target model that we want to diagnose. In this paper, f_θ denotes two types of

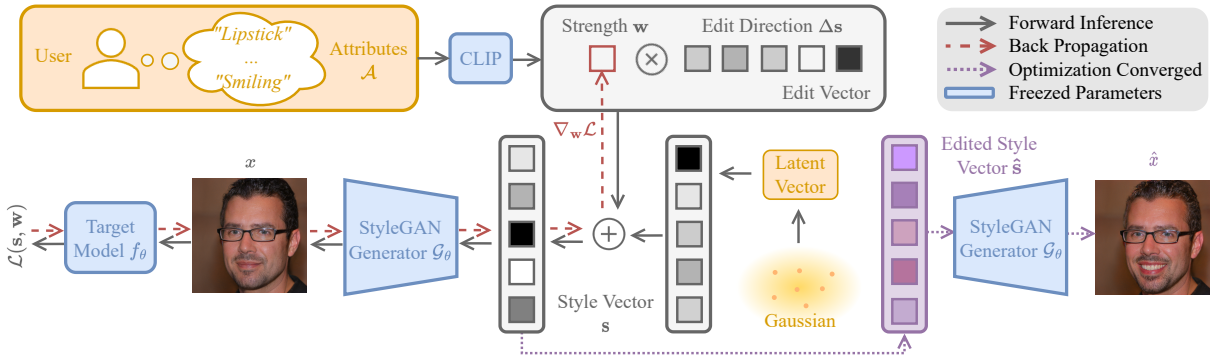


Figure 2. The ZOOM framework. Black solid lines stand for forward passes, red dashed lines stand for backpropagation, and purple dashed lines stand for inference after the optimization converges. The user inputs single or multiple attributes, and we map them into edit directions with the method in Sec. 3.2. Then we assign to each edit direction (attribute) a weight, which represents how much we are adding/removing this attribute. We iteratively perform adversarial learning on the attribute space to maximize the counterfactual effectiveness.

deep nets: binary attribute classifiers and face keypoint detectors. Note that our approach is extendable to any end-to-end differentiable target deep models. Let $\mathcal{G}_\phi$, parameterized by ϕ, be the style generator that synthesizes images by $\mathbf{x} = \mathcal{G}_\phi(\mathbf{s})$ where $\mathbf{s}$ is the style vector in Style Space $\mathcal{S}$ [34]. We denote a counterfactual image as $\hat{\mathbf{x}}$, which is a synthesized image that misleads the target model f_θ, and denote the original reference image as $\mathbf{x}$. a is defined as a single user input text-based attribute, with its domain $\mathcal{A} = \{a_i\}_{i=1}^N$ for N input attributes. $\hat{\mathbf{x}}$ and $\mathbf{x}$ differs only along attribute directions $\mathcal{A}$. Given a set of $\{f_\theta, \mathcal{G}_\phi, \mathcal{A}\}$, our goal is to perform counterfactual-based diagnosis to interpret where the model fails without manually collecting nor labeling any test set. Unlike traditional approaches of image-space noises which lack explainability to users, our method adversarially searches the counterfactual in the user-designed semantic space. To this end, our diagnosis will have three outputs, namely counterfactual images (Sec. 3.3), sensitivity histograms (Sec. 3.4), and distributionally robust models (Sec. 3.5).

3.2. Extracting Edit Directions

This section examines the terminologies, method, and modification we adopt in ZOOM to extract suitable global directions for attribute editing. Since CLIP has shown strong capability in disentangling visual representation [19], we incorporate style channel relevance from StyleCLIP [21] to find edit directions for each attribute.

Given the user's input strings of attributes, we want to find an image manipulation direction $\Delta\mathbf{s}$ for any $\mathbf{s} \sim \mathcal{S}$, such that the generated image $\mathcal{G}_\phi(\mathbf{s}+\Delta\mathbf{s})$ *only* varies in the input attributes. Recall that CLIP maps strings into a text embedding $\mathbf{t} \in \mathcal{T}$, the text embedding space. For a string attribute description a and a neutral prefix p, we obtain the CLIP text embedding difference $\Delta\mathbf{t}$ by:

$$\Delta\mathbf{t} = \text{CLIP}_{\text{text}}(p \oplus a) - \text{CLIP}_{\text{text}}(p) \tag{1}$$

where $\oplus$ is the string concatenation operator. To take 'Eyeglasses' as an example, we can get $\Delta\mathbf{t} = \text{CLIP}_{\text{text}}(\text{'a face with Eyeglasses'}) - \text{CLIP}_{\text{text}}(\text{'a face'})$.

To get the edit direction, $\Delta\mathbf{s}$, we need to utilize a style relevance mapper $\mathbf{M} \in \mathbb{R}^{c_\mathcal{S} \times c_\mathcal{T}}$ to map between the CLIP text embedding vectors of length $c_\mathcal{T}$ and the Style space vector of length $c_\mathcal{S}$. StyleCLIP optimizes $\mathbf{M}$ by iteratively searching meaningful style channels: mutating each channel in $\mathcal{S}$ and encoding the mutated images by CLIP to assess whether there is a significant change in $\mathcal{T}$ space. To prevent undesired edits that are irrelevant to the user prompt, the edit direction $\Delta\mathbf{s}$ will filter out channels that the style value change is insignificant:

$$\Delta\mathbf{s} = (\mathbf{M} \cdot \Delta\mathbf{t}) \odot \mathbb{1}((\mathbf{M} \cdot \Delta\mathbf{t}) > \lambda), \tag{2}$$

where λ is the hyper-parameter for the threshold value. $\mathbb{1}(\cdot)$ is the indicator function, and $\odot$ is the element-wise product operator. Since the success of attribute editing by the extracted edit directions will be the key to our approach, Appendix A will show the capability of CLIP by visualizing the global edit direction on multiple sampled images, conducting the user study, and analyzing the effect of λ.

3.3. Style Counterfactual Synthesis

Identifying semantic counterfactuals necessitates a manageable parametrization of the semantic space for effective exploration. For ease of notation, we denote $(\Delta\mathbf{s})_i$ as the global edit direction for i^{th} attribute $a_i \in \mathcal{A}$ from the user input. After these N attributes are provided and the edit directions are calculated, we initialize the control vectors $\mathbf{w}$ of length N where the i^{th} element w_i controls the strength of the i^{th} edit direction. Our counterfactual edit will be a linear combination of normalized edit directions: $\mathbf{s}_{edit} = \sum_{i=1}^N w_i \frac{(\Delta\mathbf{s})_i}{||(\Delta\mathbf{s})_i||}$.

The black arrows in Fig. 2 show the forward inference to synthesize counterfactual images. Given the parametriza-

tion of attribute editing strengths and the final loss value, our framework searches for counterfactual examples in the optimizable edit weight space. The original sampled image is $\mathbf{x} = G_\phi(\mathbf{s})$, and the counterfactual image is

$$\hat{\mathbf{x}} = G_\phi(\mathbf{s} + \mathbf{s}_{edit}) = G_\phi\left(\mathbf{s} + \sum_{i=1}^{N} w_i \frac{(\Delta\mathbf{s})_i}{||(\Delta\mathbf{s})_i||}\right), \quad (3)$$

which is obtained by minimizing the following loss, $\mathcal{L}$, that is the weighted sum of three terms:

$$\mathcal{L}(\mathbf{s}, \mathbf{w}) = \alpha\mathcal{L}_{target}(\hat{\mathbf{x}}) + \beta\mathcal{L}_{struct}(\hat{\mathbf{x}}) + \gamma\mathcal{L}_{attr}(\hat{\mathbf{x}}). \quad (4)$$

We back-propagate to optimize $\mathcal{L}$ w.r.t the weights of the edit directions $\mathbf{w}$, shown as the red pipeline in Fig. 2.

The targeted adversarial loss $\mathcal{L}_{target}$ for binary attribute classifiers minimizes the distance between the current model prediction $f_\theta(\hat{\mathbf{x}})$ with the flip of original prediction $\hat{p}_{cls} = 1 - f_\theta(\mathbf{x})$. In the case of an eyeglass classifier on a person wearing eyeglasses, $\mathcal{L}_{target}$ will guide the optimization to search $\mathbf{w}$ such that the model predicts no eyeglasses. For a keypoint detector, the adversarial loss will minimize the distance between the model keypoint prediction with a set of *random* points $\hat{p}_{kp} \sim \mathcal{N}$:

$$\text{(binary classifier) } \mathcal{L}_{target}(\hat{\mathbf{x}}) = L_{CE}(f_\theta(\hat{\mathbf{x}}), \hat{p}_{cls}), \quad (5)$$

$$\text{(keypoint detector) } \mathcal{L}_{target}(\hat{\mathbf{x}}) = L_{MSE}(f_\theta(\hat{\mathbf{x}}), \hat{p}_{kp}). \quad (6)$$

If we only optimize $\mathcal{L}_{target}$ w.r.t the global edit directions, it is possible that the method will not preserve image statistics of the original image and can include the particular attribute that we are diagnosing. To constrain the optimization, we added a structural loss $\mathcal{L}_{struct}$ and an attribute consistency loss $\mathcal{L}_{attr}$ to avoid generation collapse. $\mathcal{L}_{struct}$ [32] aims to preserve global image statistics of the original image $\mathbf{x}$ including image contrasts, background, or shape identity during the adversarial editing. While $\mathcal{L}_{attr}$ enforces that the target attribute (perceived ground truth) be consistent on the style edits. For example, when diagnosing the eyeglasses classifier, ZOOM preserves the original status of eyeglasses and precludes direct eyeglasses addition/removal.

$$\mathcal{L}_{struct}(\hat{\mathbf{x}}) = L_{SSIM}(\hat{\mathbf{x}}, \mathbf{x}) \quad (7)$$

$$\mathcal{L}_{attr}(\hat{\mathbf{x}}) = L_{CE}(\text{CLIP}(\hat{\mathbf{x}}), \text{CLIP}(\mathbf{x})) \quad (8)$$

Given a pretrained target model f_θ, a domain-specific style generator G_ϕ, and a text-driven attribute space $\mathcal{A}$, our goal is to sample an original style vector $\mathbf{s}$ for each image and search its counterfactual edit strength $\hat{\mathbf{w}}$:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}}\, \mathcal{L}(\mathbf{s}, \mathbf{w}). \quad (9)$$

Unless otherwise stated, we iteratively update $\mathbf{w}$ as:

$$\mathbf{w} = \text{clamp}_{[-\epsilon,\epsilon]}(\mathbf{w} - \eta\nabla_{\mathbf{w}}\mathcal{L}), \quad (10)$$

where η is the step size and ϵ is the clamp bound to avoid synthesis collapse caused by exaggerated edit. Note that the maximum counterfactual effectiveness does not indicate the maximum edit strength (i.e., $w_i = \epsilon$), since the attribute edit direction does not necessarily overlap with the target classifier direction. The attribute change is bi-directional, as the w_i can be negative in Eq. 3. Details of using other optimization approaches (e.g., linear approximation [18]) will be discussed in Appendix C.

3.4. Attribute Sensitivity Analysis

Single-attribute counterfactual reflects the sensitivity of target model on the individual attribute. By optimizing independently along the edit direction for a single attribute and averaging the model probability changes over images, our model generates independent sensitivity score h_i for each attribute a_i:

$$h_i = \mathbb{E}_{\mathbf{x}\sim\mathcal{P}(\mathbf{x}),\hat{\mathbf{x}}=\text{ZOOM}(\mathbf{x},a_i)}|f_\theta(\mathbf{x}) - f_\theta(\hat{\mathbf{x}})|. \quad (11)$$

The sensitivity score h_i is the probability difference between the original image $\mathbf{x}$ and generated image $\hat{\mathbf{x}}$, at the most counterfactual point when changing attribute a_i. We synthesize a number of images from $\mathcal{G}_\phi$, then iteratively compute the sensitivity for each given attribute, and finally normalize all sensitivities to draw the histogram as shown in Fig. 4. The histogram indicates the sensitivity of the evaluated model f_θ on each of the user-defined attributes. Higher sensitivity of one attribute means that the model is more easily affected by that attribute.

3.5. Counterfactual Training

The multi-attribute counterfactual approach visualizes semantic combinations that cause the model to falter, providing valuable insights for enhancing the model's robustness. We naturally adopt the concept of iterative adversarial training [18] to robustify the target model. For each iteration, ZOOM receives the target model parameter and returns a batch of mutated counterfactual images with the model's original predictions as labels. Then the target model will be trained on the counterfactually-augmented images to achieve the robust goal:

$$\theta^* = \underset{\theta}{\text{argmin}}\, \mathbb{E}_{\mathbf{x}\sim\mathcal{P}(\mathbf{x}),\hat{\mathbf{x}}=\text{ZOOM}(\mathbf{x},\mathcal{A})} L_{CE}(f_\theta(\hat{\mathbf{x}}), f_\theta(\mathbf{x})) \quad (12)$$

where batches of $\mathbf{x}$ are randomly sampled from the StyleGAN generator $\mathcal{G}_\phi$. In the following, we abbreviate the process as Counterfactual Training (CT). Note that, although not explicitly expressed in Eq. 12, the CT process is a min-max game. ZOOM synthesizes counterfactuals to maximize the variation of model prediction (while persevering the perceived ground truth), and the target model is learned with the counterfactual images to minimize the variation.

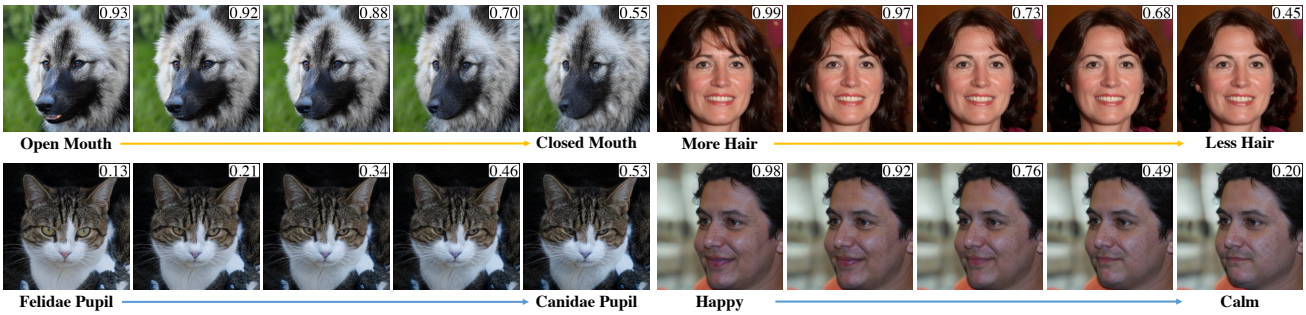
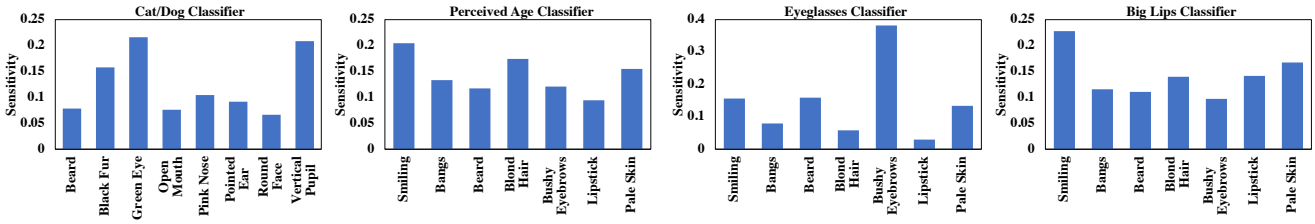
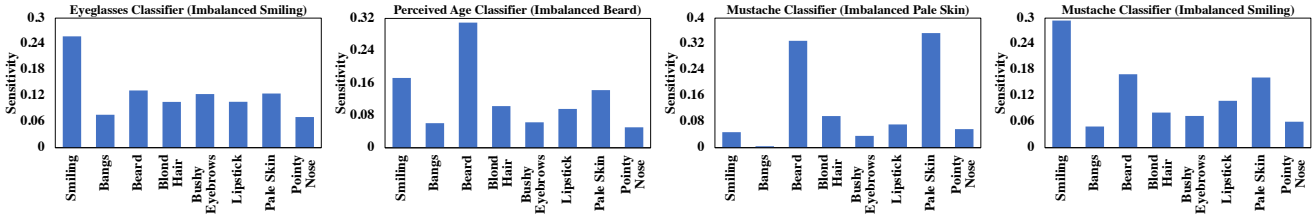


Figure 3. Effect of progressively generating counterfactual images on (left) cat/dog classifier (0-Cat / 1-Dog), and (right) perceived age classifier (0-Senior / 1-Young). Model probability prediction during the process is attached at the top right corner.



(a) Model diagnosis histograms generated by ZOOM on four facial attribute classifiers.



(b) Model diagnosis histograms generated by ZOOM on four classifiers trained on manually-crafted imbalance data.

Figure 4. Model diagnosis histograms generated by ZOOM. The vertical axis values reflect the attribute sensitivities calculated by averaging the model probability change over all sampled images. The horizontal axis is the attribute space input by user.

4. Experimental Results

This section describes the experimental validations on the effectiveness and reliability of ZOOM. First, we describe the model setup in Sec. 4.1. Sec. 4.2 and Sec. 4.3 visualize and validate the model diagnosis results for the single-attribute setting. In Sec. 4.4, we show results on synthesized multiple-attribute counterfactual images and apply them to counterfactual training.

4.1. Model Setup

Pre-trained models: We used Stylegan2-ADA [11] pre-trained on FFHQ [12] and AFHQ [1] as our base generative networks, and the pre-trained CLIP model [24] which is parameterized by ViT-B/32. We followed StyleCLIP [21] setups to compute the channel relevance matrices $\mathcal{M}$.

Target models: Our classifier models are ResNet50 with single fully-connected head initialized by TorchVision[1]. In training the binary classifiers, we use the Adam optimizer with learning rate 0.001 and batch size 128. We train binary classifiers for *Eyeglasses, Perceived Gender, Mustache, Perceived Age* attributes on CelebA and for *cat/dog* classification on AFHQ. For the 98-keypoint detectors, we used the HR-Net architecture [31] on WFLW [33].

[1]https://pytorch.org/blog/how-to-train-state-of-the-art-models-using-torchvision-latest-primitives/

4.2. Visual Model Diagnosis: Single-Attribute

Understanding where deep learning model fails is an essential step towards building trustworthy models. Our proposed work allows us to generate counterfactual images (Sec. 3.3) and provide insights on sensitivities of the target model (Sec. 3.4). This section visualizes the counterfactual images in which only one attribute is modified at a time.

Fig. 3 shows the single-attribute counterfactual images. Interestingly (but not unexpectedly), we can see that reducing the hair length or joyfulness causes the age classifier more likely to label the face to an older person. Note that our approach is extendable to multiple domains, as we change the cat-like pupil to dog-like, the dog-cat classification tends towards the dog. Using the counterfactual images, we can conduct model diagnosis with the method mentioned in Sec. 3.4, on which attributes the model is sen-

sitive to. In the histogram generated in model diagnosis, a higher bar means the model is more sensitive toward the corresponding attribute.

Fig. 4a shows the model diagnosis histograms on regularly-trained classifiers. For instance, the cat/dog classifier histogram shows outstanding sensitivity to green eyes and vertical pupil. The analysis is intuitive since these are cat-biased traits rarely observed in dog photos. Moreover, the histogram of eyeglasses classifier shows that the mutation on bushy eyebrows is more influential for flipping the model prediction. It potentially reveals the positional correlation between eyeglasses and bushy eyebrows. The advantage of single-attribute model diagnosis is that the score of each attribute in the histogram are independent from other attributes, enabling unambiguous understanding of exact semantics that make the model fail. Diagnosis results for additional target models can be found in Appendix B.

4.3. Validation of Visual Model Diagnosis

Evaluating whether our zero-shot sensitivity histograms (Fig. 4) explain the true vulnerability is a difficult task, since we do not have access to a sufficiently large and balanced test set fully annotated in an open-vocabulary setting. To verify the performance, we create synthetically imbalanced cases where the model bias is known. We then compare our results with a supervised diagnosis setting [17]. In addition, we will validate the decoupling of the attributes by CLIP.

4.3.1 Creating imbalanced classifiers

In order to evaluate whether our sensitivity histogram is correct, we train classifiers that are highly imbalanced towards a known attribute and see whether ZOOM can detect the sensitivity w.r.t the attribute. For instance, when training the perceived-age classifier (binarized as Young in CelebA), we created a dataset on which the trained classifier is strongly sensitive to Bangs (hair over forehead). The custom dataset is a CelebA training subset that consists of $20,200$ images. More specifically, there are $10,000$ images that have both young people that have bangs, represented as $(1,1)$, and $10,000$ images of people that are not young and have no bangs, represented as $(0,0)$. The remaining combinations of $(1,0)$ and $(0,1)$ have only 100 images. With this imbalanced dataset, bangs is the attribute that dominantly correlates with whether the person is young, and hence the perceived-age classifier would be highly sensitive towards bangs. See Fig. 5 (the right histograms) for an illustration of the sensitivity histogram computed by our method for the case of an age classifier with bangs (top) and lipstick (bottom) being imbalanced.

We trained multiple imbalanced classifiers with this methodology, and visualize the model diagnosis histograms of these imbalanced classifiers in Fig. 4b. We can observe that the ZOOM histograms successfully detect the

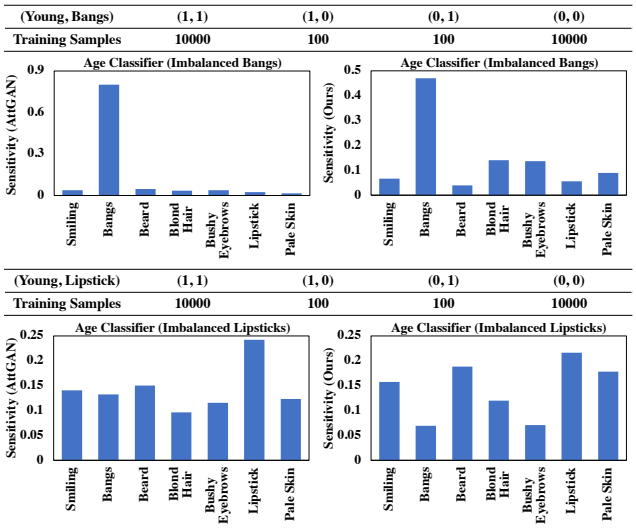
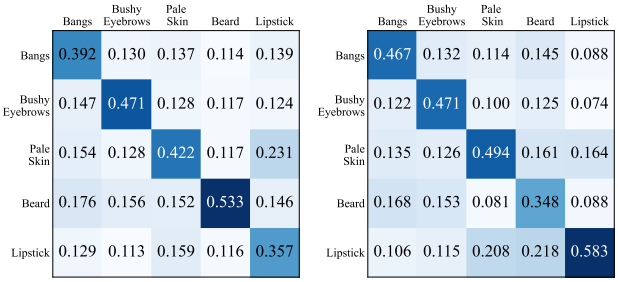


Figure 5. The sensitivity of the age classifier is evaluated with ZOOM (right) and AttGAN (left), achieving comparable results.



(a) Mustache classifier (b) Perceived age classifier

Figure 6. Confusion matrix of CLIP score variation (vertical axis) when perturbing attributes (horizontal axis). This shows that attributes in ZOOM are highly decoupled.

synthetically-made bias, which are shown as the highest bars in histograms. See the caption for more information.

4.3.2 Comparison with supervised diagnosis

We also validated our histogram by comparing it with the case in which we have access to a generative model that has been explicitly trained to disentangle attributes. We follow the work on [17] and used AttGAN [6] trained on the CelebA training set over 15 attributes[2]. After the training converged, we performed adversarial learning in the attribute space of AttGAN and create a sensitivity histogram using the same approach as Sec. 3.4. Fig. 5 shows the result of this method on the perceived-age classifier which is made biased towards bangs. As anticipated, the AttGAN histogram (left) corroborates the histogram derived from our method (right). Interestingly, unlike ZOOM, AttGAN show less sensitivity to remaining attributes. This is likely

[2] *Bald, Bangs, Black_Hair, Blond_Hair, Brown_Hair, Bushy_Eyebrows, Eyeglasses, Male, Mouth_Slightly_Open, Mustache, No_Beard, Pale_Skin, Young, Smiling, Wearing_Lipstick.*

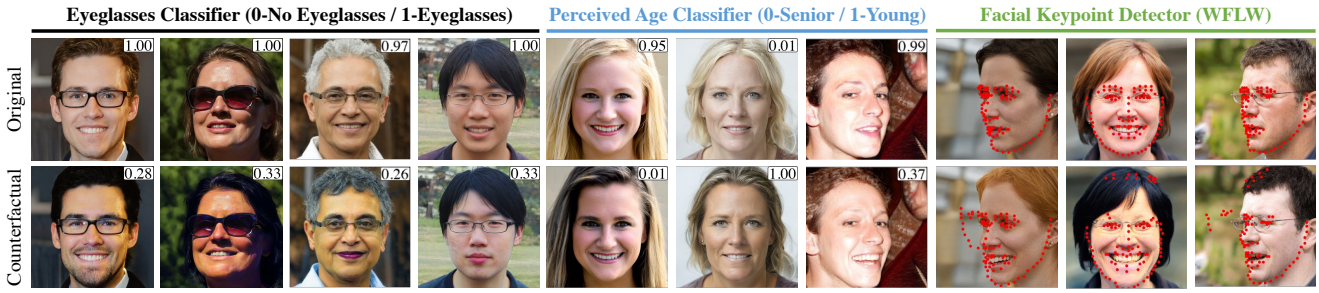


Figure 7. Multi-attribute counterfactual in faces. The model probability is documented in the upper right corner of each image.

because AttGAN has a latent space learned in a supervised manner and hence attributes are better disentangled than with StyleGAN. Note that AttGAN is trained with a fixed set of attributes; if a new attribute of interest is introduced, the dataset needs to be re-labeled and AttGAN retrained. ZOOM, however, merely calls for the addition of a new text prompt. More results in Appendix B.

4.3.3 Measuring disentanglement of attributes

Previous works demonstrated that the StyleGAN's latent space can be entangled [2, 27], adding undesired dependencies when searching single-attribute counterfactuals. This section verifies that our framework can disentangle the attributes and mostly edit the desirable attributes.

We use CLIP as a super annotator to measure attribute changes during single-attribute modifications. For $1,000$ images, we record the attribute change after performing adversarial learning in each attribute, and average the attribute score change. Fig. 6 shows the confusion matrix during single-attribute counterfactual synthesis. The horizontal axis is the attribute being edited during the optimization, and the vertical axis represents the CLIP prediction changed by the process. For instance, the first column of Fig. 6a is generated when we optimize over bangs for the mustache classifier. We record the CLIP prediction variation. It clearly shows that bangs is the dominant attribute changing during the optimization. From the main diagonal of matrices, it is evident that the ZOOM mostly perturbs the attribute of interest. The results indicate reasonable disentanglement among attributes.

4.4. Visual Model Diagnosis: Multi-Attributes

In the previous sections, we have visualized and validated single-attribute model diagnosis histograms and counterfactual images. In this section, we will assess ZOOM's ability to produce counterfactual images by concurrently exploring multiple attributes within $\mathcal{A}$, the domain of user-defined attributes. The approach conducts multi-attribute counterfactual searches across various edit directions, identifying distinct semantic combinations that result in the target model's failure. By doing so, we can effectively create more powerful counterfactuals images (see Fig. 9).

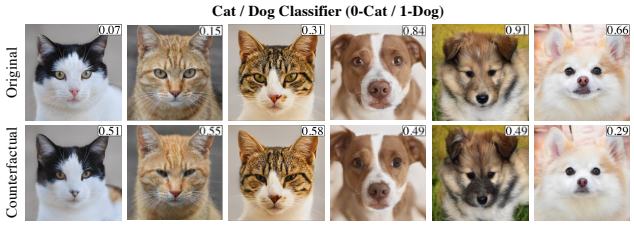


Figure 8. Multi-attribute counterfactual on Cat/Dog classifier. The number in each image is the predicted probability of being a dog.

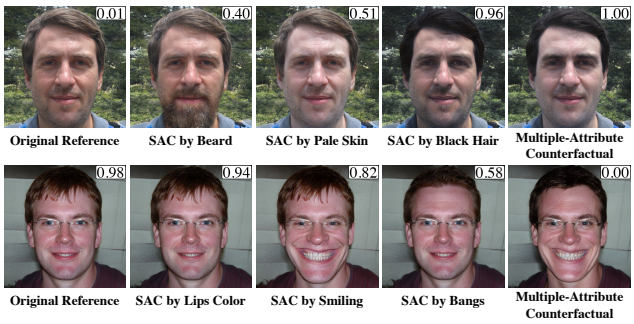
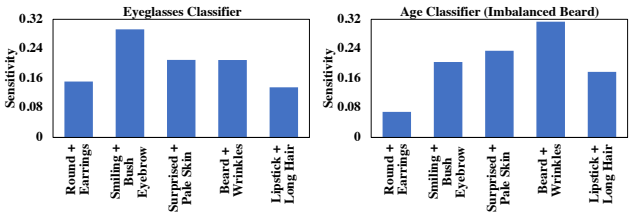


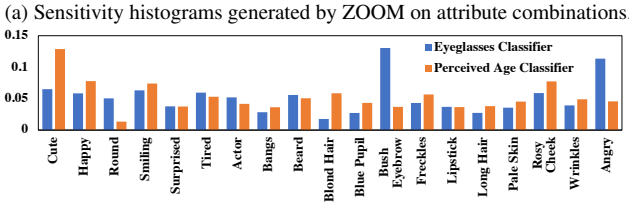
Figure 9. Multiple-Attribute Counterfactual (MAC, Sec. 4.4) compared with Single-Attribute Counterfactual (SAC, Sec. 4.2). We can see that optimization along multiple directions enable the generation of more powerful counterfactuals.

Fig. 7 and Fig. 8 show examples of multi-attribute counterfactual images generated by ZOOM, against human and animal face classifiers. It can be observed that multiple face attributes such as lipsticks or hair color are edited in Fig. 7, and various cat/dog attributes like nose pinkness, eye shape, and fur patterns are edited in Fig. 8. These attribute edits are blended to affect the target model prediction. Appendix B further illustrates ZOOM counterfactual images for semantic segmentation, multi-class classification, and a church classifier. By mutating semantic representations, ZOOM reveals semantic combinations as outliers where the target model underfits.

In the following sections, we will use the Flip Rate (the percentage of counterfactuals that flipped the model prediction) and Flip Resistance (the percentage of counterfactuals for which the model successfully withheld its prediction) to evaluate the multi-attribute setting.



(a) Sensitivity histograms generated by ZOOM on attribute combinations.



(b) Model diagnosis by ZOOM over 19 attributes. Our framework is generalizable to analyze facial attributes of various domains.

Figure 10. Customizing attribute space for ZOOM.

4.4.1 Customizing attribute space

In some circumstances, users may finish one round of model diagnosis and proceed to another round by adding new attributes, or trying a new attribute space. The linear nature of attribute editing (Eq. 3) in ZOOM makes it possible to easily add or remove attributes. Table 1 shows the flip rates results when adding new attributes into $\mathcal{A}$ for perceived age classifier and big lips classifier. We can observe that a different attribute space will results in different effectiveness of counterfactual images. Also, increasing the search iteration will make counterfactual more effective (see last row). Note that neither re-training the StyleGAN nor user-collection/labeling of data is required at any point in this procedure. Moreover, Fig. 10a shows the model diagnosis histograms generated with combinations of two attributes. Fig. 10b demonstrates the capability of ZOOM in a rich vocabulary setting where we can analyze attributes that are not labeled in existing datasets [16, 29].

4.4.2 Counterfactual training results

This section evaluates regular classifiers trained on CelebA [16] and counterfactually-trained (CT) classifiers on a mix of CelebA data and counterfactual images as described in Sec. 3.5. Table 2 presents accuracy and flip resistance (FR) results. CT outperforms the regular classifier. FR is assessed over 10,000 counterfactual images, with FR-25 and FR-100 denoting Flip Resistance after 25 and 100 optimization iterations, respectively. Both use $\eta = 0.2$ and $\epsilon = 30$. We can observe that the classifiers after CT are way less likely to be flipped by counterfactual images while maintaining a decent accuracy on the CalebA testset. Our approach robustifies the model by increasing the tolerance toward counterfactuals. Note that CT slightly improves the CelebA classifier when trained on a mixture of CelebA images (original images) and the counterfactual images generated with a generative model trained in the FFHQ [12] images (different domain).

Method	AC Flip Rate (%)	BC Flip Rate (%)
Initialize ZOOM by $\mathcal{A}$	61.95	83.47
+ Attribute: Beard	72.08	90.07
+ Attribute: Smiling	87.47	**96.27**
+ Attribute: Lipstick	90.96	94.07
+ Iterations increased to 200	**92.91**	94.87

Table 1. Model flip rate study. The initial attribute space $\mathcal{A} = \{$Bangs, Blond Hair, Bushy Eyebrows, Pale Skin, Pointy Nose$\}$. AC is the perceived age classifier and BC is the big lips classifier.

Attribute	Metric	Regular (%)	CT (Ours) (%)
Perceived Age	CelebA Accuracy	86.10	**86.29**
	ZOOM FR-25	19.54	**97.36**
	ZOOM FR-100	9.04	**95.65**
Big Lips	CelebA Accuracy	74.36	**75.39**
	ZOOM FR-25	14.12	**99.19**
	ZOOM FR-100	5.93	**88.91**

Table 2. Results of network inference on CelebA original images and ZOOM-generated counterfactual. The CT classifier is significantly less prone to be flipped by counterfactual images, while test accuracy on CelebA remains performant.

5. Conclusion and Discussion

In this paper, we present ZOOM, a zero-shot model diagnosis framework that generates sensitivity histograms based on user's input of natural language attributes. ZOOM assesses failures and generates corresponding sensitivity histograms for each attribute. A significant advantage of our technique is its ability to analyze the failures of a target deep model without the need for laborious collection and annotation of test sets. ZOOM effectively visualizes the correlation between attributes and model outputs, elucidating model behaviors and intrinsic biases.

Our work has three primary limitations. First, users should possess domain knowledge as their input (text of attributes of interest) should be relevant to the target domain. Recall that it is a small price to pay for model evaluation without an annotated test set. Second, StyleGAN2-ADA struggles with generating out-of-domain samples. Nevertheless, our adversarial learning framework can be adapted to other generative models (e.g., stable diffusion), and the generator can be improved by training on more images. We have rigorously tested our generator with various user inputs, confirming its effectiveness for regular diagnosis requests. Currently, we are exploring stable diffusion models to generate a broader range of classes while maintaining the core concept. Finally, we rely on a pre-trained model like CLIP which we presume to be free of bias and capable of encompassing all relevant attributes.

Acknowledgements: We would like to thank George Cazenavette, Tianyuan Zhang, Yinong Wang, Hanzhe Hu, Bharath Raj for suggestions in the presentation and experiments. We sincerely thank Ken Ziyu Liu, Jiashun Wang, Bowen Li, and Ce Zheng for revisions to improve this work.

References

[1] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse Image Synthesis for Multiple Domains. In *CVPR*, 2020.

[2] Edo Collins, Raja Bala, Bob Price, and Sabine Süsstrunk. Editing in Style: Uncovering the Local Semantics of GANs. In *CVPR*, 2020.

[3] Emily Denton and Ben Hutchinson and Margaret Mitchell and Timnit Gebru and Andrew Zaldivar. Image counterfactual sensitivity analysis for detecting unintended bias. *arXiv preprint arXiv:1906.06439*, 2019.

[4] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. 2014.

[5] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual Visual Explanations. In *ICML*, 2019.

[6] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen. AttGAN: Facial Attribute Editing by Only Changing What You Want. In *IEEE TIP*, 2019.

[7] Fangzhou Hong, Mingyuan Zhang, Liang Pan, Zhongang Cai, Lei Yang, and Ziwei Liu. AvatarCLIP: Zero-Shot Text-Driven Generation and Animation of 3D Avatars. In *ACM TOG*, 2022.

[8] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. GANSpace: Discovering Interpretable GAN Controls. In *NeurIPS*, 2020.

[9] Ameya Joshi, Amitangshu Mukherjee, Soumik Sarkar, and Chinmay Hegde. Semantic Adversarial Attacks: Parametric Transformations That Fool Deep Classifiers. In *ICCV*, 2019.

[10] Kimmo Karkkainen and Jungseock Joo. FairFace: Face Attribute Dataset for Balanced Race, Gender, and Age for Bias Measurement and Mitigation. In *WACV*, 2021.

[11] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training Generative Adversarial Networks with Limited Data. In *NeurIPS*, 2020.

[12] Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. In *CVPR*, 2019.

[13] Oran Lang, Yossi Gandelsman, Michal Yarom, Yoav Wald, Gal Elidan, Avinatan Hassidim, William T. Freeman, Phillip Isola, Amir Globerson, Michal Irani, and Inbar Mosseri. Explaining in Style: Training a GAN To Explain a Classifier in StyleSpace. In *ICCV*, 2021.

[14] Bo Li, Qiulin Wang, Jiquan Pei, Yu Yang, and Xiangyang Ji. Which Style Makes Me Attractive? Interpretable Control Discovery and Counterfactual Explanation on StyleGAN. *arXiv preprint arXiv:2201.09689*, 2022.

[15] Zhiheng Li and Chenliang Xu. Discover the Unknown Biased Attribute of an Image Classifier. In *ICCV*, 2021.

[16] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. In *ICCV*, 2015.

[17] Jinqi Luo, Zhaoning Wang, Chen Henry Wu, Dong Huang, and Fernando De la Torre. Semantic image attack for visual model diagnosis. *arXiv preprint arXiv:2303.13010*, 2023.

[18] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *ICLR*, 2018.

[19] Joanna Materzynska, Antonio Torralba, and David Bau. Disentangling Visual and Written Concepts in CLIP. In *CVPR*, 2022.

[20] Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations. In *ACM FAccT*, 2020.

[21] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery. In *ICCV*, 2021.

[22] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D Diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[23] Haonan Qiu, Chaowei Xiao, Lei Yang, Xinchen Yan, Honglak Lee, and Bo Li. SemanticAdv: Generating Adversarial Examples via Attribute-conditioned Image Editing. In *ECCV*, 2020.

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, 2021.

[25] Vikram V. Ramaswamy, Sunnie S. Y. Kim, and Olga Russakovsky. Fair Attribute Classification Through Latent Space De-Biasing. In *CVPR*, 2021.

[26] Axel Sauer and Andreas Geiger. Counterfactual Generative Networks. In *ICLR*, 2021.

[27] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. InterFaceGAN: Interpreting the Disentangled Face Representation Learned by GANs. In *IEEE TPAMI*, 2020.

[28] Yujun Shen and Bolei Zhou. Closed-Form Factorization of Latent Semantics in GANs. In *CVPR*, 2021.

[29] Philipp Terhörst, Daniel Fährmann, Jan Niklas Kolf, Naser Damer, Florian Kirchbuchner, and Arjan Kuijper. MAAD-Face: A Massively Annotated Attribute Dataset for Face Images. In *IEEE TIFS*, 2021.

[30] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. CLIP-NeRF: Text-and-Image Driven Manipulation of Neural Radiance Fields. In *CVPR*, 2022.

[31] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep High-Resolution Representation Learning for Visual Recognition. In *IEEE TPAMI*, 2019.

[32] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image Quality Assessment: from Error Visibility to Structural Similarity. In *IEEE TIP*, 2004.

[33] Wayne Wu, Chen Qian, Shuo Yang, Quan Wang, Yici Cai, and Qiang Zhou. Look at Boundary: A Boundary-Aware Face Alignment Algorithm. In *CVPR*, 2018.

[34] Zongze Wu, Dani Lischinski, and Eli Shechtman. StyleSpace Analysis: Disentangled Controls for StyleGAN Image Generation. In *CVPR*, 2021.

[35] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. GAN Inversion: A Survey. In *IEEE TPAMI*, 2022.

[36] Chaowei Xiao, Bo Li, Jun-yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating Adversarial Examples with Adversarial Networks. In *IJCAI*, 2018.

[37] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. MotionDiffuse: Text-Driven Human Motion Generation with Diffusion Model. *arXiv preprint arXiv:2208.15001*, 2022.