Covering up bias with Markov blankets: A post-hoc cure for attribute prior avoidance

Vinay Prabhu [1] **Dian Ang Yap** [2] **Alexander Wang** [2]

Abstract

Attribute prior avoidance entails subconscious or willful non-modeling of (meta)attributes that datasets are oft born with, such as the semantic facial attributes associated with the CelebA and CelebA-HQ dataset. The consequences of this infirmity are especially stark in generative models, especially normalizing flows, that just model the pixel-space measurements resulting in an attribute-bias-laden latent space. This viscerally manifests itself when we indulge in face manipulation experiments based on latent vector interpolations. In this paper, we address this and propose a post-hoc solution that utilizes an Ising attribute prior learned in the latent attribute space and showcase its efficacy via qualitative experiments.

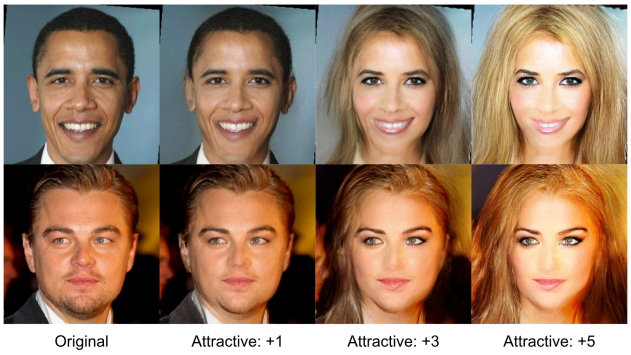

Figure 1. Interpolating along the 'Attractive' latent attribute in different scales converges to pictures of white blonde women.

1. Motivation: Attractive Barack Obama $\neq$ White Blonde Barack Obama

Gif animations showcasing nifty manipulation of celebrity faces[1] by vector operations in the discovered latent space have emerged as a modality of choice in order to advertise the effectiveness of image generative models of all hues, be it GANs(Goodfellow et al., 2014), VAEs(Kingma & Welling, 2013) or Flow-based models(Dinh et al., 2014). In fact, the idea that deep generative models, especially flow-based generative models, can be trained without attribute labels by attempting to directly model the distribution of the input images, $\boldsymbol{x} \sim p(\boldsymbol{x})$, and yet yield latent-space representations worthy of *downstream tasks* like manipulating the semantic attributes of images is seen as a flagship feature of the model (Kingma & Dhariwal, 2018). This is however tantamount to not just ignoring the rich co-occurrence structure between the image attributes in the dataset that these models are trained on but also making a faux image-attribute marginal independence assumption. That is, for a dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{a}_i)\}_{i=1}^{N}$ where $\mathbf{x}_i$ is the i^{th} image and $\mathbf{a}_i$ is the attribute vector associated with that i^{th} image, modeling just $p(\mathbf{x})$ is akin to assuming $p(\mathbf{x}, \mathbf{a}) = p(\mathbf{x})p(\mathbf{a})$ given that marginalization yields: $\sum_{\mathbf{a}} p(\mathbf{x}, \mathbf{a}) = \sum_{\mathbf{a}} p(\mathbf{x})p(\mathbf{a}) = p(\mathbf{x})$.

Specifically, with regards to CelebA-HQ (Liu et al., 2015), we have 30,000 images, each with 40 binary semantic attributes. If we ignore the co-occurrence of these attributes and not account for it, we will end up with inter-attribute spill-over effects, an example of which is showcased in Fig 1. Using Glow models as an example, we see the end result of trying to *enhance* the `attractive` attribute in Barack Obama's and Leonardo DiCaprio's [2] pictures[3].

As we increase the *weight* of influence of the attractive-attribute manipulation latent vector, we end up rendering Obama and DiCaprio as almost Caucasian-blonde women. One can attribute this to the possible idiosyncratic co-prevalence of attributes such as `Blond_Hair`, `Attractive` and `Pale_Skin` in the images of the original dataset.

[1]UnifyID, Redwood City, California [2]Department of Computer Science, Stanford University, Stanford, California. Correspondence to: Dian Ang Yap <dayap@stanford.edu>, Alexander Wang <aswang96@stanford.edu>.

Presented at the ICML 2019 Workshop on Invertible Neural Nets and Normalizing Flows.

[1] https://youtu.be/G06dEcZ-QTg

[2] https://www.imdb.com/name/nm0000138/

[3] https://openai.com/blog/glow/

In this paper, we address this issue by using graph structured Ising priors to model the inter-attribute statistical dependence and later harnessing this model to propose a corrective procedure that allays some of the concerns laid thus far.

The rest of the paper is organized as follows. In Section-2, we cover the estimation procedure for learning the Inter-Attribute Ising Prior (IAIP). In section 3, we propose our post hoc bias- corrective algorithm that generalizes beyond reversible generative models. In Section 4, we showcase some qualitative results that highlights the efficacy of our approach and conclude the paper. This is a work in progress and we stay committed to not just open-sourcing the implementation but are also in the cusp of unveiling an interactive portal provides interfaces to both the IAIP as wel as the corrective procedure.

2. Learning the Inter-Attribute Ising Prior

Probabilistic graphical models are the unifying framework of choice in areas such as bioinformatics, statistical physics and communication theory, for modeling complex dependencies among random variables using ideas from both graph theory and probability theory. This framework not just facilitates building large-scale multivariate statistical models but also allows for elegant visualization and knowledge discovery through the learned statistical dependency graph (Wainwright et al., 2008). In this paper, we choose a specific type of binary Markov Random Fields termed as *Ising Models* (See (Stanley, 1971)) to model the inter-attribute statistical dependencies. The model specifies a distribution $p(\mathbf{a})$ defined over a *substrate* graph, $G(V, E)$. Specifically the conditional distribution of an attribute $a_i \in \{0, 1\}$ given the rest of the attributes $a_{\setminus j}$ and the graph $G(V, E)$ takes the form:

$$p_\Omega\left(a_j|a_{\setminus j}\right) = \frac{\exp\left[\tau_j a_j + a_j \sum\limits_{k \in V_{\setminus j}} \left(\beta_{jk} a_k\right)\right]}{1 + \exp\left[\tau_j + \sum\limits_{k \in V_{\setminus j}} \left(\beta_{jk} a_k\right)\right]} \quad (1)$$

Here, τ_j and β_{jk} are the node parameters and Ising edge parameters respectively.

In order to learn the underlying graph and the associated node and edge weights, we use a network estimation procedure termed `eLasso`, that combines ℓ_1-regularized logistic regression with model selection criterion based on the Extended Bayesian Information Criterion (EBIC)[4].

[4] `eLASSO` is implemented in the R package *IsingFit*: http://cran.r-project.org/web/packages/IsingFit/IsingFit.pdf

The EBIC cost function is define as:

$$BIC_\gamma\left(\hat{\Omega}_j\right) = -2\ell\left(\hat{\Omega}_j\right) + |J| \log\left(n\right) + 2\gamma\,|J| \log\left(p-1\right) \quad (2)$$

Here, n represents is the number of observations (which is 30,000 in the case of Celeb-A-HQ dataset), $p-1$ represents the number of covariates (predictors), and γ is the *prior strength* hyperparameter (Van Borkulo et al., 2014).

Finally, $\ell\left(\hat{\Omega}_j\right)$ is the likelihood component, defined as:

$$\ell\left(\hat{\Omega}_j\right) = \sum_{i=1}^{n} \left(\tau_j a_{ij} + a_j \sum_{k \in V_{\setminus j}} \left(\beta_{jk} a_{ij} a_{ik}\right) - \log\left(1 + \exp\left\{\tau_j + \sum_{k \in V \setminus j} \beta_{jk} a_{ik}\right\}\right)\right) \quad (3)$$

After learning the node weights and the edge-weights using the procedure above, we symmetrize the weights as:

$$w_{jk} = \begin{cases} \frac{(\beta_{jk} + \beta_{kj})}{2}, & \text{if } \beta_{jk} \neq 0 \textit{ and } \beta_{kj} \neq 0 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

2.1. The learned IAIP for the CelebA-HQ dataset

As we learn in the following section, we'd ideally want the learned graph $G(V, E)$ to be connected (no disconnected nodes) as well as sparse. Therefore, we set the the sparsity penalty parameter (γ) of the `eLasso` algorithm to be the largest value above which the resultant graph was disconnected. This variation of the sparsity level of the graph obtained with regards to the number of disconnected nodes as well as the number of edges (a sparsity measure) is seen in Fig 17. As seen, that critical value of the sparisty paramter is $\gamma = 6.4$. In Fig 14, we see that the weighted graph representing the Ising model learned for the 40 binary attributes of the CelebA-HQ dataset (with $\gamma = 6.4$).

This graph will not just pave the way for the bias corrective algorithm to be proposed in the upcoming section, but will also serve as a succinct representation of uncovering anthropocentric bias ingrained during the labeling process itself by the human annotators, by means of Graph exploration using popular GUI Graph visualization software such as Gephi and NodeXL.

With regards to this `eLasso` fitting procedure, we have the following figures. In Fig 16, we see the attribute-wise

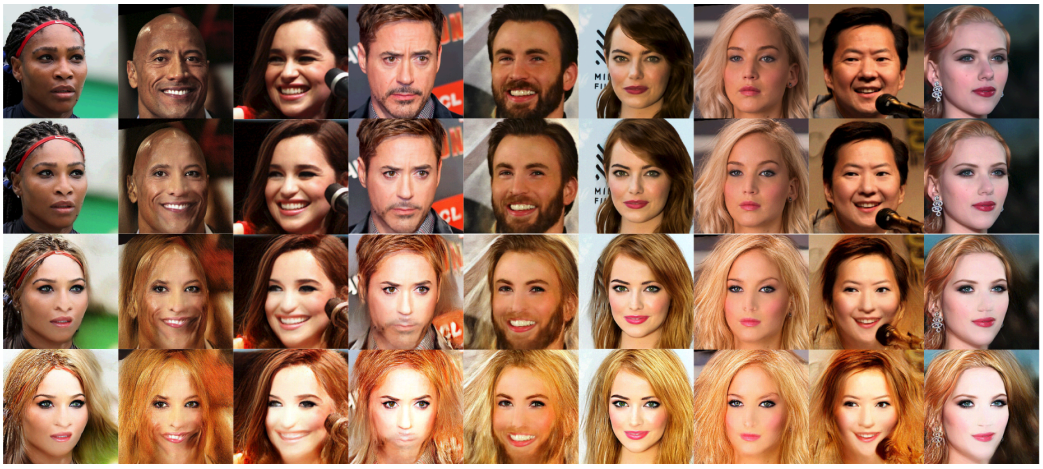

Figure 2. More visualizations of how images of different gender and races converge to young, white-skinned women with long blonde or red hair as we only naively interpolate along the 'attractive' attribute. *From top to bottom: original, attractive: + 1, attractive: +3, attractive: +5.*

thresholds (or *node parameters* $\{\tau_i \text{ i} \in V\}$) .This threshold captures the propensity of a specific attribute to be present in an image in the dataset. As seen, most of the thresholds (with the exception of `Male`, `No-Beard` and `Young`) are all negative . This implies most of the attributes in the Celeb-A-HQ dataset in-fact exude a normative disposition to be absent. The attributes `Rosy-cheeks`, `Blurry` and `Wearing-necktie` had the strongest negative thresholds.This implies that these symptoms had the strongest probability of being present in a celeb-A-HQ image compared to the other attributes. In Fig 15, we see that variation of the tuning parameter per attribute that was used to ensure the best-fit set of neighbors (See (Van Borkulo et al., 2014)). As seen, the features `Mouth-slightly-open` and `Pointy-nose` had the largest values of the tuning parameters associated with them.

3. Methods and Experiments

Formally, we have input data $\boldsymbol{x}$ with associated prior attributes $\boldsymbol{\alpha}$. We have model that maps input $\boldsymbol{x}$ to a latent space representation $\boldsymbol{z}$ parameterized by θ, $f_\theta: \boldsymbol{x} \to \boldsymbol{z}$, and another model $g_\phi: \boldsymbol{z} \to \boldsymbol{x}$ denote a function parameterized by ϕ that maps $\boldsymbol{z}$ to $\boldsymbol{x}$. These models could be any latent variable models, including but not limited to, variational autoencoders (VAE) or flow models such as Glow. We have some attributes $\boldsymbol{\alpha} \subseteq \boldsymbol{a}$ that we would want to interpolate, and $\boldsymbol{\delta}$ denote the degree of interpolation.

We first perform Ising Fit on $\boldsymbol{a}$ to obtain a graph. After fitting an Ising model on the graph prior attributes, we have a neighborhood of nodes $\mathcal{N}$ that are directly connected to i (termed the *Markov blanket*), where $i \in \boldsymbol{\alpha}$ is the attribute we want to interpolate. By considering the weighting of each node $j \in \mathcal{N}$, we propose a new update rule if we would like to interpolate i by some factor δ:

$$\text{Interpolate}(j) = \begin{cases} \delta, & \text{if } j = i \\ -k \cdot \delta \cdot w_{ij}, & \text{if } j \in \mathcal{N}, i \neq j \end{cases} \tag{5}$$

where $k \in \mathbb{R}^+$ is some constant to be found experimentally (in our experiments, we use $k = 0.1$). This step can be iteratively done if we have multiple attributes to interpolate.

Algorithm 1 Corrective Bias Latent Manipulation Algorithm

Input: data $\boldsymbol{x}$, priors $\boldsymbol{a}$, attributes to interpolate $\boldsymbol{\alpha}$, degree of interpolaton $\boldsymbol{\delta}$, hyperparameter $\gamma, k \in \mathbb{R}^+$
$G = (V, E) \leftarrow$ IsingFit $(\boldsymbol{a}, \gamma)$
$f_\theta \leftarrow$ model mapping $\boldsymbol{x}$ to $\boldsymbol{z}$
$g_\phi \leftarrow$ model mapping $\boldsymbol{z}$ to $\boldsymbol{x}$
Initialize $\boldsymbol{z'}$ as $f_\theta(\boldsymbol{x})$
for $\alpha_i = \alpha_1, \alpha_2, \ldots, \alpha_m$ **in** $\boldsymbol{\alpha}$ **do**
  $\boldsymbol{z'} \leftarrow \boldsymbol{z'} + \delta_i \alpha_i$
  $\mathcal{N} \leftarrow$ neighborhood/adjacent vertices of α_i in G
  for $n_j = n_1, n_2, \ldots, n_n$ **in** $\mathcal{N}$ **do**
    $w_{ij} \leftarrow$ weight of edge $\alpha_i n_j$
    $\boldsymbol{z'} \leftarrow -k \cdot \delta_i \cdot w_{ij}$
  end for
end for
$\hat{\boldsymbol{x}} \leftarrow g_\phi(\boldsymbol{z'})$, the interpolated image with corrective bias

4. Results and Discussion

In this section, we focus on showcasing the efficacy of our procedure by qualitative methods. We first focus on Fig 3 that presents the scatter-plots between the Ising weights w_{ij} and d_{ij} as the cosine distance between z_i and z_j, the latent manipulation vectors associated with attributes a_i and a_j:

$$\begin{aligned} d_{ij} &= \text{cosine\_distance}(z_i, z_j) \\ &= 1 - \frac{z_i \cdot z_j}{\|z_i\|_2 \, \|z_j\|_2} \end{aligned} \tag{6}$$

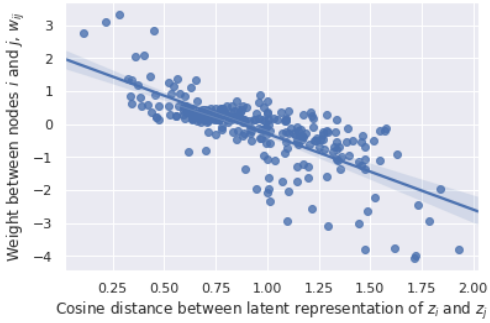

Figure 3. Plot of Ising weights against latent manipulation vectors between any two nodes in Ising graph.

The striking negative correlation confirms our hunch that the co-occurrence of attributes has influenced the latent space in a fairly obvious way. That is, two attributes that always co-occurred in the image dataset (meaning high edge-weight w_{ij} in the Ising prior) ended up with latent space representations with a low cosine distance between them. This is more of a bug than a feature of the model as coincidental occurrences should not dictate the semantic distancing in the latent space. For example, we do not see any philosophical justification for high co-occurrence of `Attractive` and `Pale_Skin` to result in latent space representations of these attributes being close to each other in a *semantically rich* latent space.

Some attributes are are mutually associated with others through stereotypes, which experimentally justifies the need for bias correction. As shown in Fig 5, increasing `High_Cheekbones` also subtly increases the attributes of `Smiling`, `Female` and `Wearing_Lipstick` which are usually associated with feminine characteristics. On the other hand, increasing attributes such as `Bald` also shows increases likelihood of `Goatee`, `Chubby` and `Male`. Applying such attributes in images in contrasting domains (increasing `Lipstick` for male, `Bald` for females) cause the transformed images to lose intrinsic properties of the original image by generalizing towards the implicitly learnt bias.

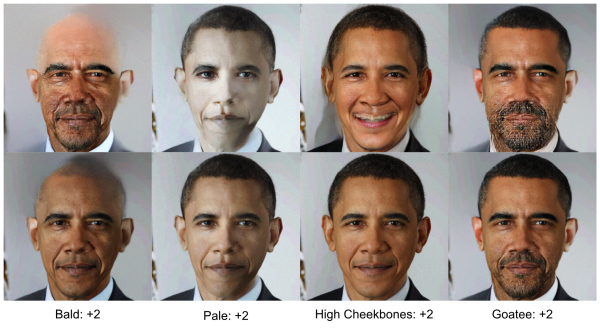

Figure 4. Top: Naively manipulating attribute by +2. Bottom: manipulating attribute by +2 while applying corrective bias penalty to corresponding Markovian blankets.

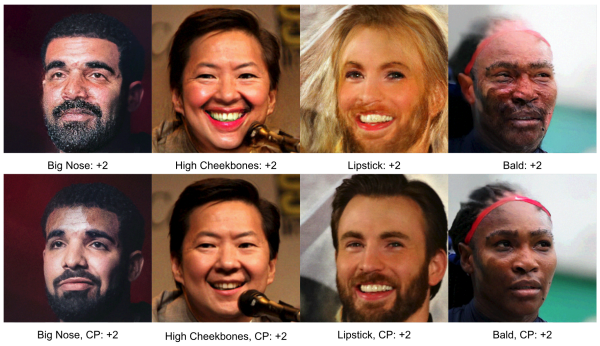

Figure 5. Naive manipulation across one attribute (top) vs manipulation with corrective penalty on neighborhood attributes.

This is also confirmed in Fig 4, where increasing baldness changes the skin color to be that of a white forehead.

We also demonstrate how biases can be corrected as shown in Fig 6. Using `Wearing_Lipstick` as an example, which is implicitly associated with feminine characteristics, the image learns to transform with biases such as `Blonde_Hair`, `Female` and `Wearing_Makeup`. However, by accounting for biases, we show how `Wearing_Lipstick` can be applied across different characteristic domains such as gender and race, while preserving the inherent attributes in the image transformation process.

5. Conclusion

We demonstrate that models that directly learn to model the input data without considering attribute priors while training reveal significant bias. Using Glow as an example of flow-based generative models, we reveal the existing bias by manipulating across the latent space, which mirrors common racial and gender stereotypes.

We also emphasize that this bias also exists in other generative models such as GANs and VAEs as shown in the appendix, and we propose a post-hoc corrective measure for

Figure 6. Demonstration on how inherent bias in the 'wearing lipstick' attribute commonly correlated with blonde hair, young and women can be corrected using our proposed corrective bias algorithm. *Top: wearing lipstick: +2. Bottom: wearing lipstick: +2 with neighborhood corrective penalty.*

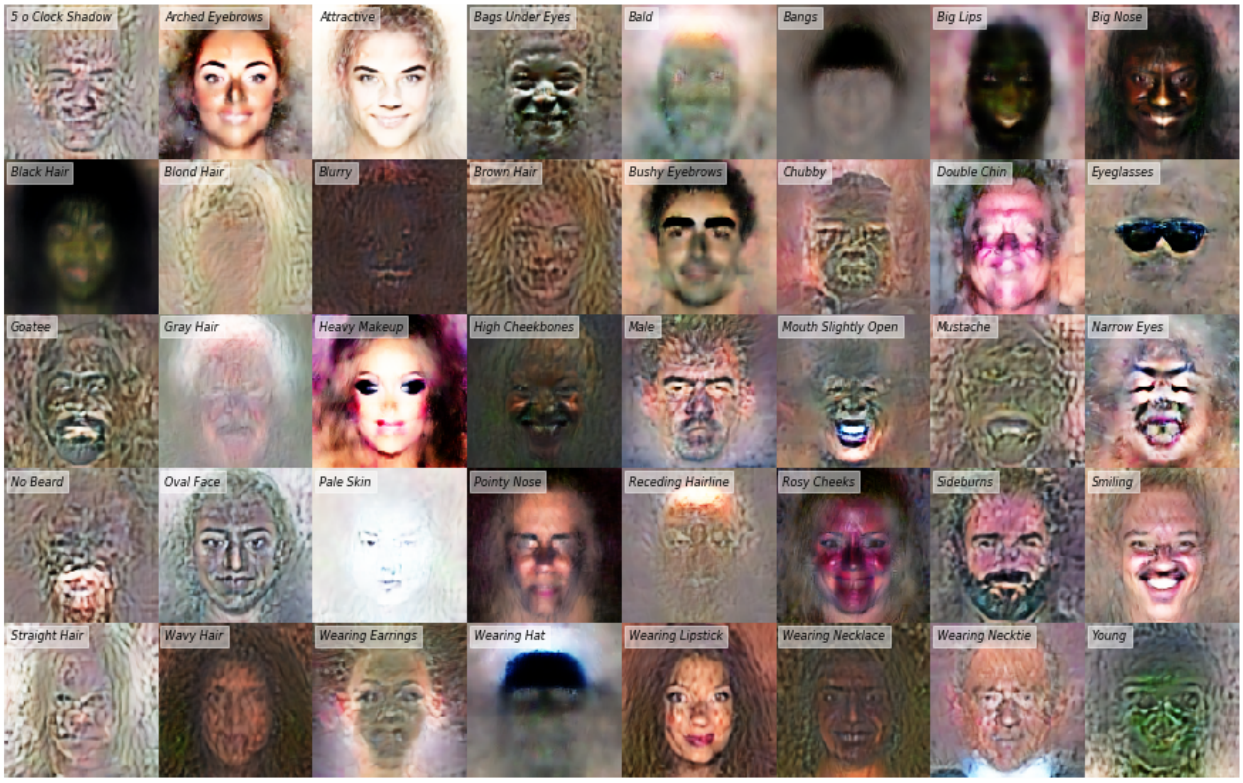

Figure 7. Bias of CelebA illustrated: Visualization of biases in latent space of Generative Adversarial Interpolative Autoencoding (GAIA) (Sainburg et al., 2018). Visualization of latent '`attractive`', '`big nose`', and '`big lips`' correlate to racial profiles that reveal the inherent bias in CelebA

models trained with existing bias that using Ising models fitted on attribute priors.

References

Dinh, L., Krueger, D., and Bengio, Y. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural*

information processing systems, pp. 2672–2680, 2014.

He, Z., Zuo, W., Kan, M., Shan, S., and Chen, X. Arbitrary facial attribute editing: Only change what you want. *CoRR*, abs/1711.10678, 2017. URL http://arxiv.org/abs/1711.10678.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pp. 10215–10224, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Liu, M., Ding, Y., Xia, M., Liu, X., Ding, E., Zuo, W., and Wen, S. Stgan: A unified selective transfer network for arbitrary image attribute editing. 2019.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

Sainburg, T., Thielk, M., Theilman, B., Migliori, B., and Gentner, T. Generative adversarial interpolative autoencoding: adversarial training on latent space interpolations encourage convex latent distributions. *arXiv preprint arXiv:1807.06650*, 2018.

Stanley, H. E. *Phase transitions and critical phenomena*. Clarendon Press, Oxford, 1971.

Van Borkulo, C. D., Borsboom, D., Epskamp, S., Blanken, T. F., Boschloo, L., Schoevers, R. A., and Waldorp, L. J. A new method for constructing networks from binary data. *Scientific reports*, 4:5918, 2014.

Wainwright, M. J., Jordan, M. I., et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

Xiao, T., Hong, J., and Ma, J. Dna-gan: Learning disentangled representations from multi-attribute images. *International Conference on Learning Representations, Workshop*, 2018.

Zhou, S., Xiao, T., Yang, Y., Feng, D., He, Q., and He, W. Genegan: Learning object transfiguration and attribute subspace from unpaired data. *arXiv preprint arXiv:1705.04932*, 2017.

6. Appendix

6.1. Generalizing beyond Normalizing Flows: Revealing Bias in Generative Adversarial Networks Trained on CelebA/CelebA-HQ

We can further confirm that the CelebA dataset is the root cause of the bias shown experimentally by revealing similar biases in other models, such as Generative Adversarial Networks. Below, we experiment with two GAN architectures trained on the CelebA dataset.

Note that the pretrained models only allow the manipulation of 13 attributes: `Bald`, `Bangs`, `Black_Hair`, `Blond_Hair`, `Brown_Hair`, `Bushy_Eyebrows`, `Eyeglasses`, `Male`, `Mouth_Slightly_Open`, `Mustache`, `No_Beard`, `Pale_Skin`, `Young`.

6.1.1. AttGAN: Facial Attribute Editing by Only Changing What You Want

AttGAN, by He et al., was created specifically to modify attributes realistically and in isolation, so that users can "change what [they] want" (He et al., 2017). Users can choose to modify any number of attributes simultaneously, as well as adjust the magnitude of adjustment per attribute.

The authors of AttGAN understand that attributes in CelebA are highly correlated, claiming that in previous models, "adding blond hair always makes a male become a female because most blond hair objects are female in the training set". He et al. correct the effects of this correlation by placing constraints, on the images generated from latent representations, enforcing these constraints by employing an attribute classifier, a main contribution of their paper.

We begin by observing that adjusting the `Male` attribute negatively causes the AttGAN model to generate long hair and add makeup. Next, we adjust the `Blond_Hair` attribute, and observe that this causes the GAN to generate blue eyes and long hair.

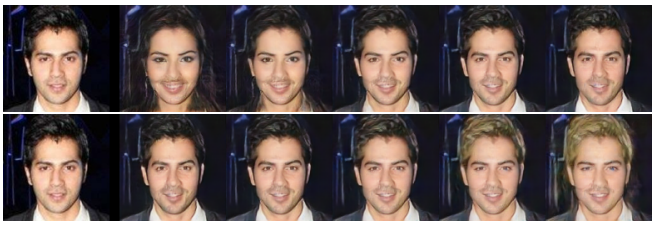

Figure 8. Top: Example of interpolating `Male` attribute between -2 and +2. Bottom: Example of interpolating `Blond_Hair` attribute between -2 and +2. Leftmost image is original image.

We also increase the `Blond_Hair` attribute and the `Male` attributes together. One might expect increasing the `Male` attribute to counteract the long-hair that the `Blond_Hair`

attribute creates. However, as we can see below, it on occasion acts constructively. It is clear that, even with the addition of an attribute classifier to constrain the model, the latent representation manipulations for separate attributes are still not independent. We note, though, that this behavior is seen mostly when the original images are of males. He et al. themselves conclude that model errors occur mostly when large modifications are made to images, such as adding hair to a bald figure.

Lastly, we investigate the manipulation of facial hair. It seems that mustaches and beards, are always generated black, regardless of the original subject's hair color. In fact, if we attempt to generate a mustache on top of a subject who already has facial hair, it simply changes the color of the existing facial hair to black. Additionally, if we generate a mustache on top of a young or less masculine subject, the rest of the face is morphed into that of a more elderly and rugged man.

Figure 9. Top: Examples of manipulating both `Male` and `Blond_Hair` attributes by +1. Middle: Manipulating `Mustache` attribute by +1 on a subject with existing facial hair. Bottom: Manipulating `Mustache` attribute by +1 on a young subject with less masculine features.

6.1.2. STGAN: A Unified Selective Transfer Network for Arbitrary Image Attribute Editing

We also carry out experiments on STGAN, a model built by Liu et al (Liu et al., 2019) extending He et al.'s work. Their main contribution is the selectivity of information passed into their model's generators, choosing only to pass in the difference in attribute vector between the source and target image, so that the generator supposedly only affects the targeted change in attribute.

As with AttGAN, we begin by manipulating the `Male` attribute. We can see that STGAN is less inclined to generate longer hair when decreasing this attribute, but it generates black splotches in the shape of hair, simply without the texture and details of hair. It is obvious that STGAN still suffers from the safe bias in the priors as other models.

Like AttGAN, STGAN changes the eye color of its subject towards blue when increasing the `Blond_Hair` attribute. However, it does not generate long hair like AttGAN does.

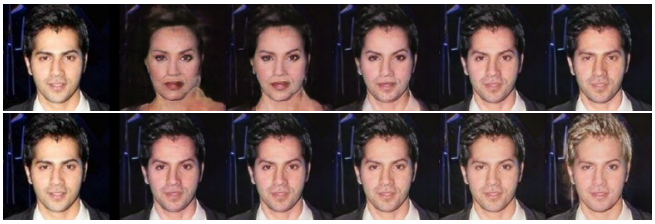

Figure 10. Top: Example of interpolating `Male` attribute between -2 and +2. Bottom: Example of interpolating `Blond_Hair` attribute between -2 and +2. Leftmost image is original image.

Increasing the `Blond_Hair` and the `Male` attributes together reproduces the constructive interaction seen in AttGAN, where even more blond long hair is generated than increasing just `Blond_Hair` on its own. This underlying complex interaction in the latent space seems to be a consistent problem across generative models, as far as we can tell.

Like AttGAN, generated facial hair is always black, regardless of the original subject's hair color, generating a mustache on top of a subject who already has facial hair simply darkens the color of the existing facial hair, and generating a mustache on a young/soft-featured subject makes them appear older and more masculine.

Figure 11. Top: Examples of manipulating both `Male` and `Blond_Hair` attributes by +1. Middle: Manipulating `Mustache` attribute by +1 on a subject with existing facial hair. Bottom: Manipulating `Mustache` attribute by +1 on a young subject with less masculine features.

6.1.3. DNA-GAN: Learning Disentangled Representations from Multi-Attribute Images

Figure 12. Interpolating on '`wearing eyeglasses`' changes the racial profile of the subject matter for DNA-GAN.

DNA-GAN is a supervised method for disentangling multiple factors of variation simultaneously by using multi-attribute images (Xiao et al., 2018). Trained on CelebA, it can manipulate several attributes in the latent representations of images, which is a generalization of GeneGAN (Zhou et al., 2017). DNA-GAN replaces the explicit nulling loss with the annihilating operation and employes a single discriminator for guiding images generation on multiple attributes.

While it seeks to disentangle multiple factors and attributes of images, we see the inherent bias reflected in the CelebA dataset as per Fig 12. While we manipulate across the '`wearing eyeglasses`', the racial profile of the subject matter being manipulated also changes accordingly.

6.1.4. Generative Adversarial Interpolative Autoencoding (GAIA)

The Generative Adversarial Interpolative Autoencoder (GAIA) is novel hybrid between the Generative Adversarial Network (GAN) and the Autoencoder (AE) (Sainburg et al., 2018). It addresses the issue of GANs which are non-bidirectional, while also addressing issues of autoencoders which produces blurry images, and addressing the non-conves latent spaces of autoencoders.

GAIA promotes a convex latent distribution by training adversarially on latent space interpolations. Trained on CelebA, GAIA produces non-blurry samples that match both high- and low-level features of the original images. Upon visualizing the inherent latent spaces learnt, we see that attributes such as '`big nose`' and '`big lips`' correlate to racial profiles that reveal the inherent bias in CelebA, as per Fig 7.

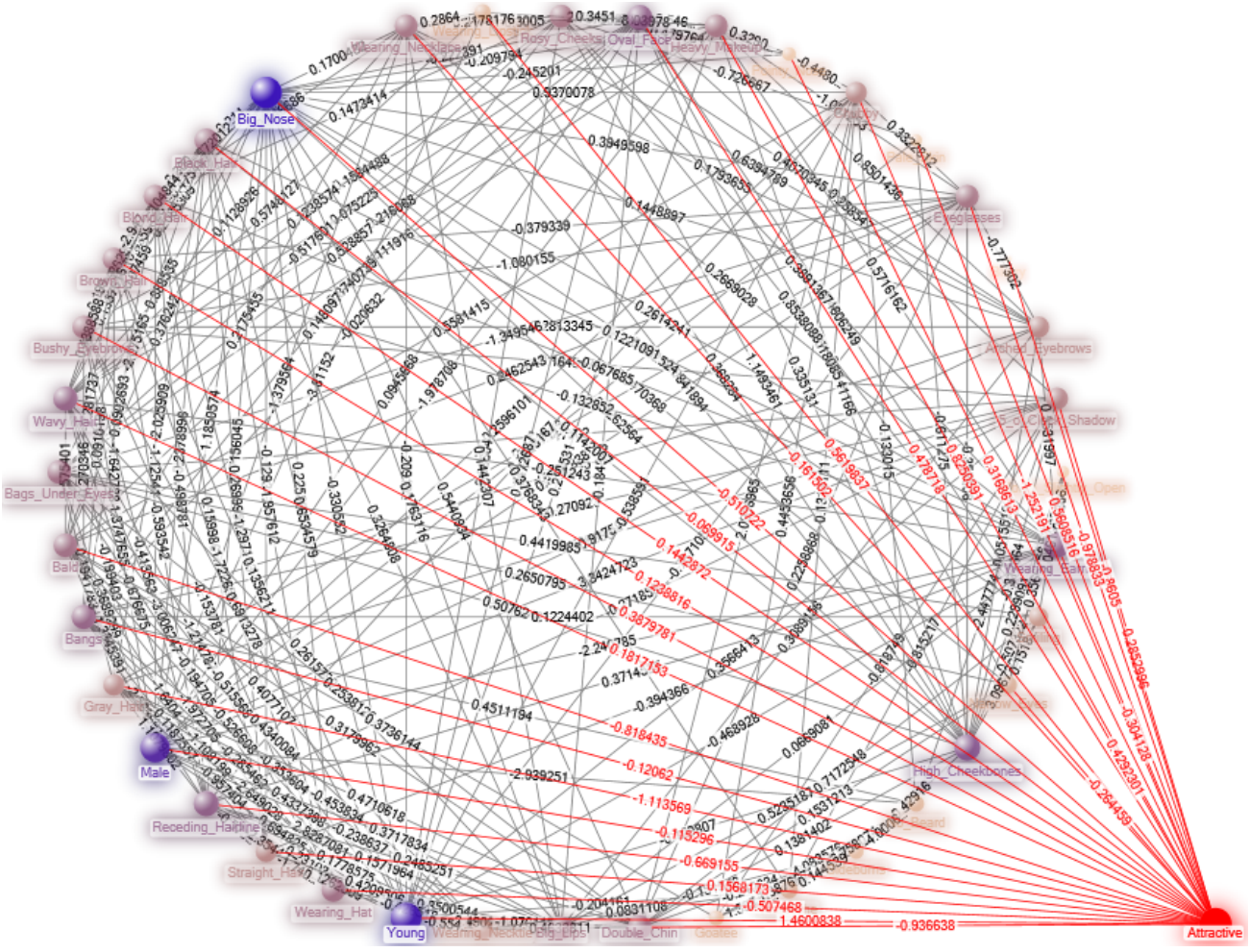

Figure 13. The weighted graph representing the Ising model for the binary attributes of the CelebA-HQ dataset.

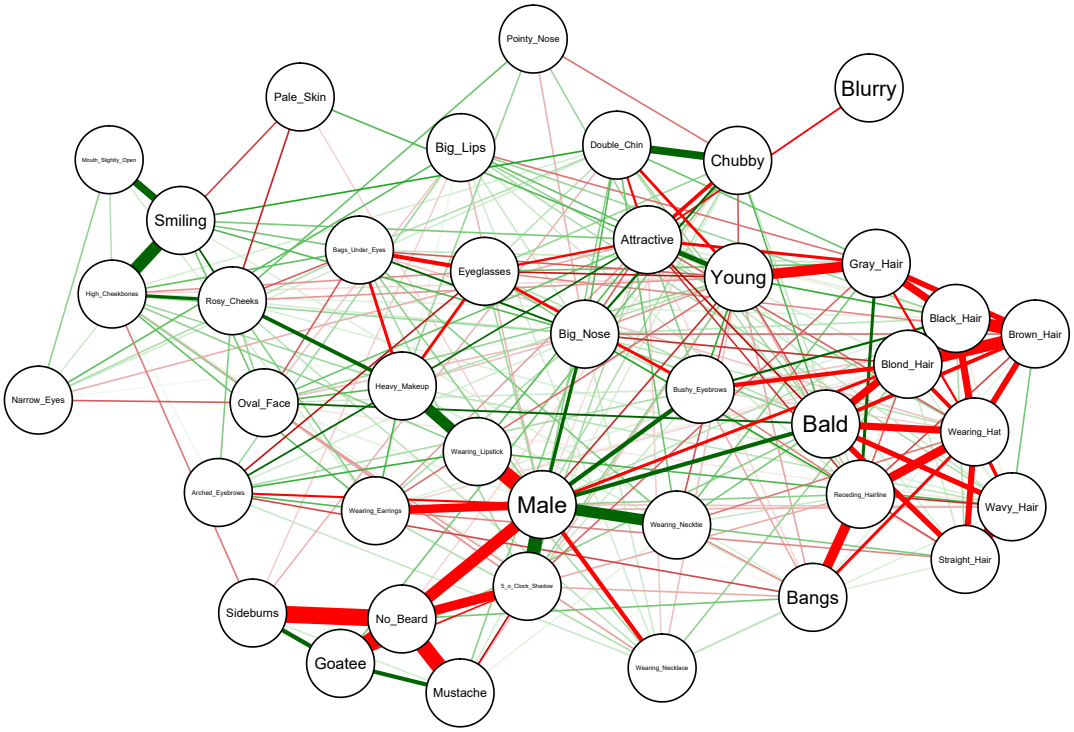

Figure 14. Visualization of Ising model of binary attributes, with green denoting positive edges and red denoting negative edges.

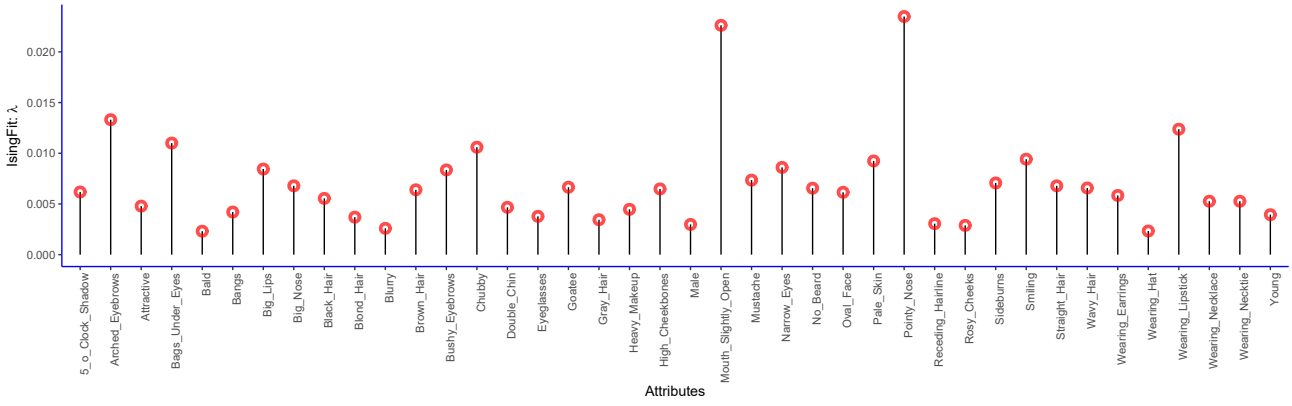

Figure 15. Attribute-wise variation of the tuning parameter (λ) that was used to achieve the best fit set of neighbors during the Ising prior learning

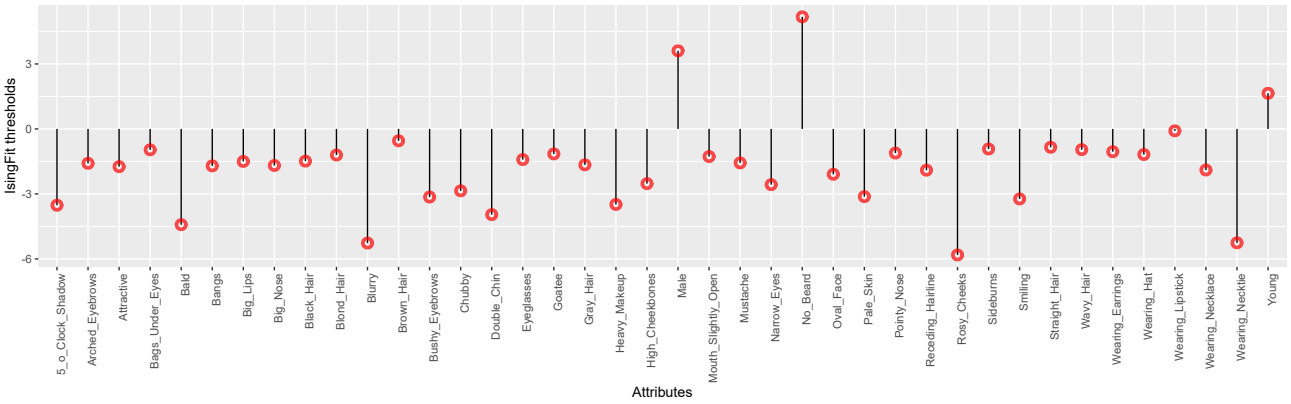

Figure 16. Attribute-wise thresholds (or *node parameters* τ_i) used during the `eLASSO` fitting procedure

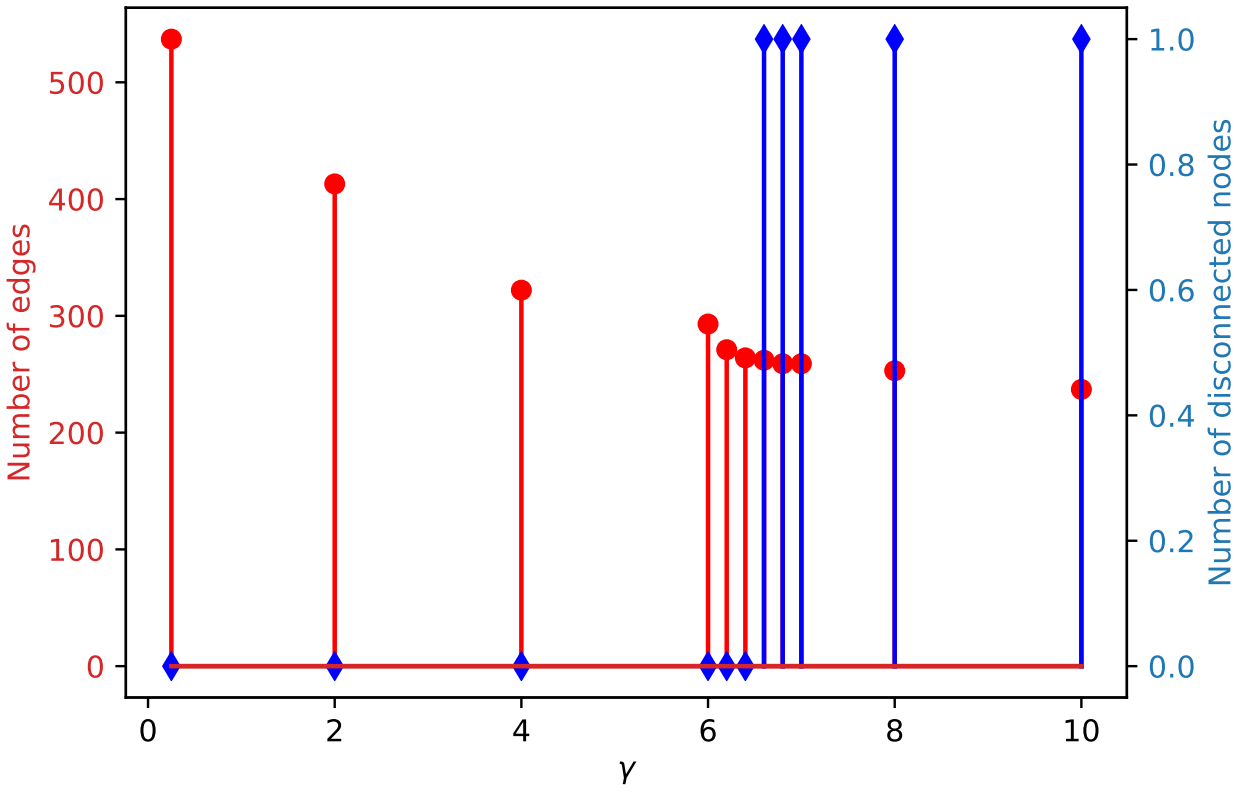

Figure 17. Sparsity penalty (γ) selection.