LARGE: Latent-Based Regression through GAN Semantics - Supplementary Materials

Yotam Nitzan* Tel-Aviv University
Rinon Gal* Tel-Aviv University
Ofir Brenner Tel-Aviv University
Daniel Cohen-Or Tel-Aviv University

A. Reproducibility

We provide the URL to the project's Github repository containing our source code: https://github.com/YotamNitzan/LARGE. Additionally, we have made an effort throughout the paper towards making the results reliable and reproducible. Specifically, as noted in Section 4, whenever feasible we repeated experiments a thousand times and report mean and standard deviation in Figures 3, 4, 1 to 3, 6 and 19. Additionally, we report detailed information on subtle aspects of the model and data in Section E.

B. Outline of StyleGAN Editing Methods

In the following section we provide a high-level outline of the StyleGAN editing methods which were used to discover the latent hyper-planes used in the core paper.

For more in-depth explanations and an outline of additional editing methods, we refer the readers to each method's respective paper [14, 20, 21] and to a recent survey [2].

B.1. InterFaceGAN

InterFaceGAN [20] introduces a simple, weakly supervised method for discovering linear editing directions in the latent space of a pre-trained GAN. The method requires access to a pre-trained binary attribute classifier (*e.g.* one that can predict whether the individual in an image is looking left or right). With such a classifier at hand, one samples a large number of latent codes (typically $500k$), uses them to synthesize images, and labels these images with the classifier. The result is a large set of pairs of latent codes and their associated classifier-label. These pairs are pruned to keep only those with the highest prediction confidence, and these are then used to learn a linear SVM in the latent space. Finally, the SVM's decision boundary serves as a hyperplane that separates the latent space into two binary regions, and the normal to this hyperplane serves as an editing direction which smoothly controls the classifier's property in generated images (*e.g.* it controls pose in the case of a left / right classifier).

B.2. SeFA

SeFA [21] introduce a closed-form method for identifying meaningful latent directions in an unsupervised manner. Their method considers the weights, A, of the first (fully connected) layer that operates on the latent codes. Their intuition is that latent directions $\mathbf{n}$ which maximize $\|A\mathbf{n}\|^2$ are those that will produce the most significant change post the linear projection, and will likely impact the generated image in a similar manner. It is further shown that when these latent directions are constrained to a unit norm, the ones that provide the largest change those eigenvectors of A^TA with the largest eigenvalues. The task of finding meaningful directions can therefore be tackled by closed-form eigendecomposition of A^TA. Not all such directions are meaningful or highly disentangled, however, and so brief human observation is required to evaluate the meaning and the quality of these direction.

B.3. StyleCLIP

StyleCLIP [14] discovers editing directions in the latent space by leveraging CLIP, a pre-trained vision-language model [16]. They consider all entries of the GAN's latent code, and modify them one at a time through $\mathbf{w}_{+,-} = \mathbf{w}_0 \pm \alpha\mathbf{e}^i$ where $\mathbf{e}^i_j = \delta_{ij}$ and $\mathbf{w}_0$ is some randomly sampled, initial code. The two codes, $\mathbf{w}_{+,-}$ are used to synthesize a pair of images which are then passed through CLIP's image encoder, resulting in two points in CLIP's embedding space. The process is repeated using a large number of initially sampled latents, w_0, and the average CLIP-space direction between each pair of generated samples is recorded. Finally, this process repeats for every entry of the latent codes.

In a sense, this process uses CLIP to encode the semantic direction of change induced by a modification of a specific entry of the latent code. At inference time, one can then use a pair of textual prompts to similarly describe a direction (*e.g.*'Face with hair' to 'Face with long hair'). These prompts are embedded into CLIP's space using the CLIP text encoder, and the direction between them can be calculated. Finally,

*Indicates equal contribution

the latent editing direction is given by determining which latent code entries produced a CLIP-space change which is closely aligned with the textually described direction (*i.e.*, which pre-recorded directions have the highest dot-product with the textual direction).

C. Ablation Study

We conduct an ablation study on several components of our method. Namely, the use of a linear regression model, our choice of inversion mechanism and the importance of our layer-weighting approach (as detailed in Subsection 3.2 of the core paper).

C.1. Evaluating Linearity

We use a linear regression model to calibrate distance features to real-world values. We next provide an additional experiment, motivating the use of a linear model over higher degree polynomials. Using the same set of labeled images and their distances from a semantic hyperplane, we compare the accuracy of a linear model trained on these data to the accuracy of polynomial models of higher degrees - specifically 2, 3 and 5. Results are shown in Figure 1, As can be seen, for all attributes tested, the linear model is superior when a few labeled samples are provided. Additionally, despite having greater expressive power, the polynomial models do not outpace the linear model even when a thousand labeled samples are provided.

C.2. Inversion Comparisons

We evaluate the performance of our model when utilizing different inversion methods in order to obtain the latent code for both train and test images. We compare our results on the human face pose and age estimation tasks using the CelebA-HQ [8] dataset. Specifically, we compare four *GAN Inversion* encoder models - pSp [17], e4e [23], ReStyle-psp and ReStyle-e4e [1] which uses the former encoders in an iterative refinement scheme. In all cases we use the official pre-trained models provided by the authors.

For reference, we also include the most related baseline, GHFeat-SVM which was introduced in Subsection 4.1. As a reminder, this baseline was devised based on the same latent-distance principles but applied in the feature space of GHFeat [25]. Results are displayed in Figure 2. As can be seen, our method outperforms GHFeat-SVM regardless of the choice of inversion method. Furthermore, e4e [23] is consistently superior to other methods, and e4e-based methods are superior to pSp-based methods. Tov *et al*. [23] demonstrated that there exists a trade-off between distortion and editability. This tradeoff stems from the ability to invert images into more semantic regions of the latent space. While the e4e encoder tends towards preserving such semantics, pSp is trained with the sole purpose of image reconstruction. As our method relies heavily on such latent space semantics, it is unsurprising to see a consistent, even though minor, advantage towards e4e-based methods.

While superior results are obtained with all inversion methods, we conclude that our method works best with semantic preserving encoders and recommend using such.

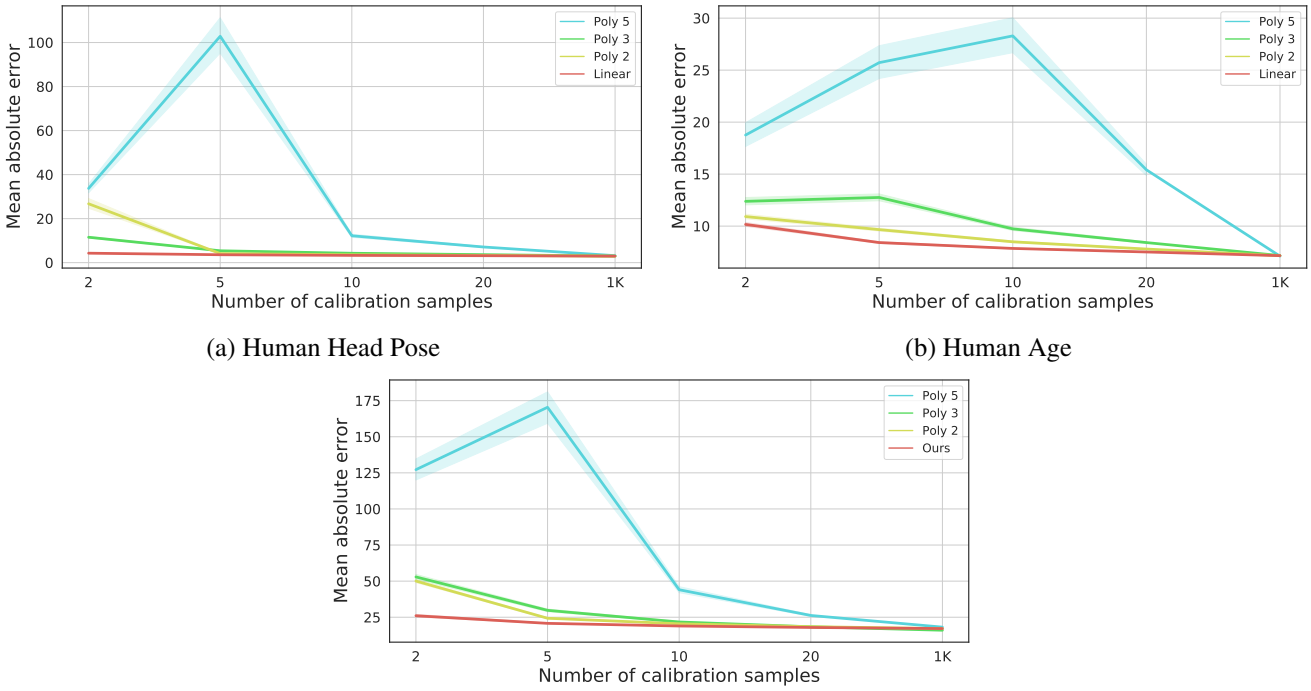
C.3. Layer-Importance Weighting

We evaluate the contribution of our per-layer latent direction weighting approach, described in Subsection 3.2 of the main paper.

We compare our proposed approach with three alternatives for computing distances between latent codes in $\mathcal{W}+$ and boundaries found in $\mathcal{W}$. First, we use a simple model dubbed "All Layers" in which we compute the distance of the latent code in each separate layer to the same $\mathcal{W}$-space boundary. We then use all such distances (18 in total) as features for the regression model. Second, we consider a model dubbed "Euclidean", in which we duplicate the boundary over all layers and compute a simple Euclidean distance. Last, we consider a model which uses only the distance along the single layer which provides optimal performance. In the case of poses, for example, this is layer 2. Note that determining the optimal layer in this manner requires a large continuously tagged dataset to evaluate against, which may not be feasible in practical applications. Our own model uses a weighted distance metric where the contribution of each layer is scaled according to our semantic-mapping importance scores determined in an unsupervised manner.

The performance of our method and all alternative is reported in Figure 3. Our weighted model consistently outperforms the other alternatives in low-supervision settings, and achieves similar results to the "All Layers" model with extensive supervision. This shows that our weighted distances accurately reflect the importance of the distance across each layer, without having to rely on the additional supervision required to determine such a weighting with multiple-feature regression.

Beside obtaining superior performance, we next demonstrate that our approach in fact identifies layers that are highly correlated with a semantic attribute. For this end, we first fit a set of linear models for the pose estimation task, using the individual distances along each layer of $\mathcal{W}+$, one at a time. The results and their R^2 coefficients are shown in Figure 5. Note that obtaining accurate correlation scores in this manner requires a large labeled dataset and may hence be unfeasible. Next, in Figure 4 we show the layer importance scores extracted by our unsupervised method. As can be seen, our unsupervised layer scoring approach successfully identifies layers with high correlation to the semantic property.

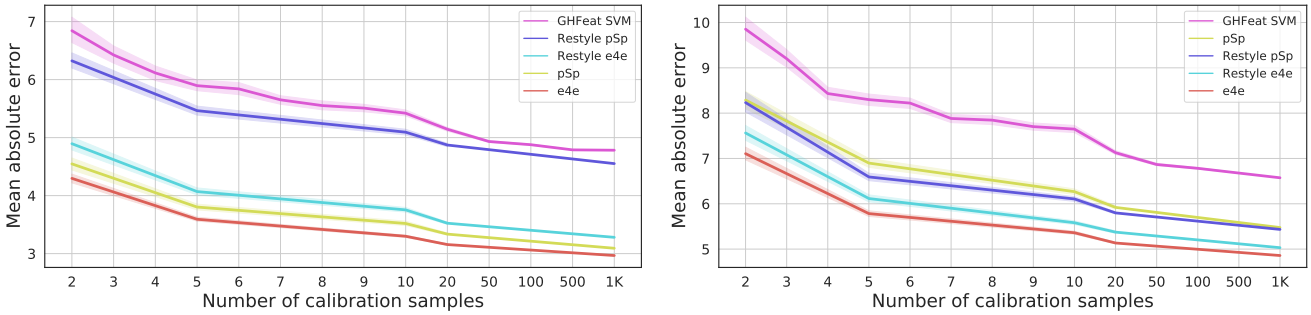


(a) Human Head Pose

(b) Human Age

(c) Car Pose

Figure 1. Quantitative comparison of four different choices for the degree of fitted regression functions. In addition to the linear model outlined in our method, we evaluate polynomial degrees of 2, 3 and 5. The linear model outperforms the alternatives when only a few samples are available, and is equivalent for a thousand samples.



(a) Pose

(b) Age

Figure 2. Quantitative comparisons of four different *GAN Inversion* encoders: e4e [23], pSp [17] and ReStyle-e4e and ReStyle-pSp [1] on the CelebA-HQ dataset [8].

C.4. Comparing Against Generating a Continuously-Labeled Data

As discussed in the paper, a popular approach to using Deep Generative Models for discriminative task is to generate labeled datasets and use them for training. Although our method provides means to perform regression directly in the latent space, it may also be used to generate labeled datasets. After calibrating the model, we can apply it to any latent code, including codes simply sampled from the Gaussian prior. We can thus generate a new dataset from randomly sampled codes, and dictate their labels by the latent regression model.

We perform an experiment where we generate such a dataset for human head pose and train a regression CNN directly on the generated images. To make sure the attribute varies enough in the generated dataset we perform the following process. We sample a latent code $w \in \mathcal{W}$ and a scalar $\alpha \in [-9, 9]$ which was observed to be the maximal range for which the generated image does not degrade in quality. We then edit the sampled latent code by applying

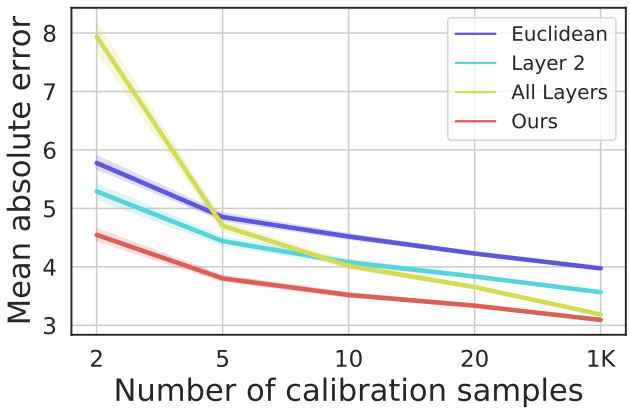
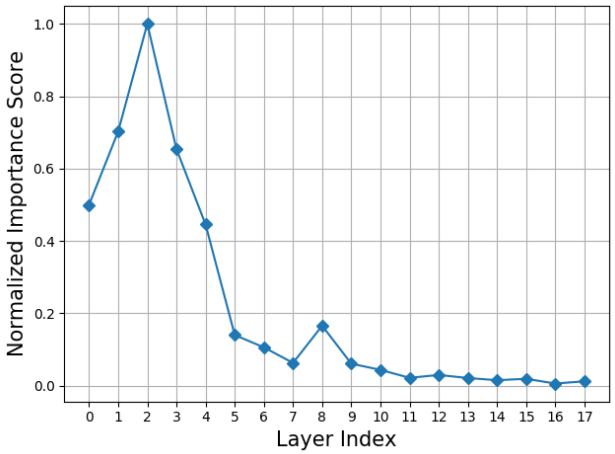


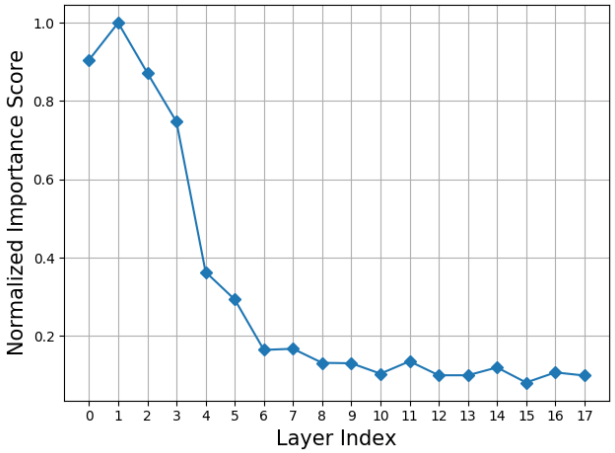
Figure 3. Comparing several approaches for calculating latent-space distances between codes in $\mathcal{W}+$ and boundaries in $\mathcal{W}$. The "Euclidean" model calculates the Euclidean distance between the latents and a boundary obtained by replicating the $\mathcal{W}$ boundary along all layers of $\mathcal{W}+$. "All Layers" calculates a per-layer distance and uses all 18 distances as features. "Layer 2" uses only the distance calculated on layer 2 of the latent code, which was experimentally observed to provide the best single-layer results for pose. Finally, our model uses a weighted distance as outlined in Subsection 3.2 of the main paper. As can be seen, our proposed method is superior to other method in the few-shot domain and is matched by "All Layers" only when provided with a thousand labeled samples.

$w' = w + \alpha\vec{n}$. Now, we generate the image $I' = G(w')$ and infer the head pose by inputting α to the regression model. For the regression model, we use a model trained with 1000 labeled samples. Repeating this process 45K times, we now posses a continuously-labeled, roughly balanced generated dataset. We train multiple CNN backbones for the task of regression and test them over the annotated CelebA-HQ test set [8]. The lowest Mean Average Error, 3.23 ± 3.67, was obtained with EfficientNet-b3 [22]. For comparison, applying our approach with exactly the same set and supervision obtained 2.97 ± 2.76.

We conclude that it is preferable to apply our method directly to regress test images, rather than generating a labeled set for downstream training. We speculate that a possible explanation for the degradation in performance is the domain gap. In the generated-dataset approach, the classifier is trained on a generated dataset and tested on a real one, without adaptation. On the other hand, in our approach, the *GAN Inversion* encoder may mitigate some of that gap. Additionally, the generator and inversion encoder are trained once per-domain while the latent and CNN regressor as well as the data generation happens once per attribute. As a result, our approach, requiring just the training of a latent simple regression model requires roughly x1000 less time to train.



(a) Un-normalized



(b) Normalized

Figure 4. The results of our per-layer importance scores approach as outlined in Subsection 3.2, for the head pose attribute. (a) Un-normalized importance scores, before accounting for the scale of gradients in each layer. (b) Normalized importance scores, after accounting for gradient scales.

D. Additional Results

D.1. Calibrated Results - Cars

We repeat the pose experiment of section 4.2 on the car image domain. We compare our model to SSV. Our model utilizes the official StyleGAN2 [11] LSUN Car [28] pre-trained checkpoint, and the official e4e [23] inversion encoder trained on the train split of Stanford Cars [12]. The semantic boundary was extracted using InterFaceGAN [20]. SSV was trained as in the original paper, using the Comp-Cars dataset [26]. Both models were evaluated on the test split of Stanford Cars, with pose labels acquired through Pose Contrast [24]. After labeling the test-set images, we discarded all images with yaw angles exceeding 90° in ei-

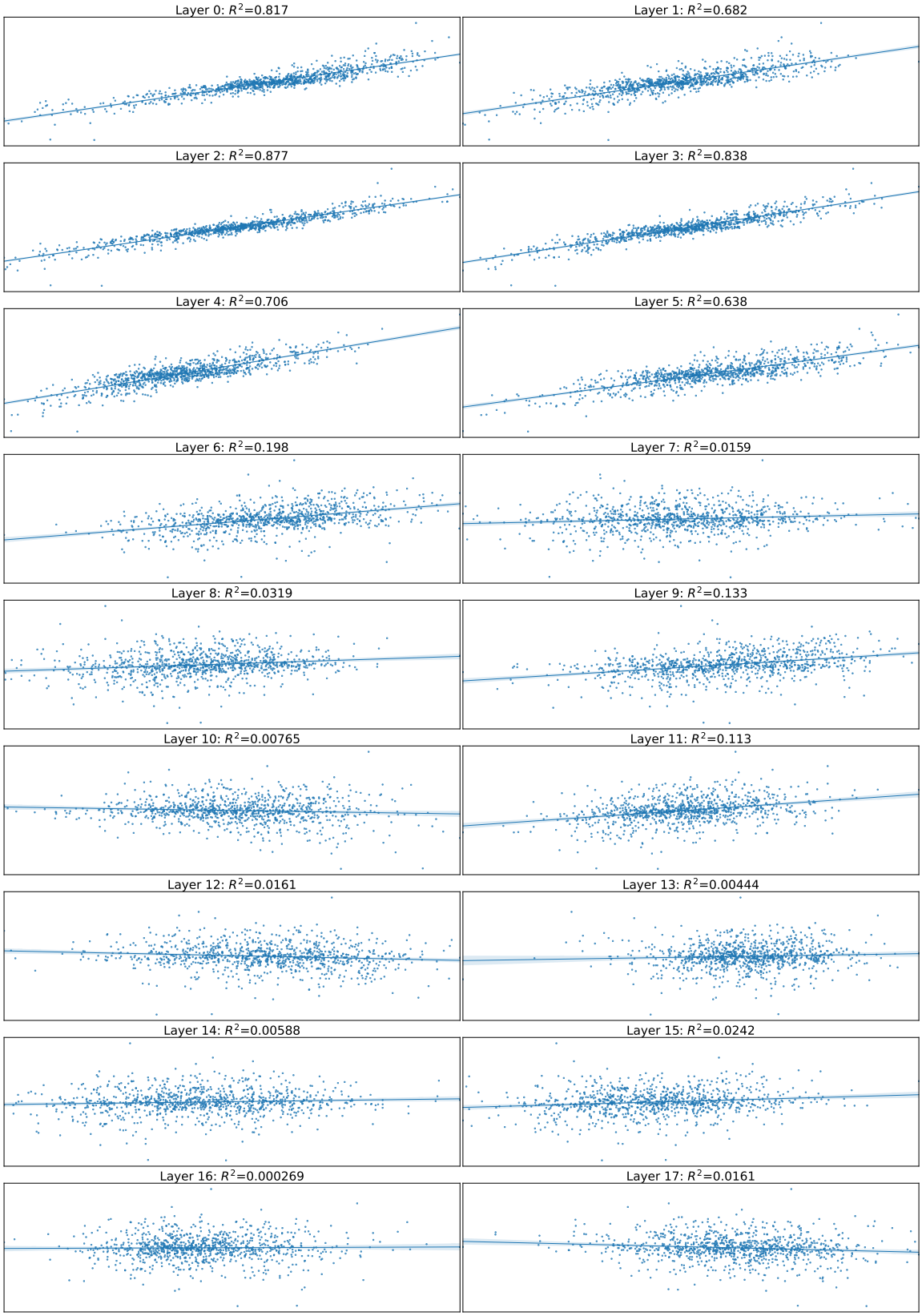


Figure 5. Measuring the linear correlation of yaw angle with distance from hyperplane for each layer separately. In each subplot, the x-axis is the distance of this layer in the latent code from the boundary while the y-axis is the ground truth yaw angle. As can be seen, distance in first layers are better linearly correlated to head pose than last layers.

ther direction, *i.e*. we evaluated only on images for which $\theta_{yaw} \in [-90°, 90°]$.

The results are shown in Figure 6. Our model outperforms SSV over all tested supervision ranges, indicating that it can generalize well to the car domain.

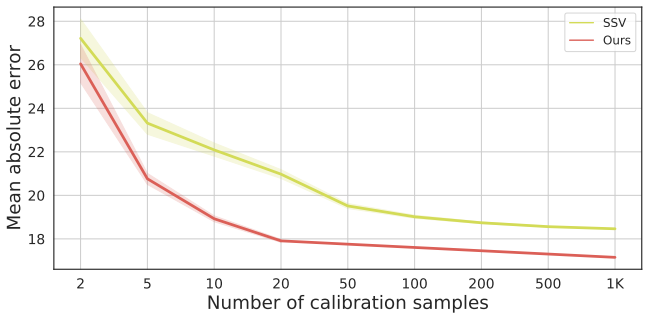


Figure 6. Pose estimation error comparisons on the Stanford Car [12] dataset tagged by Pose Contrast [24], as a function of the number of labeled images used for calibration.

D.2. Uncalibrated Results

As discussed in the core paper, our method can be applied to downstream tasks even in the absence of direct supervision. We demonstrated the applicability of our method to an image sorting task. Here, we demonstrate an application to ordinal regression. Specifically, we perform sentiment analysis on facial images, dividing them into four bins that represent different levels of contentment.

To perform ordinal regression, we use our method to calculate an uncalibrated distance score for each image, as described in section 3. Then, we simply divide the range of distances into four bins. As can be seen in Figure 7, even this simple method obtains good results.

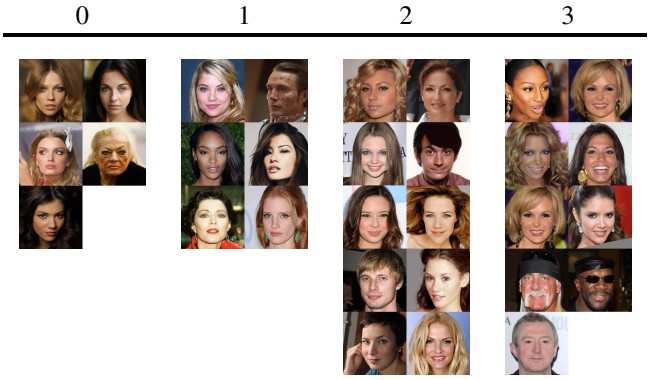


Figure 7. Ordinal regression applied to sentiment analysis using our method. Images are divided into bins, from discontent - 0 to most content - 4. All images were randomly sampled from from CelebA-HQ [8]. Sentiment is measured by distance from a 'smiling' semantic boundary, identified by StyleCLIP [14].

In Figures 8 to 13, we show additional sorting results on cats and dogs, using a StyleGAN-ADA [9] model trained on the AFHQ dataset [5]. Latent semantics were identified using SeFA [21]. In Figures 14 to 17, we show additional sorting results on leaves, using a StyleGAN-ADA model trained on the Plant-Village [7] dataset. Latent semantics were identified using InterFaceGAN [20].

D.3. Additional Applications of Unsupervised Layer-Importance Weighting

We further investigate the effects of our layer-importance weighting approach by considering its effects on image editing. To do so, we discover pose editing directions in $\mathcal{W}$ for the FFHQ [11] and AFHQ Cat [9] models. The directions were discovered using InterFaceGAN [20] and SeFA [21] respectively. We then invert real images into $\mathcal{W}+$ and edit them in two manners. First, we use the conventional way of applying the linear editing direction equally to all layers along $\mathcal{W}+$. Second, we apply the linear direction differently in each layer by multiplying the step size with the per-layer importance score.

The results are shown in Figure 4. As can be seen, by weighting the layers appropriately we can increase the level of disentanglement and better avoid spurious changes. For example, the conventional editing of head pose for humans also wrongly affects the eye-gaze – maintaining eye-contact with the viewer. For cats, the pose direction affects the "identity" and background. Using our per-layer weighting, these effects are mitigated.

These results further highlight the need for appropriate layer weighting when calculating distances in different latent-spaces. If we were to simply calculate the distance in the naïve manner, our pose regression results would be affected by these entangled properties - *e.g*. gaze or even background color.

E. Complementing Experiments' Details

We provide additional details about experiments conducted in the main paper.

E.1. Accuracy of Trained SVMs

In Subsection 4.1 of the paper, we demonstrate that distances in the latent space of the GAN are more semantically meaningful and better behaved than equivalent distances in alternative feature spaces. For this end, we train SVMs in all feature spaces. In Table 1 we report the accuracy of those SVMs on validation sets. As can be seen, the gap in performance for the task of regression cannot be easily explained by the performance of the SVM as a binary classifier. This is further evidence that StyleGAN's latent space possesses unique properties which make it suitable for the task of regression.

E.2. What Points to Label?

In order to calibrate the latent distances to actual real-world values, a few labeled samples are required. These labeled samples are then used to train a simple linear regression model. In some real-world scenarios, one won't have pre-defined disjoint sets of labeled and unlabeled samples. Rather, as most datasets form, at first the dataset is simply a collection of unlabeled samples and only later will those be annotated. While our method performs better as it gets more labeled samples, working with a few labeled samples is usually preferred. We thus provide some simple practical suggestions as to what samples one should label. Following these suggestions is increasingly more important as less points are sampled.

First we suggest to invert the unlabeled dataset to the

Figure 8. Sorting images from AFHQ-dog [5] using a "fur fluffiness" semantic directions extracted by SeFA [21].

Figure 9. Sorting images from AFHQ-dog [5] using a "head pitch" semantic directions extracted by SeFA [21].

Figure 10. Sorting images from AFHQ-dog [5] using a "head yaw" semantic directions extracted by SeFA [21].

Figure 11. Sorting images from AFHQ-cat [5] using an “age” semantic directions extracted by SeFA [21].

Figure 12. Sorting images from AFHQ-cat [5] using a “head pitch” semantic directions extracted by SeFA [21].

Figure 13. Sorting images from AFHQ-cat [5] using a “head yaw” semantic directions extracted by SeFA [21].

latent space and obtain the distances from the hyperplane corresponding to the semantic latent direction, which results in a distribution of distances. This process does not require any labels. Now, we suggest choosing and labeling a set which is roughly evenly-spaced throughout the center of the distribution. There are two motivations for this sampling strategy, stemming from a single simple principle - sampling points that best represent the distribution. First, samples on the edge of the distribution are more likely to be outliers. Latent outliers may come about when the original image is in itself an outlier in the image distribution. Thus, discarding the noisy edges and sampling from the center of distribution is likely to better represent the dataset. Second, sampling points which are ”close” to each other on the distance axis, is prone to error. The linear relationship between the attribute and distance is modeled by $y = a \cdot d + b + \varepsilon$ where a, b are the

Table 1. Validation accuracy of SVM baseline models operating on pose and age, using the different feature spaces described in Subsection 4.1. As can be seen, most model are decent classifiers.

Feature space	Pose	Age
Ours	**0.93**	0.82
Pixel	0.87	0.74
Binary-cls	0.91	**0.88**
GHFeat [25]	0.91	0.83
ImageNet [29]	0.73	0.84
ID [6]	0.65	0.85
SwaV [3]	0.71	**0.88**

function coefficients and ε is an error term. Consider the case of sampling two points with distances d_1, d_2. When $\Delta d = d_1 - d_2$ is small, it may be the case that $\varepsilon > a \cdot \Delta d + b$. In such case, the noise in the observed attribute may overwhelm any signal due to the modified latent-distance, and a linear model fit to these points will fail to predict the underlying a, b. Sampling a set which is roughly evenly-spaced maximizes the minimal distance between any two points.

We follow these guidelines when conducting all experiments described in the paper. For the center, we consider 95% of the data. For evenly-spaced distances we first observe that for n points sampled from a uniform distribution over $[a, b]$, the minimal distance between a pair of samples is smaller or equal to $\frac{b-a}{n}$. However, choosing such a minimal distance will only allow for, at most, one sampled set. To allow greater flexibility in the choice of samples, we loosen the restriction, and sample points whose minimal allowed

Figure 14. Sorting images from Plant-Village [7] using sick-to-healthy semantic directions extracted by InterFaceGAN [20]. To facilitate easy visual comparisons, all sick leaves have the same disease - “Early Blight”.

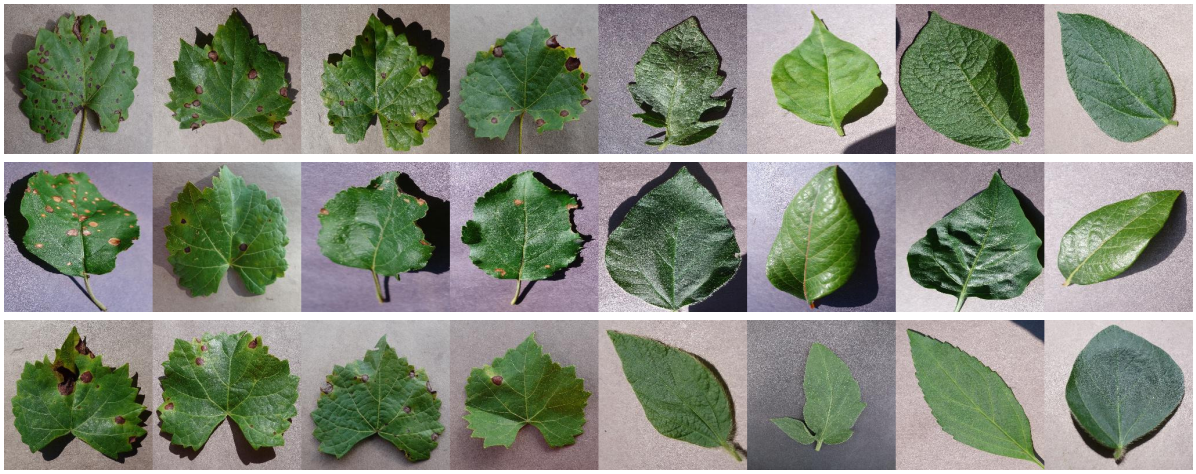


Figure 15. Sorting images from Plant-Village [7] using sick-to-healthy semantic directions extracted by InterFaceGAN [20]. To facilitate easy visual comparisons, all sick leaves have the same disease - “Black Rot”.

Figure 16. Sorting images from Plant-Village [7] using sick-to-healthy semantic directions extracted by InterFaceGAN [20]. To facilitate easy visual comparisons, all sick leaves have the same disease - "Rust".

Figure 17. Sorting images from Plant-Village [7] using sick-to-healthy semantic directions extracted by InterFaceGAN [20]. To facilitate easy visual comparisons, all sick leaves have the same disease - "Late Blight".

difference is $\frac{b-a}{n^{1.3}}$.

E.3. Regression Model Regularization

Our final regression model is a simple linear regression model. However, there is still room to choose regularization. We experiment with our model without regularization, with $L1$ regularization (*i.e*. Lasso), $L2$ regularization (*i.e*. Ridge) or both (*i.e*. ElasticNet). We find that there's only a slight difference in the few-shot setting and it diminishes as the number of samples increases, as demonstrated in Figure 19. We used ElasticNet regularization in all experiments presented in the paper. We use the default penalty weighting provided by scikit-learn [15].

F. Linearity Origin Hypothesis

Our method uses linear regression to calibrate distance features to real-world values. This choice was motivated by empircal results (see Subsection C.1). Nevertheless, we hypothesize on the origin of this property. As datasets are non uniform, the model is tasked with representing values with different densities. One solution would be to "allocate" more space in $\mathcal{W}$ for more frequent values, creating a non-linear space. However, the model already has a mechanism to represent non-uniform densities - through the original, Gaussian distribution of $\mathcal{Z}$ space and it's non-linear mapping into $\mathcal{W}$. We thus speculate the model has no incentive to model densities in the $\mathcal{W}$ space itself. Instead, it can model a simpler, linear space, in order to make the generative process simpler. Such a scenario is in line with the unwarping intuition provided in the original StyleGAN paper [10].

Figure 18. Using the layer-importance weighting to improve pose editing. Our approach reduces changes in unrelated properties. In the case of human faces, we observe that layer weighting can prevent the gaze from changing along with the pose. In the case of cats, large pose changes lead to severe changes in background and identity. Appropriate layer weighting reduces these effects.

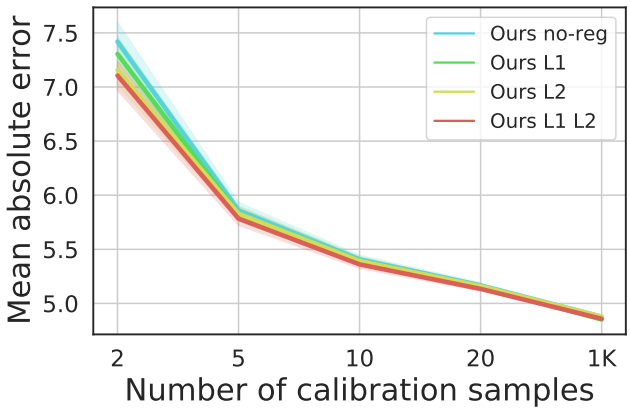


Figure 19. We compare the results of our approach using different types of regularization on the simple linear regression model. As can be observed, only slight difference exist when two calibration points are used, and the difference diminishes as more points are sampled.

Table 2. Datasets and models used in our work and their respective licenses.

Dataset	Source	License
FFHQ	[10]	CC BY-NC-SA 4.0*
CelebA-HQ	[8]	CC BY-NC 4.0
AFHQ	[5]	CC BY-NC 4.0
Stanford Cars	[12]	ImageNet License
CompCars	[26]	Non-Commercial Research
CACD	[4]	Academic Research
PlantVillage	[7]	CC0 1.0

(a) Datasets

Model	Source	License
StyleGAN2	[11]	Nvidia Source Code License-NC
GHFeat	[25]	No License
SSV	[13]	Nvidia Source Code License-NC
Scikit-Learn	[15]	BSD 3-Clause
WHENet	[30]	BSD 3-Clause
DEX	[18]	No License
pSp	[17]	MIT License
e4e	[23]	MIT License
ReStyle	[1]	MIT License
InterFaceGAN	[20]	MIT License
SeFa	[21]	MIT License
StyleCLIP	[14]	MIT License
CLIP	[16]	MIT License
PoseContrast	[24]	MIT License
FSA	[27]	Apache License V2.0
StyleGAN2-pytorch	[19]	MIT License
StyleGAN-ADA	[9]	Nvidia Source Code License

(b) Models

G. Licenses and privacy

The datasets and models used in our work and their respective licenses are outlined in Table 2.

Some of the datasets in use, an in particular FFHQ [10], CelebA-HQ [8] and CACD [4], contain personally identifiable data in the form of face images.

We have not reached out to receive consent from the individuals portrayed in the images. However, all three image sets are composed of publicly available celebrity images or faces of individuals crawled from flicker, all of which were uploaded under permissive licenses which allow free use, redistribution, and adaptation for non-commercial purposes. The curators of all sets provide contact details for individuals who wish to have their images removed from the set.

H. User Study

In this section we provide all the details of our user study.

The user study was conducted through https://freeonlinesurveys.com/. It was performed over a period of 5 days, with a total of 62 different responders. Individual questions saw anywhere from 20 to 62 responses

Table 3. User study answer key and the number of responders that picked each answer

Question	Ours	CLIP	Random
	Hair color		
1	**Bottom (46)**	Top (16)	Middle (0)
2	**Middle (50)**	Top (11)	Bottom (1)
3	**Bottom (34)**	Top (27)	Middle (1)
4	**Middle (53)**	Top (9)	Bottom (0)
5	**Bottom (45)**	Middle (17)	Top (0)
	Makeup		
6	**Bottom (45)**	Top (8)	Middle (4)
7	**Bottom (35)**	Top (9)	Middle (13)
8	**Middle (36)**	Bottom (17)	Top (4)
9	**Bottom (41)**	Middle (11)	Top (5)
10	**Top (44)**	Middle (7)	Bottom (6)
	Expression		
11	**Bottom (40)**	Top (13)	Middle (2)
12	**Bottom (46)**	Top (9)	Middle (0)
13	**Bottom (38)**	Top (15)	Middle (2)
14	Top (19)	**Bottom (35)**	Middle (1)
15	Bottom (4)	**Middle (47)**	Top (4)
	Hair length		
16	**Top (52)**	Bottom (1)	Middle (2)
17	**Middle (46)**	Top (1)	Bottom (8)
18	**Top (33)**	Middle (0)	Bottom (22)
19	**Middle (50)**	Bottom (1)	Top (4)
20	**Middle (52)**	Top (3)	Bottom (0)

(a) Human faces

Question	Ours	SSV	Random
	Yaw		
21	Bottom (4)	**Middle (18)**	Top (1)
22	**Top (22)**	Middle (0)	Bottom (1)
23	**Top (20)**	Middle (2)	Bottom (1)
	Pitch		
24	**Bottom (17)**	Middle (0)	Top (3)
25	**Top (16)**	Bottom (3)	Middle (1)
26	**Top (17)**	Bottom (1)	Middle (2)
27	**Middle (16)**	Bottom (2)	Top (2)

(b) Cats

(see Table 3 for exact numbers). All responders were unpaid volunteers which responded (anonymously) to a link shared among colleagues and acquaintances of the authors.

We used a three-alternative forced choice setting. Users were provided with randomly sampled sets of 10 images, sorted in three manners - once using our method, once by randomly assigning an order and once by using a dedicated baseline. For each question, the visual order of the three sorting options was randomized. Users were asked to choose the order that better matches a textual description.

For the human face domain, we used the same textual prompts as Figure 5 in the main paper, with 5 randomly sampled image sets for each prompt. For a baseline, we sorted the images according to their cosine-distances from the same textual prompts directly in the CLIP [16] embedding space. For the cat domain we used the pitch and yaw directions displayed in Figure 7 in the main paper and for a baseline we used pose predictions from SSV [13] trained on AFHQ-Cat [5].

The questions and their associated image sets are shown in Figure 20. In Table 3 we provide the list of the methods used to generate each row of each question (*i.e.* the answer key), along with the number of responders who chose each answer.

Hair Color

Choose the row in which images are better sorted from black hair (left) to blonde hair (right):

1)

A 

B 

C 

2)

A 

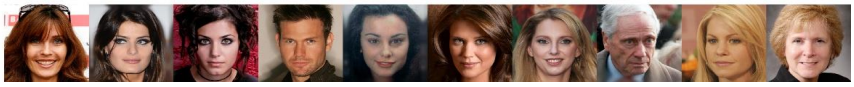
B 

C 

3)

A 

B 

C 

4)

A 

B 

C 

5)

A 

B 

C 

Figure 20. All questions asked in our survey and their associated images. Page 1/6.

Makeup

Choose the row in which images are better sorted from less makeup (left) to more makeup (right):

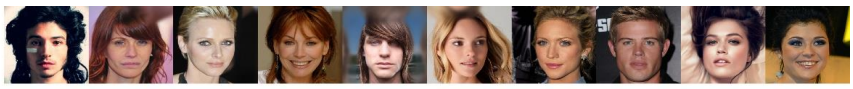
6)

A 

B 

C 

7)

A 

B 

C 

8)

A 

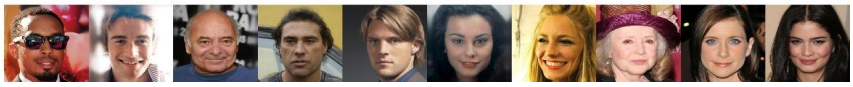
B 

C 

9)

A 

B 

C 

10)

A 

B 

C 

Figure 20. All questions asked in our survey and their associated images. Page 2/6.

Expression

Choose the row in which images are better sorted from a sad expression (left) to a happy expression (right):

11)

A

B

C

12)

A

B

C

13)

A

B

C

14)

A

B

C

15)

A

B

C

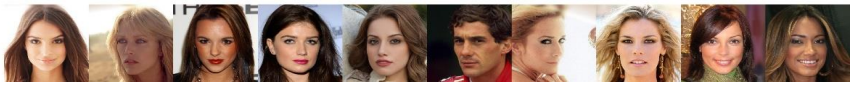


Figure 20. All questions asked in our survey and their associated images. Page 3/6.

Hair Length

Choose the row in which images are better sorted from short hair (left) to long hair (right):

16)

A 

B 

C 

17)

A 

B 

C 

18)

A 

B 

C 

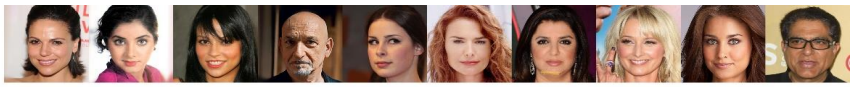
19)

A 

B 

C 

20)

A 

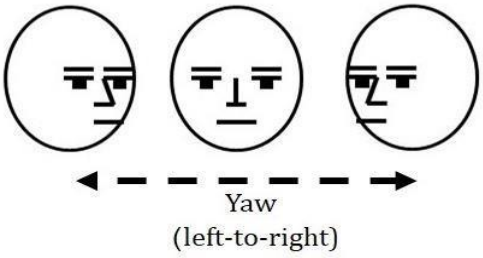
B 

C 

Figure 20. All questions asked in our survey and their associated images. Page 4/6.

Cats Yaw

Choose the row in which images are better sorted according to their left-to-right head angle (yaw):



21)

A 

B 

C 

22)

A 

B 

C 

23)

A 

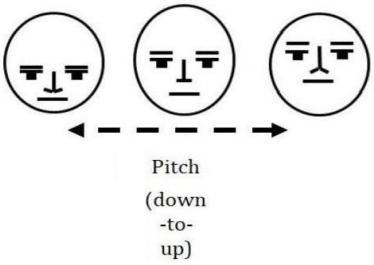
B 

C 

Figure 20. All questions asked in our survey and their associated images. Page 5/6.

Cats Pitch

Choose the row in which images are better sorted according to their up-to-down head angle (pitch):



24)

A

B

C

25)

A

B

C

26)

A

B

C

27)

A

B

C

Figure 20. All questions asked in our survey and their associated images. Page 6/6.

References

[1] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. *arXiv preprint arXiv:2104.02699*, 2021. 2, 3, 11

[2] Amit H. Bermano, Rinon Gal, Yuval Alaluf, Ron Mokady, Yotam Nitzan, Omer Tov, Oren Patashnik, and Daniel Cohen-Or. State-of-the-art in the architecture, methods and applications of stylegan. *ArXiv*, abs/2202.14020, 2022. 1

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020. 9

[4] Bor-Chun Chen, Chu-Song Chen, and Winston H. Hsu. Cross-age reference coding for age-invariant face recognition and retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. 11

[5] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 6, 7, 8, 11, 12

[6] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 9

[7] David Hughes, Marcel Salathé, et al. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:1511.08060*, 2015. 6, 9, 10, 11

[8] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 2, 3, 4, 6, 11

[9] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020. 6, 11

[10] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 10, 11

[11] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 4, 6, 11

[12] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 4, 6, 11

[13] Siva Karthik Mustikovela, Varun Jampani, Shalini De Mello, Sifei Liu, Umar Iqbal, Carsten Rother, and Jan Kautz. Self-supervised viewpoint learning from image collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3971–3981, 2020. 11, 12

[14] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. *arXiv preprint arXiv:2103.17249*, 2021. 1, 6, 11

[15] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011. 10, 11

[16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 1, 11, 12

[17] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. *arXiv preprint arXiv:2008.00951*, 2020. 2, 3, 11

[18] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Dex: Deep expectation of apparent age from a single image. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, December 2015. 11

[19] Kim Seonghyeon. Stylegan2-pytorch. https://github.com/rosinality/stylegan2-pytorch, 2020. 11

[20] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9243–9252, 2020. 1, 4, 6, 9, 10, 11

[21] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. *arXiv preprint arXiv:2007.06600*, 2020. 1, 6, 7, 8, 11

[22] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 4

[23] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *arXiv preprint arXiv:2102.02766*, 2021. 2, 3, 4, 11

[24] Yang Xiao, Yuming Du, and Renaud Marlet. Posecontrast: Class-agnostic object viewpoint estimation in the wild with pose-aware contrastive learning. In *ArXiv*, 2021. 4, 6, 11

[25] Yinghao Xu, Yujun Shen, Jiapeng Zhu, Ceyuan Yang, and Bolei Zhou. Generative hierarchical features from synthesizing images. *arXiv e-prints*, pages arXiv–2007, 2020. 2, 9, 11

[26] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3973–3981, 2015. 4, 11

[27] Tsun-Yi Yang, Yi-Ting Chen, Yen-Yu Lin, and Yung-Yu Chuang. Fsa-net: Learning fine-grained structure aggregation for head pose estimation from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1087–1096, 2019. 11

[28] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a

large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 4

[29] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 9

[30] Yijun Zhou and James Gregson. Whenet: Real-time fine-grained estimation for wide range head pose. *arXiv preprint arXiv:2005.10353*, 2020. 11