ADVERSARIAL ATTRIBUTE LEARNING BY EXPLOITING NEGATIVE CORRELATED ATTRIBUTES

Anonymous authors
Paper under double-blind review

ABSTRACT

A typical method for classifying visual attributes in images to use convolutional neural networks (CNNs) with multi-task learning. However, this approach often suffers from negative transfer, which means that classifiers trained together to classify multiple attributes at a time perform worse than classifiers trained separately. Many multi-task learning techniques attempt to circumvent this issue, but we are interested in negative transfer itself from a different point of view: can we take advantage of negative transfer to improve our classifiers? In this paper, we propose adversarial attribute learning (AAL) where two classifiers compete with each other so that the primary classifier can learn a representation that is invariant to an attribute exhibiting negative transfer. Our experiments on human attribute classification datasets demonstrate that our method can take advantage of this negative relationship.

1 INTRODUCTION

Identifying visual attributes of objects and images is a fundamental problem in computer vision (Farhadi et al., 2009; Scheirer et al., 2012; Johnson & Grauman, 2011; Lampert et al., 2013), with a wide range of real-world applications including image retrieval (Siddiquie et al., 2011; Kumar et al., 2011), face recognition (Hu et al., 2017; Jiang et al., 2019), few-shot learning (Fu et al., 2018), etc. As has become common across many problems in computer vision, the de facto standard technique for attribute classification is the Convolutional Neural Network (CNN).

Typical applications require classifying multiple attributes at a time, so this creates a choice: we can either train a separate CNN for each attribute, or perform join training where multiple attribute classifiers share at least some layers of the CNN. The latter is appealing because it is usually more efficient in terms of both training time and model size, and sometimes also improves the classification accuracy overall, presumably because the representation learned for one attribute is helpful for recognizing another. For example, in the CelebA (Liu et al., 2015) dataset, we have found that the "Straight Hair" attribute is better predicted when sharing representations with "Gray Hair."

Unfortunately, this accuracy boost is not universal: sometimes classifiers trained independently on two attributes perform better than when trained jointly He et al. (2017); Lu et al. (2017); Hand & Chellappa (2017); Sener & Koltun (2018). We observe this *negative transfer* problem among many attributes in CelebA: both "Straight Hair" and "Gray Hair" are classified more accurately when trained separately than when trained with another attribute, "Big Lips." This suggests that sharing representations for hair style and lip size is harmful presumably because they are different facial parts. A straightforward solution would be not to share representation for those that have negative transfer. However, not only using separate representations, we would like to take advantage of this negative relationship for further improving the representation. We interpret this negative transfer as that representations for hair style and lip size should be invariant to each other. Thus we would like to design a method to explicitly encourage the CNN representation to learn this invariance.

Some work has attempted to avoid negative transfer. A classical view in multi-task learning assumes that features across tasks should have a common subspace (Argyriou et al., 2008). In the deep learning era, this view lies in the effort to design elegant layer-sharing strategies (Yang & Hospedales, 2016; Lu et al., 2017; Lee et al., 2018; Yang & Hospedales, 2017; Long et al., 2017; Meyerson & Miikkulainen, 2018).

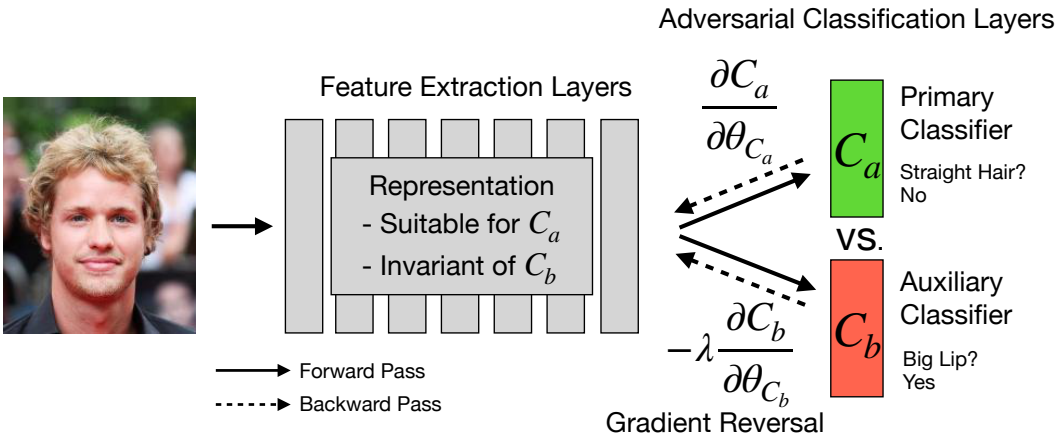

Figure 1: **Framework of Adversarial Attribute Learning (AAL)**. Our adversarial training has two competing classifiers with shared CNN layers. The primary classifier (C_a) uses the CNN representation to classify the main attribute (e.g., Straight Hair), and at the same time, the auxiliary classifier (C_b) encourages a representation not sensitive to the secondary attribute (e.g., Big Lips). In the end, the representation should be more suitable to classify the main attribute. It is intuitive, for example, that a representation that is trained invariant of lip size is better to predict hair style, as they are totally different facial parts. To make this training possible, we re-purpose a domain adaptation technique called gradient reversal (see Sec.3.2).

In this paper, we take a fundamentally different and orthogonal approach, by viewing these negative transfers as an *opportunity to help improve learning* instead of as a weakness to avoid. Our hypothesis is that representations for an attribute can be improved if it is encouraged to be invariant to another attribute with which it exhibits negative transfer. Using the same example above, a representation to classify hair style should not be affected by the size of lips, so we encourage the representation of hair style to be invariant of lip size. To achieve this, we propose an adversarial attribute classification approach (see Figure 1). We use two neural networks, a primary CNN that learns the representation to classify an attribute, and an auxiliary classifier that predicts another attribute exhibiting negative transfer from the CNN representation. The CNN not only classifies the main attribute but also tries to make the auxiliary classifier fail to predict the negative transfer attribute. These two networks are jointly trained against each other as a min-max game. At the equilibrium, the CNN representation should be invariant to the negative attribute.

To summarize, we have several contributions. (1) To the best of our knowledge, we for the first time introduce the idea of Adversarial Attribute Learning (AAL) by directly exploiting and utilizing contradictory attributes to improve attribute prediction. (2) Such an idea is implemented in a min-max optimization similar to Generative Adversarial Network (GANs), which is also for the first time employed to address attribute prediction. (3) A gradient reversal technique, which is oriented in domain adaptation (Ganin & Lempitsky, 2015), is introduced to this problem to help optimize the min-max attribute prediction game. (4) Our extensive experiments and ablation study on attribute datasets (Liu et al., 2015; Lin et al., 2019) reveal that AAL benefits from negative-transferred attribute pairs (Sec 4).

2 Related Work

Visual attributes Attributes have many useful applications in computer vision (Siddiquie et al., 2011; Kumar et al., 2011; Hu et al., 2017; Jiang et al., 2019; Johnson & Grauman, 2011; Lampert et al., 2013), and estimating visual attributes in images has been extensively studied. Classical work (Farhadi et al., 2009; Johnson & Grauman, 2011) uses manually-designed features to classify attributes, while recent work (He et al., 2017; Lu et al., 2017; Liu et al., 2015; Hand & Chellappa, 2017) uses deep models such as CNNs to learn attribute classification models in an end-to-end manner. Instead of learning separate models for each visual attribute, these CNNs are often trained in a multi-task learning framework so that different attributes share internal representations.

Multi-task/multi-label learning Attribute estimation is a multi-label prediction problem (Gong et al., 2014; Tsoumakas & Katakis, 2007) and is thus often posed as multi-task learning Caruana (1997), which has a long history in machine learning. Here we summarize recent progress in the context of deep learning. One direction of investigation is how exactly to share parameters across the attribute models, including simply sharing some internal layers, softly sharing with regularization (Yang & Hospedales, 2016), and more complex approaches (Long et al., 2017; Meyerson & Miikkulainen, 2018). Another direction is how to balance the loss among the tasks. Kendall et al. (2018) propose an approach to use the uncertainty of each task for loss weighting. In terms of attribute recognition, Lu et al. (2017) use a fully-adaptive method to determine the sharing structure, and He et al. (2017) propose a way to dynamically balance the loss among attributes. Our work is related but orthogonal to these papers, because we do not focus on how to improve the multi-task learning. In fact, although we use an auxiliary task to predict another attribute, our final model consists of a single classifier per attribute, just like single task learning. We adapt a basic multi-task learning framework to find attributes exhibiting negative transfer, but unlike work that tries to alleviate the negative transfer, our purpose is to actually exploit it.

Adversarial networks Our adversarial training approach is the same min-max optimization that is widely used in Generative Adversarial Networks (GANs) (Goodfellow et al., 2014). Typical GANs consist of two neural networks: a generator that tries to produce realistic images and a discriminator that tries to distinguish generated images from real. The two networks are trained against jointly, so that over time the generator produces more and more realistic images (Brock et al., 2019). While our work has nothing to do with image generation, our approach can nevertheless be viewed as generating representations for attribute classification. Instead of the classical training techniques of GANs (Goodfellow et al., 2014; Arjovsky et al., 2017), we introduce a optimization method – gradient reversal technique, which is oriented in domain adaptation (Ganin & Lempitsky, 2015), into our attribute learning problem. We further empirically validate that gradient reversal is better than GAN-oriented optimizations for attribute learning.

Domain adaptation Domain adaptation tries to develop classifiers that are robust to data outside the domain of the training set. For example, domain adaptation aim for models that can classify outdoor photos even if only trained on indoor ones. While there is much work in this area, recent papers assume more realistic scenarios such as semi- or un-supervised domain adaptation with little or no available annotations in the test domain (Hosseini-Asl et al., 2019; Sohn et al., 2019), open-set adaptation where the test domain is not known at training time (Baktashmotlagh et al., 2019), adaptation for multiple domains (Schoenauer-Sebag et al., 2019), or incorporating distribution shift of labels (in addition to data) (Azizzadenesheli et al., 2019). Since the key challenge among these problems is how to learn robust domain-invariant representations, the approaches are often trained in a adversarial manner similar to GANs. While domain classifiers have a role similar to our auxiliary negative-transfer attribute classifiers, so that we adapt the same adversarial learning technique (Ganin & Lempitsky, 2015), our work is distinct from domain adaptation because we use identical domains in both training and test.

3 Adversarial Attribute Learning (AAL)

3.1 Preliminary: Define Negative Transfer

Suppose we are given a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \{y_{ia}, y_{ib}\})\}_{i=1}^N$ where $\boldsymbol{x}_i$ is i-th image, and y_{ia}, y_{ib} are the binary labels for two attributes of interest, a and b. (For clarity, we restrict our setting here to two attributes; the more general case of M attributes is presented in Appendix A.1.) A neural network to predict attribute a is composed of a feature extraction layer F_a and classification layer C_a. The feature extractor maps the input image $\boldsymbol{x}_i$ into a feature map $\boldsymbol{h}_{ia} = F_a(\boldsymbol{x}_i; \boldsymbol{\theta}_{F_a})$ and a classifier predicts the label probability $\tilde{y}_{ia} = \sigma\left(C_a(\boldsymbol{h}_{ia}; \boldsymbol{\theta}_{C_a})\right)$ where σ is a sigmoid function. Typically, F_a consists of deep convolutional layers and C_a consists of fully connected layers. Similar functions F_b and C_b. are used for attribute b.

We can train predictors with cross-entropy loss $\mathcal{L}\left(\tilde{y}_{ia}, y_{ia}\right) = y_{ia} \log \tilde{y}_{ia} + (1 - y_{ia}) \log(1 - \tilde{y}_{ia})$ for a and b independently (i.e., single task learning),

$$\min_{(\boldsymbol{\theta}_{F_a}, \boldsymbol{\theta}_{C_a})} \sum_{i=1}^{N} \mathcal{L}\left(\tilde{y}_{ia}, y_{ia}\right) \quad \text{and} \quad \min_{(\boldsymbol{\theta}_{F_b}, \boldsymbol{\theta}_{C_b})} \sum_{i=1}^{N} \mathcal{L}\left(\tilde{y}_{ib}, y_{ib}\right). \tag{1}$$

Alternatively, we can share the parameters of the feature extractors by setting $\boldsymbol{\theta}_F = \boldsymbol{\theta}_{F_a} = \boldsymbol{\theta}_{F_b}$, and train two predictors together as a multi-attribute learning problem,

$$\min_{(\boldsymbol{\theta}_F, \boldsymbol{\theta}_{C_a}, \boldsymbol{\theta}_{C_b})} \sum_{i=1}^{N} \left(\mathcal{L}\left(\tilde{y}_{ia}, y_{ia}\right) + \mathcal{L}\left(\tilde{y}_{ib}, y_{ib}\right) \right). \tag{2}$$

Sharing the representation often improves the classification performances for both a and b, but not always: sometimes the independent model for a or b or both actually works better. We call this phenomenon *negative transfer*, and define it as a directed relationship: a negative transfer from b to a exists when the accuracy of a drops when training classifiers for a and b together, and vice versa. Negative transfer is usually seen as a negative outcome to be avoided, but here we try to exploit it. *Given that negative transfer exists between two attributes, can we use it during training to improve classification performance?*

3.2 Adversarial Attribute Learning (AAL) from a negative transfer pair

For clarity, suppose we observe negative transfer from attribute b to a; in other words, multi-task training with a and b yields a model with lower performance for a than the model trained with a single task just for a. This suggests that the representation for attribute b is harmful for predicting a, or, in other words, that a representation that is not tuned for b can better predict attribute a. Using this observation, we propose to train the neural network to learn a representation that is not only predictive of a but also *invariant to* b.

To do this, we propose an adversarial training formluation,

$$\min_{(\boldsymbol{\theta}_F, \boldsymbol{\theta}_{C_a})} \max_{\boldsymbol{\theta}_{C_b}} \sum_{i=1}^{N} \left(\mathcal{L}\left(\tilde{y}_{ia}, y_{ia}\right) + \mathcal{L}\left(\tilde{y}_{ib}, y_{ib}\right) \right), \tag{3}$$

which is exactly the same optimization used in GANs (Goodfellow et al., 2014). The generator creates a representation for a primary classifier for a, and the discriminator is the auxiliary classifier for b that helps the primary classifier learn a better representation for a. Therefore, we can directly adapt GAN optimization, except that we use labels of attribute b instead of the fake/real labels. We reformulate the learning into alternating optimization of two objectives,

$$\min_{(\boldsymbol{\theta}_F, \boldsymbol{\theta}_{C_a})} \sum_{i=1}^{N} \left(\mathcal{L}\left(\tilde{y}_{ia}, y_{ia}\right) + \lambda \mathcal{L}\left(\tilde{y}_{ib}, 1 - y_{ib}\right) \right), \tag{4}$$

$$\min_{\boldsymbol{\theta}_{C_b}} \sum_{i=1}^{N} \mathcal{L}\left(\tilde{y}_{ib}, y_{ib}\right), \tag{5}$$

where λ is a hyper-parameter.

However, GAN optimization is notoriously unstable (Salimans et al., 2016). Although several techniques such as Wasserstein-GAN (Arjovsky et al., 2017) have been proposed, our min-max problem has more technical similarities to a domain adaptation technique (Ganin & Lempitsky, 2015) called gradient reversal. We propose to re-introduce it here for adversarial attribute learning. In this case, we replace equation 4 with:

$$\min_{(\boldsymbol{\theta}_F, \boldsymbol{\theta}_{C_a})} \sum_{i=1}^{N} \left(\mathcal{L}\left(\tilde{y}_{ia}, y_{ia}\right) - \lambda \mathcal{L}\left(\tilde{y}_{ib}, y_{ib}\right) \right), \tag{6}$$

while keeping equation 5 the same.

Remarks. (1) Note that the above equation involves only two attributes, but it can be easily generalized to two groups of attributes, i.e, Group a and b, as discussed in Appendix A.2. (2) We assume that the existence of a pair of attributes exhibiting negative transfer is already known to us; in practice, we can discover it empirically from the training or validation set. (3) Our adversarial attribute learning approach aims at improving attribute a, and C_b is the auxiliary classifier for assisting the primary classifier learn the representation for a. Therefore, C_b is not suitable for classifying b in the end. In order to obtain a classifer for b, it is advisable to use ALL with the a and b inverse, rather than directly utilizing the auxiliary classifier of b. (4) Finally, practices of multi-task learning for shared representations over multi-attributes can also be incorporated into our framework.

3.3 TRAINING FOR MORE THAN TWO ATTRIBUTES

Of course, in practice we will often have more than two attributes in a dataset, and so we must identify and choose among the pairs of attributes having negative transfer. To do this, we try all possible pairs of M attributes, empirically finding the attribute with the worst negative transfer attribute for each attribute. In other word, we try $\binom{M}{2} = \frac{M(M-1)}{2}$ multi-task trainings in addition to M single task trainings, and use the validation accuracy to find the negative transfer pairs. Then, for each attribute, we perform an adversarial training with the most negatively affected attribute. After this, we perform M adversarial trainings and yield a classifier per attribute. In order to better illustrate the overall training, we include a concrete example in Appendix A.3.

An alternative appracoh would be to divide the attributes into groups having the worst mutual negative transfer, and then perform adversarial training. However, this would niavely involve trying all possible 2^{M-1} partitions, is computationally intractible for all but small value of M. We leave exploring this direction for future work.

4 EXPERIMENTS

We tested our techniques on two specific applications: facial attributes and pedestrian attributes.

4.1 EXPERIMENTS ON FACIAL ATTRIBUTE DATASET

We use CelebA (Liu et al., 2015) as our primary dataset for experimentation. This set has 202,599 face images annotated with 40 attributes. We try pairwise multi-task learning to find negative transfer attribute pairs, and then perform adversarial training for each attribute paired with the most negatively affected attribute. Our evaluation metric is the mean accuracy over attributes.

Implementation details We use ResNet18 (He et al., 2016) pretrained on ImageNet as the backbone CNN. When training for multiple attributes, we simply share all convolutional layers and have one fully-connected layer per attributes. In other words, F_a and F_b are ResNet18 without fully-connected layers, and C_a and C_b are linear binary classifiers. We use the stochastic gradient descent algorithm of Adam (Kingma & Ba, 2015) with learning rate 0.0001 and weight decay 0.0005. We train for 6 epochs, dividing the learning rate by a factor of 10 at epoch 4. We manually tune the hyper-parameter λ in equation 6 by trying 0.01, 0.1, 0.3, 0.5, and 1.0. During training, we check the accuracy on the validation set, and then use the best model to compute the final accuracy on the test set. We follow the dataset-provided split of 162,770 training, 19,867 validation, and 19,962 test images.

4.2 RESULTS

Negative transfer Figure 2 (with details in Appendix Table 3) summarizes validation accuracy with single task training and multi-task training with the negative transfer attributes. As an example, Attribute 13 (Bushy Eyebrows) exhibits negative transfer from Attribute 22 (Mouth Slightly Open), as it has a single task accuracy of 92.96%, but multi-task accuracy of 92.80% when trained with Attribute 22. The negative attribute is selected by trying all possible attribute pairs (See Appendix A.4 and Table 4 for more details). Notably, Attribute 7 (Big Lips) is negatively paired with 30 out of 39 other attributes. This suggests that good representations to classify most of the facial attributes should be invariant to the size of lip (e.g., hair color has nothing to do with lips).

Adversarial training Figure 2 also shows the validation accuracy of our adversarial training with the negative transfer attribute. We observe that accuracy is improved from the single task training for 34 out of 40 attributes. For the other six attributes (2, 5, 16, 23, 24, 35, and 38), we do not observe improvement from single task training, but the accuracy is still higher than multi-task training. Overall, our adversarial training achieves 92.46% mean accuracy, which is a 0.13% absolute improvement compared to the 92.33% of single task training.

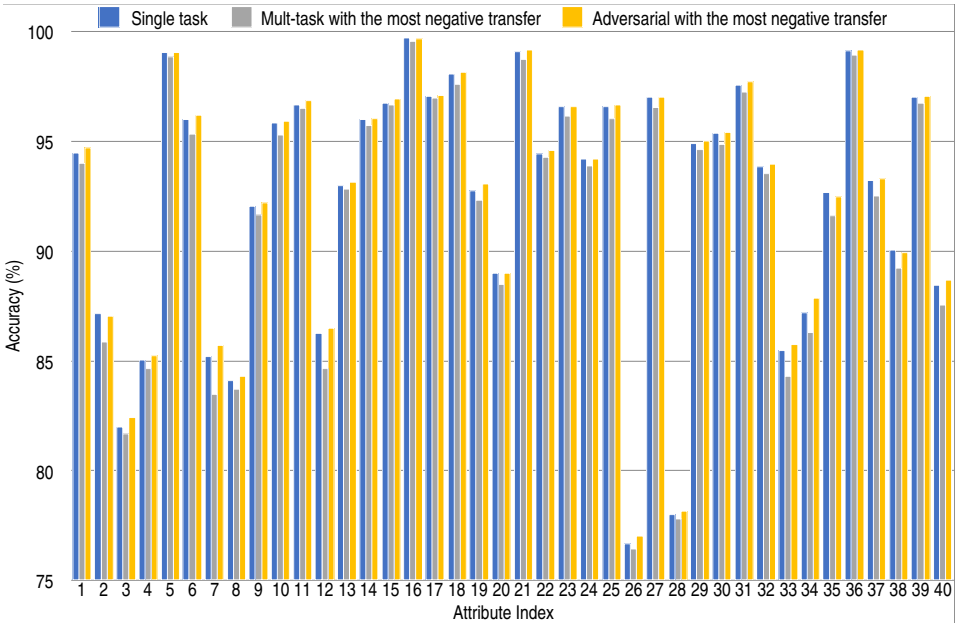

Figure 2: Accuracy (%) on CelebA validation set. *Single task* is the accuracy trained with only a single attribute. *Multi-task with the most negative transfer* is the accuracy trained with the attribute that empirically has the highest accuracy drop from single task training. *Adversarial with the most negative transfer* is trained with the same other attribute but with adversarial learning. Appendix Table 3 shows the original data including attribute names and the most negative attribute.

Table 1: Ablation study with the mean validation accuracy on CelebA.

Single task training per attribute	92.33
Multi task with all attributes	92.34
Pairwise adversarial training	**92.46**
Pairwise adversarial training with GAN loss	92.40
Pairwise adversarial training with Wasserstein-GAN loss	92.41
Pairwise adversarial training with shared CNN representation	92.36
Multiple adversarial training	91.31

4.2.1 Ablation Study

We conduct ablative experiments on the adversarial loss function, CNN representation sharing, and adversarial attributes. We summarize the results in Table 1 and discuss each point below.

GAN based adversarial loss As discussed in Sec. 3.2, we introduced the gradient reversal technique for optimizing min-max equation 3, borrowed from domain adaptation due to its technical similarities of the equation. The results presented above use gradient reversal technique but we also experiment two GAN oriented losses: 1) a original GAN loss and Wasserstein-GAN (WGAN) loss. When we train with the GAN (or WGAN) based loss, the mean validation accuracy falls slightly to 92.40% (or 92.41%), which is 0.06% (or 0.05%) lower than the gradient reversal training.

Sharing representations Our framework creates one classifier per attribute without sharing internal representations at all, and is not the most efficient way to use computational resources. We also experiment with the CNN sharing all convolutional layers, which are pretrained from multi-task learning with all attributes. We then add two more hidden layers that produces the representation for both primary and auxiliary attribute classifiers. The adversarial training only fine-tunes the hidden layers and the classification layers. This training yielded validation accuracy of 92.36%, which is 0.10% worse than the model without sharing representation, but 0.03% better than single task train-

Table 2: Mean attribute classification accuracy (%) on CelebA test set.

Single Task	91.73	Sener & Koltun (2018)	91.75	Hand & Chellappa (2017)	91.26
Multi Task	91.79	He et al. (2017)	91.80	He et al. (2018)	91.81
Ours	**91.94**	Lu et al. (2017)	90.74	Kalayeh et al. (2017)	91.80

ing. This indicates that future work might incorporate more elegant layer sharing strategies from multi-task learning in order to share internal representations.

Multiple adversarial attributes Since we search for negative transfers only in pairwise multi-tasks, our default adversarial training is also pairwise. However, it is possible to select all attributes where negative transfer is observed. For example, Attribute 8 (Big Nose) has the highest accuracy drop when trained with Attribute 7 (Big Lips), but it also has accuracy drops when trained with Attribute 13 (Bushy Eyebrows), 29 (Receding Hairline), or 33 (Straight Hair). In this case, we can train the classifier for Attribute 8 adversarially with 7, 13, and 29. We show these negative attributes in Appendix Table 5. The mean accuracy is 92.31%, which is worse than the pairwise adversarial training. This suggests that it is not easy to use pairwise negative transfer information for discovering a group of attributes that are beneficial for adversarial classification.

4.2.2 Compare with other attribute prediction methods

Finally, we compute the mean accuracy of our method on the test set and compare with other methods. Our method is adversarial training per attribute paired with the most negative transfer attribute based on the validation accuracy. Table 2 compares this with our baselines of single-task training per attribute, and multi-task learning with all attributes. We also show accuracy reported from other methods: multi-objective optimization that approximate a Pareto optima(Sener & Koltun, 2018), multi-task training with adaptive loss weighting (He et al., 2017), adaptive CNN layer sharing (Lu et al., 2017), relationship modeling between attributes (Hand & Chellappa, 2017), and two other methods (He et al., 2018; Kalayeh et al., 2017) utilizing semantic segmentation of facial regions. We note that the comparison is not totally equalized due to some stronger backbones (ResNet50 for He et al. (2017; 2018), and customized architecture for Kalayeh et al. (2017); Hand & Chellappa (2017); Lu et al. (2017) while ResNet18 for ours, our baselines, and Sener & Koltun (2018)) in addition to external facial segmentation data used by He et al. (2018); Kalayeh et al. (2017). Nevertheless, our method achieves the highest accuracy of 91.94%.

Discussion and Future Work The purpose of our experiments is to show that our AAL framework can make use of negative transfer, so we use a brute-force approach to find the negative pairs from all possible pairs. However, this does not scale to learning millions of attributes. Towards the goal of lowering this computation cost, we compute the Pearson correlation coefficient and compare with the relative accuracy change as in (Jayaraman et al., 2014). For example, if the label of an attribute a has correlation of 0.1 with another attribute b, and single-task training of a has accuracy of 80% but multi-task training of a with b gives an accuracy of 79%, the relative accuracy change is $\frac{79-80}{80} = -1.25\%$, and we plot a point (0.1, -1.25). We check the correlation and the relative accuracy change for all $40 \times 39 = 1,560$ pairs and show them in Figure 3. The Pearson correlation coefficient on the scatter plot itself is 0.004, which indicates no correlation. Unfortunately, the connection between label correlation and the negative transfer is still unclear, and thus discovering negative transfer without explicit training is future work.

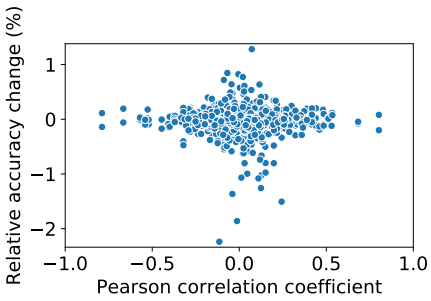

Figure 3: The correlation and the relative accuracy change for all 40 attribute pairs on CelebA validation set . The correlation is computed from labels of an attribute pair, and the relative accuracy change is the relative difference of accuracy when trained with multi-task versus single task learning.

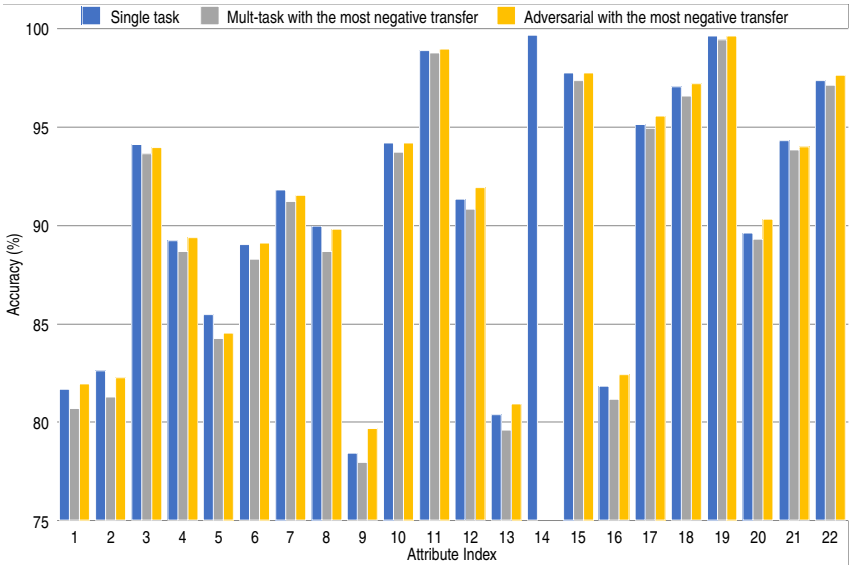

Figure 4: Accuracy (%) on the DukeMTMC-attribute dataset. *Single task* is the accuracy trained only with an attribute. Except for Attribute 14, which does not have any negative transfer pair, multi-task is trained with another attribute whose negative transfer was the worst, and the adversarial one is trained with the same negative attribute but with adversarial learning. Appendix Table 6 shows the original data including attribute names and the most negative attribute.

4.3 Experiments on person attribute dataset

To further validate our technique, we also perform experiments on the person attribute dataset of DukeMTMC-attribute Lin et al. (2019). The dataset defines 23 attributes and contains 16,522 training and 17,661 test pedestrian images. We follow the same protocol as for CelebA dataset, perform pairwise multi-task training, find negative transfers, and adversarially train each attribute classifier with the most negatively-transferred attribute.

Negative transfer Figure 4 (with details in Appendix Table 6) shows the results for training with the most negative transfer attribute. Except Attribute 14 (Green lower-body clothes), we observe negative transfer for every other attribute. Attribute 8 (Long-sleeve upper-body clothes) has four most negative transfer pairs out of 22 other attributes: 10 (White lower-body clothes), 11 (Red lower-body clothes), 15 (Brown lower-body clothes), and 8 (White upper-body clothes). Intuitively, we can interpret that the length of sleeve should not affect the colors of other clothing.

Adversarial training Figure 4 also shows the accuracy of adversarial training with the negative transfer attribute. Of 22 attributes that have negative transfers, 15 attributes see accuracy improvement. For seven attributes (2, 3, 5, 7, 8, 19, and 21), we do not observe improvement compared to single task training, but the accuracies are higher than the corresponding multi-task training. Overall, our adversarial training achieves 92.32% mean accuracy, which is higher than the 92.19% of single task training. Lastly, we also train a multi-task CNN with all attributes, and obtain 91.21% mean accuracy, which is lower than our adversarial training.

5 Conclusion

In this paper, we introduced the idea of utilizing negative relationships for the visual attribute prediction problem. Given a negative transfer between attributes, our AAL framework trains an attribute classification CNN with an auxiliary classifier that predicts the harmful attribute. We adversarially train the two classifiers to allow the CNN to learn a representation agnostic to the harmful attribute. In our experiments, we applied this framework on negative transfer attribute pairs and confirmed an improvement. Effective discovery of the negative pairs without performing every possible training combination remains is future work.

REFERENCES

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

Martin Arjovsky, Soumith Chintala, and Leon Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017.

Kamyar Azizzadenesheli, Anqi Liu, Fanny Yang, and Animashree Anandkumar. Regularized learning for domain adaptation under label shifts. In *International Conference on Learning Representations*, 2019.

Mahsa Baktashmotlagh, Masoud Faraki, Tom Drummond, and Mathieu Salzmann. Learning factorized representations for open-set domain adaptation. In *International Conference on Learning Representations*, 2019.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009.

Yanwei Fu, Tao Xiang, Yu-Gang Jiang, Xiangyang Xue, Leonid Sigal, and Shaogang Gong. Recent advances in zero-shot recognition: Toward data-efficient understanding of visual content. *IEEE Signal Processing Magazine*, 35(1):112–125, 2018.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, 2015.

Yunchao Gong, Yangqing Jia, Thomas Leung, Alexander Toshev, and Sergey Ioffe. Deep convolutional ranking for multilabel image annotation. In *International Conference on Learning Representations*, 2014.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Neural Information Processing Systems*, 2014.

Emily M Hand and Rama Chellappa. Attributes for improved attributes: A multi-task network utilizing implicit and explicit relationships for facial attribute classification. In *AAAI Conference on Artificial Intelligence*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Keke He, Zhanxiong Wang, Yanwei Fu, Rui Feng, Yu-Gang Jiang, and Xiangyang Xue. Adaptively weighted multi-task deep network for person attribute classification. In *ACM International Conference on Multimedia*, 2017.

Keke He, Yanwei Fu, Wuhao Zhang, Chengjie Wang, Yu-Gang Jiang, Feiyue Huang, and Xiangyang Xue. Harnessing synthesized abstraction images to improve facial attribute recognition. In *International Joint Conference on Artificial Intelligence*, 2018.

Ehsan Hosseini-Asl, Yingbo Zhou, Caiming Xiong, and Richard Socher. Augmented cyclic adversarial learning for low resource domain adaptation. In *International Conference on Learning Representations*, 2019.

Guosheng Hu, Yang Hua, Yang Yuan, Zhihong Zhang, Zheng Lu, Sankha S Mukherjee, Timothy M Hospedales, Neil M Robertson, and Yongxin Yang. Attribute-enhanced face recognition with neural tensor fusion networks. In *IEEE International Conference on Computer Vision*, 2017.

Dinesh Jayaraman, Fei Sha, and Kristen Grauman. Decorrelating semantic visual attributes by resisting the urge to share. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

Luo Jiang, Juyong Zhang, and Bailin Deng. Robust rgb-d face recognition using attribute-aware loss. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

Mark Johnson and Kristen Grauman. Relative attributes. In *IEEE International Conference on Computer Vision*, 2011.

Mahdi M. Kalayeh, Boqing Gong, and Mubarak Shah. Improving facial attribute prediction using semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Neeraj Kumar, Alexander Berg, Peter N Belhumeur, and Shree Nayar. Describable visual attributes for face verification and image search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):1962–1977, 2011.

Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2013.

Hae Beom Lee, Eunho Yang, and Sung Ju Hwang. Deep asymmetric multi-task feature learning. In *International Conference on Machine Learning*, 2018.

Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, Zhilan Hu, Chenggang Yan, and Yi Yang. Improving person re-identification by attribute and identity learning. *Pattern Recognition*, 2019. doi: https://doi.org/10.1016/j.patcog.2019.06.006.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *IEEE International Conference on Computer Vision*, 2015.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and S Yu Philip. Learning multiple tasks with multilinear relationship networks. In *Neural Information Processing Systems*, 2017.

Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogerio Feris. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Elliot Meyerson and Risto Miikkulainen. Beyond shared hierarchies: Deep multitask learning through soft layer ordering. In *International Conference on Learning Representations*, 2018.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Neural Information Processing Systems*, pp. 2234–2242, 2016.

Walter J. Scheirer, Neeraj Kumar, Peter N. Belhumeur, and Terrance E. Boult. Multi-attribute spaces: Calibration for attribute fusion and similarity search. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

Alice Schoenauer-Sebag, Louise Heinrich, Marc Schoenauer, Michele Sebag, Lani Wu, and Steve Altschuler. Multi-domain adversarial learning. In *International Conference on Learning Representations*, 2019.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Neural Information Processing Systems*, 2018.

Behjat Siddiquie, Rogerio S Feris, and Larry S Davis. Image ranking and retrieval based on multi-attribute queries. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

Kihyuk Sohn, Wenling Shang, Xiang Yu, and Manmohan Chandraker. Unsupervised domain adaptation for distance metric learning. In *International Conference on Learning Representations*, 2019.

Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *Int J Data Warehousing and Mining*, 2007:1–13, 2007.

Yongxin Yang and Timothy M Hospedales. Trace norm regularised deep multi-task learning. *arXiv preprint arXiv:1606.04038*, 2016.

Yongxin Yang and Timothy M Hospedales. Deep multi-task representation learning: A tensor factorisation approach. In *International Conference on Learning Representations*, 2017.

A Appendix

A.1 Problem Setup with more than two attributes

A dataset with M attributes and N images is denoted as $\mathcal{D} = \left\{ \left(\boldsymbol{x}_i, \{y_{ij}\}_{j=1}^{M} \right) \right\}_{i=1}^{N}$ where $\boldsymbol{x}_i$ is i-th image, and $y_{ij} \in \{0, 1\}$ is a label for attribute j of the image.

A neural network to predict attribute j has a feature extraction layer F_j and classification layer C_j. The feature extractor maps the input image $\boldsymbol{x}_i$ into a feature map $\boldsymbol{h}_{ij} = F_j(\boldsymbol{x}_i; \boldsymbol{\theta}_{F_j})$ and a classifier for attribute j gives the label probability $\tilde{y}_{ij} = \sigma\left(C_j(\boldsymbol{h}_{ij}; \boldsymbol{\theta}_{C_j})\right)$ where σ is a sigmoid function.

We can train a predictor for each attribute j with cross-entropy loss $\mathcal{L}\left(\tilde{y}_{ij}, y_{ij}\right) = y_{ij} \log \tilde{y}_{ij} + (1 - y_{ij}) \log(1 - \tilde{y}_{ij})$.

$$\min_{(\boldsymbol{\theta}_{F_j}, \boldsymbol{\theta}_{C_j})} \sum_{i=1}^{N} \mathcal{L}\left(\tilde{y}_{ij}, y_{ij}\right) \tag{7}$$

We also train the model sharing the feature extractors as $\boldsymbol{\theta}_F = \boldsymbol{\theta}_{F_1} = \ldots = \boldsymbol{\theta}_{F_M}$ and constuct a multi-task learning problem as follows.

$$\min_{(\boldsymbol{\theta}_F, \{\boldsymbol{\theta}_{C_j}\}_{j=1}^{M})} \sum_{i=1}^{N} \sum_{j=1}^{M} \mathcal{L}\left(\tilde{y}_{ij}, y_{ij}\right) \tag{8}$$

The multi-task training tends to improve the overall classification performance but the performance for some attributes often drops, which is called negative transfer. Our focus in this paper is, how to make use of the negative task relationship for improving the classification performance.

A.2 Adversarial Training for Two Sets of Attributes with Negative Transfer

The adversarial training described in Sec. 3.2 can be easily extended into two groups of attributes with negative transfer. Let us assume that we have M attributes and divided into a group $a = \{a_k\}_{k=1}^{A}$ and $b = \{b_k\}_{k=1}^{B}$ where $M = A+B$. We also assume a negative transfer from b to a, which means that the multi-task training with all attributes in a and b gives lower average performance for attributes in a than another multi-task training only with attributes in a. The adversarial training can be denoted as follows.

$$\min_{(\boldsymbol{\theta}_F, \boldsymbol{\theta}_{C_{a_1}}, \ldots, \boldsymbol{\theta}_{C_{a_A}})} \max_{\left(\boldsymbol{\theta}_{C_{b_1}}, \ldots, \boldsymbol{\theta}_{C_{b_B}}\right)} \sum_{i=1}^{N} \left(\frac{1}{A} \sum_{j \in a} \mathcal{L}\left(\tilde{y}_{ij}, y_{ij}\right) + \frac{1}{B} \sum_{j \in b} \mathcal{L}\left(\tilde{y}_{ij}, y_{ij}\right) \right) \tag{9}$$

The reformulated optimization with the gradient reversal (Ganin & Lempitsky, 2015) technique is the following.

$$\min_{(\boldsymbol{\theta}_F, \boldsymbol{\theta}_{C_{a_1}}, \ldots, \boldsymbol{\theta}_{C_{a_A}})} \sum_{i=1}^{N} \left(\frac{1}{A} \sum_{j \in a} \mathcal{L}\left(\tilde{y}_{ij}, y_{ij}\right) - \lambda \frac{1}{B} \sum_{j \in b} \mathcal{L}\left(\tilde{y}_{ij}, y_{ij}\right) \right) \tag{10}$$

$$\min_{\left(\boldsymbol{\theta}_{C_{b_1}},\ldots,\boldsymbol{\theta}_{C_{b_B}}\right)} \sum_{i=1}^{N} \frac{1}{B} \sum_{j \in b} \mathcal{L}\left(\tilde{y}_{ij}, y_{ij}\right) \tag{11}$$

where λ is a hyper parameter to balance out the loss between the two groups of attributes.

A.3 An example of overall training procedure given a dataset

Let us assume we have a dataset of three attributes $A1$, $A2$, and $A3$. We will perform three single task trainings and also all six possible multi-task trainings. Then, we get the following results (the accuracies are created only for illustration purpose).

- The accuracy for $A1$ is 90% with single task training, 80% when trained with $A2$, and 90% when trained with $A3$
- The accuracy for $A2$ is 70% with single task training, 80% when trained with $A1$, and 60% when trained with $A3$
- The accuracy for $A3$ is 80% with single task training, 70% when trained with $A1$, and 75% when trained with $A2$

Then, we perform the following three adversarial trainings and obtain a classifier per attribute.

- $A1$ is the most negatively paired with $A2$ so train $A1$ classifier adversarially with $A2$.
- $A2$ is the most negatively paired with $A3$ so train $A2$ classifier adversarially with $A3$.
- $A3$ is the most negatively paired with $A1$ so train $A3$ classifier adversarially with $A1$.

A.4 Pairwise Multi-task Accuracy

We give the pairwise multi-task accuracy on CelebA in Table. 4. As mentioned in Sec. 3.3, we train with all possible attribute pairs to find the most negative transfer for each attribute. The row direction is the main attribute of the accuracy that we care, and the column is another attribute trained with the main one. The diagonal elements are filled with the accuracy from the single task training. For example, the value in the (row,column) = (10,10) is 95.83%, which is the accuracy of the single task training only with attribute 10 (Blond Hair). In the same row but in the 9-th column ((row,column) = (10,9)), the value is 95.86, which means the accuracy of attribute 10 is 95.86% (0.03% increase) when trained with attribute 9. On the other hand, the same row with 11-th column ((row,column) = (10,11)) has the value of 95.79%, which means the accuracy of attribute 10 drops 0.04% when trained with attribute 11. Table 3 discussed in Sec. 4 is constructed from Table 4. Table 3 shows that the most negative attribute for attribute 10 is attribute 7 with the multi-task accuracy of 95.29%. This is the lowest accuracy of the row 10 in Table 4, corresponding to the (row,column) = (10,7).

Table 3: Accuracy (%) on CelebA validation set. *Neg Att* is the attribute (Att) that has the highest accuracy drop when trained with multi-task learning. *Sin* is the accuracy from single task training. The *Mul* is the accuracy from multi-task training with the *Neg Att Adv* is the accuracy from adversarial training with the *Neg Att*.

.

Att	Att Name	Neg Att	Sin	Mul	Adv	Mul - Sin	Adv - Sin	Adv - Mul
1	5_o_Clock_Shadow	7	94.47	93.97	94.71	-0.50	0.24	0.74
2	Arched_Eyebrows	7	87.15	85.87	87.02	-1.28	-0.13	1.15
3	Attractive	7	81.99	81.68	82.42	-0.31	0.43	0.74
4	Bags_Under_Eyes	2	85.02	84.66	85.24	-0.36	0.22	0.58
5	Bald	24	99.05	98.84	99.05	-0.21	0.00	0.21
6	Bangs	7	96.00	95.31	96.17	-0.69	0.17	0.86
7	Big_Lips	26	85.20	83.48	85.70	-1.72	0.50	2.22
8	Big_Nose	7	84.10	83.72	84.32	-0.38	0.22	0.60
9	Black_Hair	7	92.02	91.63	92.19	-0.39	0.17	0.56
10	Blond_Hair	7	95.83	95.29	95.91	-0.54	0.08	0.62
11	Blurry	7	96.65	96.48	96.86	-0.17	0.21	0.38
12	Brown_Hair	7	86.24	84.65	86.50	-1.59	0.26	1.85
13	Bushy_Eyebrows	22	92.96	92.80	93.15	-0.16	0.19	0.35
14	Chubby	7	95.99	95.72	96.01	-0.27	0.02	0.29
15	Double_Chin	7	96.75	96.67	96.93	-0.08	0.18	0.26
16	Eyeglasses	7	99.69	99.55	99.66	-0.14	-0.03	0.11
17	Goatee	7	97.03	96.96	97.08	-0.07	0.05	0.12
18	Gray_Hair	7	98.07	97.60	98.13	-0.47	0.06	0.53
19	Heavy_Makeup	7	92.74	92.31	93.05	-0.43	0.31	0.74
20	High_Cheekbones	7	88.97	88.48	89.00	-0.49	0.03	0.52
21	Male	34	99.09	98.74	99.13	-0.35	0.04	0.39
22	Mouth_Slightly_Open	7	94.43	94.28	94.56	-0.15	0.13	0.28
23	Mustache	7	96.56	96.14	96.56	-0.42	0.00	0.42
24	Narrow_Eyes	4	94.19	93.89	94.18	-0.30	-0.01	0.29
25	No_Beard	7	96.57	96.04	96.64	-0.53	0.07	0.60
26	Oval_Face	9	76.66	76.45	77.02	-0.21	0.36	0.57
27	Pale_Skin	34	96.98	96.52	97.00	-0.46	0.02	0.48
28	Pointy_Nose	38	78.02	77.80	78.16	-0.22	0.14	0.36
29	Receding_Hairline	26	94.91	94.63	95.00	-0.28	0.09	0.37
30	Rosy_Cheeks	7	95.37	94.85	95.41	-0.52	0.04	0.56
31	Sideburns	7	97.56	97.25	97.69	-0.31	0.13	0.44
32	Smiling	7	93.85	93.52	93.96	-0.33	0.11	0.44
33	Straight_Hair	7	85.47	84.31	85.76	-1.16	0.29	1.45
34	Wavy_Hair	7	87.17	86.30	87.84	-0.87	0.67	1.54
35	Wearing_Earrings	7	92.66	91.59	92.47	-1.07	-0.19	0.88
36	Wearing_Hat	7	99.12	98.90	99.16	-0.22	0.04	0.26
37	Wearing_Lipstick	7	93.19	92.51	93.28	-0.68	0.09	0.77
38	Wearing_Necklace	7	90.03	89.20	89.91	-0.83	-0.12	0.71
39	Wearing_Necktie	7	97.02	96.73	97.04	-0.29	0.02	0.31
40	Young	7	88.46	87.54	88.66	-0.92	0.20	1.12
	Mean		92.33	91.82	92.46	-0.51	0.13	0.64

Table 4: Pairwise multi-task accuracy on CelebA validation set. Best viewed with zoom and rotation.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	94.47	94.47	94.55	94.49	94.63	94.58	93.97	94.64	94.44	94.50	94.60	94.48	94.44	94.44	94.62	94.52	94.54	94.48	94.46	94.59	94.54	94.53	94.64	94.53	94.64	94.61	94.43	94.43	94.51	94.44	94.51	94.50	94.49	94.56	94.53	94.52	94.43	94.47	94.53	94.54
2	86.91	87.15	87.09	87.03	87.14	87.19	85.87	87.08	87.01	87.09	87.12	86.95	87.06	86.96	87.12	87.10	87.01	87.03	87.03	86.98	87.00	87.04	86.91	86.94	87.01	87.57	86.96	86.96	86.96	87.00	87.04	86.83	87.11	86.97	87.10	87.01	86.98	86.98	87.04	86.96
3	81.94	82.09	81.99	82.32	82.11	82.24	81.68	82.16	82.15	81.99	82.00	82.00	82.11	81.91	82.13	82.14	82.29	82.20	82.15	82.18	82.03	82.10	82.15	82.01	82.25	82.18	81.82	81.97	82.15	82.10	82.03	82.22	81.88	82.25	82.08	81.85	82.19	82.16
4	84.92	84.66	84.84	85.02	84.84	85.12	84.85	85.07	84.93	85.04	85.15	84.88	85.16	85.04	85.06	85.07	85.13	84.95	84.82	85.01	84.93	84.95	85.02	85.13	84.99	85.04	85.00	84.90	84.99	84.97	84.88	84.90	85.02	84.96	84.89	84.84	84.80	84.85	85.07	84.78
5	99.01	99.02	98.96	99.01	99.05	98.98	98.86	99.00	99.00	99.04	99.01	99.05	98.98	98.86	99.00	99.04	99.03	99.06	99.05	99.07	99.04	99.06	99.07	98.84	99.04	98.98	98.97	99.01	99.06	99.08	98.99	99.00	98.94	98.94	99.05	99.05	99.07	99.05	99.05	98.97
6	95.84	95.97	95.99	95.94	96.01	96.00	95.31	95.93	96.03	95.91	96.05	96.01	95.91	95.93	95.98	95.92	95.94	95.98	95.95	95.99	95.98	95.94	95.97	95.98	95.90	96.01	95.95	95.87	95.95	95.96	95.89	96.05	96.05	95.96	95.99	96.00	95.91	95.95	95.95	95.95
7	85.80	84.90	85.63	85.40	86.01	85.42	85.20	86.27	84.85	85.28	85.65	84.85	84.80	85.75	85.08	85.68	85.95	85.90	85.57	84.31	85.22	85.15	85.79	85.80	85.64	83.48	85.29	85.42	85.01	84.80	85.55	84.19	84.93	84.62	85.57	85.11	84.75	84.48	86.02	84.93
8	84.24	84.33	84.27	84.15	84.12	84.24	83.72	84.10	84.23	84.15	84.32	84.29	84.07	84.26	84.34	84.27	84.43	84.15	84.27	84.23	84.25	84.33	84.43	84.37	84.22	84.20	84.35	84.39	84.06	84.27	84.31	84.40	84.05	84.18	84.13	84.19	84.19	84.26	84.21	84.27
9	92.24	92.21	92.05	92.08	92.21	92.00	91.63	92.12	92.02	92.14	92.15	92.10	92.13	92.05	92.16	92.13	92.10	92.30	92.15	92.12	92.09	92.11	92.18	92.18	92.09	92.00	92.13	92.11	92.15	91.99	92.21	92.12	92.02	92.16	92.20	92.11	92.02	92.16	92.08
10	95.87	95.88	95.85	95.90	95.78	95.81	95.29	95.94	95.86	95.83	95.79	95.81	95.76	95.85	95.77	95.74	95.79	95.73	95.82	95.78	95.80	95.68	95.79	95.83	95.87	95.89	95.84	95.70	95.81	95.71	95.81	95.80	95.81	95.71	95.83	95.87	95.67	95.74	95.79	95.78
11	96.70	96.69	96.68	96.67	96.61	96.62	96.48	96.64	96.65	96.72	96.65	96.66	96.69	96.69	96.62	96.71	96.65	96.70	96.71	96.71	96.70	96.80	96.72	96.68	96.67	96.66	96.58	96.73	96.73	96.67	96.66	96.76	96.67	96.68	96.79	96.71	96.67	96.72	96.69	96.74
12	86.15	86.15	86.08	86.06	86.15	86.17	84.65	86.25	86.19	86.24	86.20	86.24	86.23	86.22	86.16	86.23	86.02	86.18	86.00	86.15	86.18	86.11	86.14	86.05	85.98	86.33	86.24	86.20	86.18	86.16	86.05	86.31	86.07	86.01	86.27	86.12	86.05	86.19	86.08
13	93.11	93.06	92.92	92.96	93.02	93.00	92.90	93.08	93.12	93.04	93.12	93.03	92.96	92.90	93.00	92.99	92.86	93.13	92.99	93.04	92.98	92.80	92.94	92.98	93.03	93.01	93.10	92.95	92.95	93.02	93.07	92.89	93.04	92.91	93.04	93.05	93.09	93.03	93.13	93.10
14	95.86	95.96	95.96	95.99	95.83	95.90	95.72	95.89	95.75	95.97	95.86	95.84	96.04	95.99	95.87	95.93	95.83	95.91	95.93	95.98	96.01	95.78	95.96	95.87	95.84	96.00	95.95	95.87	95.91	95.95	95.91	95.93	95.91	95.95	95.91	95.96	95.90	96.00	95.89	95.84
15	96.80	96.75	96.79	96.87	96.74	96.90	96.67	96.83	96.74	96.88	96.78	96.82	96.86	96.84	96.75	96.85	96.73	96.87	96.82	96.92	96.87	96.88	96.76	96.84	96.83	96.93	96.85	96.90	96.87	96.85	96.80	96.82	96.83	96.70	96.76	96.83	96.92	96.71	96.79	96.82
16	99.65	99.64	99.67	99.65	99.68	99.65	99.55	99.65	99.67	99.65	99.67	99.62	99.69	99.67	99.65	99.69	99.67	99.65	99.69	99.64	99.66	99.65	99.66	99.67	99.68	99.68	99.68	99.67	99.63	99.66	99.65	99.65	99.67	99.61	99.67	99.67	99.68	99.67	99.69	99.65
17	97.02	97.08	97.07	97.08	97.03	96.97	97.03	97.08	97.08	97.07	97.02	97.03	97.09	96.97	96.99	97.01	97.03	96.96	97.07	97.09	97.00	97.02	97.05	97.03	97.02	97.03	97.13	96.98	97.12	97.05	97.05	97.04	97.05	97.03	96.99	97.03	97.05	97.07	97.01	97.05
18	98.08	98.13	98.12	98.06	98.05	98.11	97.60	98.03	98.16	98.16	98.15	98.08	98.16	98.06	98.06	98.06	98.07	98.07	98.11	98.04	98.09	98.05	98.11	98.05	98.06	98.07	98.04	98.08	98.05	98.10	98.08	98.05	98.01	98.14	98.04	98.04	98.18	98.09	98.08	98.13
19	92.80	92.91	92.66	92.78	92.69	92.81	92.31	92.65	92.84	92.60	92.78	92.77	92.75	92.72	92.84	92.80	92.83	92.66	92.74	92.90	92.93	92.89	92.78	92.73	92.76	92.82	92.72	92.79	92.75	92.82	92.72	92.86	92.77	92.87	92.80	92.78	92.83	92.76	92.78	92.69
20	88.93	88.86	88.97	88.98	88.91	88.98	88.48	88.86	88.86	88.86	88.99	88.91	89.03	89.10	88.94	88.87	88.89	88.99	88.93	88.97	88.86	88.95	88.77	88.70	88.96	88.79	88.90	88.86	88.91	88.92	88.75	88.88	88.90	88.90	88.84	88.83	88.90	88.81	88.77	88.71
21	99.00	99.01	98.98	99.01	99.09	99.03	98.77	99.01	99.03	99.07	99.00	99.03	98.98	99.03	98.96	99.08	98.99	99.09	99.03	99.05	99.09	99.03	99.08	98.94	99.00	98.99	99.03	98.90	98.99	99.03	99.06	99.02	99.10	98.74	99.10	99.07	98.96	99.03	99.09	99.02
22	94.46	94.47	94.40	94.57	94.45	94.46	94.28	94.51	94.52	94.45	94.48	94.42	94.45	94.46	94.52	94.47	94.44	94.35	94.42	94.60	94.45	94.43	94.41	94.52	94.40	94.50	94.53	94.47	94.48	94.55	94.53	94.53	94.33	94.31	94.37	94.50	94.56	94.43	94.54	94.43
23	96.53	96.63	96.49	96.45	96.53	96.44	96.14	96.38	96.47	96.52	96.53	96.49	96.46	96.53	96.48	96.49	96.45	96.53	96.43	96.46	96.40	96.43	96.56	96.55	96.40	96.61	96.39	96.52	96.53	96.41	96.47	96.42	96.45	96.31	96.46	96.42	96.47	96.59	96.43	96.44
24	94.18	93.91	94.00	93.89	94.36	94.15	94.02	94.11	93.89	94.20	94.26	94.02	94.14	94.14	93.95	94.06	94.22	94.15	94.26	94.29	94.08	93.97	94.17	94.19	94.10	93.93	94.35	94.16	93.99	94.22	94.32	94.00	94.01	94.23	94.05	94.18	94.20	94.00	94.17	94.11
25	96.58	96.50	96.57	96.53	96.51	96.58	96.04	96.42	96.46	96.46	96.53	96.45	96.54	96.48	96.44	96.50	96.60	96.62	96.58	96.55	96.56	96.49	96.56	96.47	96.57	96.57	96.48	96.53	96.53	96.58	96.47	96.46	96.52	96.54	96.40	96.52	96.52	96.46	96.62	96.37
26	76.87	76.84	76.69	76.63	76.74	76.89	76.61	76.59	76.45	76.88	76.62	76.70	76.93	76.76	76.81	76.65	76.74	76.69	76.76	76.83	76.80	76.56	76.76	76.76	76.67	76.66	76.68	76.85	76.76	76.96	76.75	76.74	76.65	76.65	76.72	76.86	76.55	77.03	76.88	76.73
27	96.98	96.91	96.89	96.77	97.00	96.95	96.58	96.92	97.01	96.88	96.86	96.92	96.89	96.89	96.97	96.91	96.87	96.89	96.84	96.89	97.00	96.76	96.91	96.80	96.91	96.93	96.98	96.94	96.87	96.94	96.93	96.97	96.99	96.52	96.95	97.04	96.90	96.88	97.01	96.86
28	78.02	78.04	77.94	77.88	78.27	78.14	77.92	78.13	78.14	78.04	78.18	78.14	78.08	78.22	78.02	78.24	78.08	78.19	77.86	78.04	77.94	78.06	78.24	78.21	77.95	78.05	78.26	78.02	78.11	77.98	78.02	77.93	77.90	78.17	77.88	78.26	77.98	77.80	78.17	78.19
29	94.81	94.85	94.93	94.87	94.93	94.84	94.77	94.83	94.73	94.85	94.81	94.67	94.84	94.72	94.77	94.83	94.80	94.91	94.91	94.77	94.86	94.80	94.73	94.85	94.87	94.63	94.83	94.90	94.91	94.82	94.84	94.80	94.84	94.82	94.90	94.95	94.85	94.83	94.81	94.89
30	95.32	95.25	95.25	95.29	95.32	95.36	94.85	95.32	95.27	95.36	95.34	95.28	95.35	95.33	95.42	95.30	95.29	95.38	95.31	95.32	95.29	95.34	95.29	95.37	95.31	95.23	95.28	95.21	95.34	95.37	95.23	95.33	95.20	95.32	95.18	95.33	95.27	95.31	95.28	95.23
31	97.62	97.66	97.59	97.61	97.53	97.60	97.25	97.51	97.63	97.61	97.58	97.55	97.63	97.58	97.57	97.64	97.57	97.55	97.62	97.48	97.52	97.46	97.58	97.39	97.55	97.39	97.49	97.49	97.44	97.59	97.56	97.55	97.49	97.50	97.49	97.56	97.59	97.51	97.63	97.51
32	93.68	93.88	93.85	93.79	93.81	93.83	93.52	93.78	93.74	93.89	93.84	93.82	93.81	93.78	93.78	93.93	93.80	93.76	93.75	93.81	93.80	93.92	93.78	93.79	93.79	93.82	93.76	93.77	93.90	93.75	93.70	93.85	93.71	93.68	93.88	93.92	93.86	93.79	93.76	93.83
33	85.60	85.47	85.77	85.67	85.72	85.74	84.31	85.50	85.46	85.44	85.48	85.48	85.53	85.56	85.69	85.50	85.59	85.55	85.60	85.48	85.44	85.63	85.76	85.64	85.47	85.55	85.61	85.42	85.50	85.62	85.70	85.64	85.47	85.06	85.57	85.62	85.64	85.64	85.56	85.62
34	87.05	86.96	86.97	86.84	87.46	86.79	86.30	86.95	86.88	87.11	86.77	87.12	87.25	86.92	87.06	87.10	86.96	86.67	86.90	87.21	87.38	87.19	87.52	87.65	86.93	87.01	87.14	86.85	86.98	86.83	86.70	87.10	87.29	87.17	86.49	86.74	86.91	86.88	87.18	86.96
35	92.61	92.57	92.55	92.57	92.65	92.64	91.59	92.59	92.59	92.62	92.59	92.57	92.53	92.52	92.67	92.64	92.63	92.63	92.60	92.55	92.54	92.55	92.54	92.69	92.54	92.64	92.60	92.48	92.63	92.69	92.54	92.40	92.59	92.69	92.66	92.62	92.62	92.74	92.54	92.51
36	99.11	99.08	99.07	99.12	99.10	99.16	98.90	99.15	99.14	99.12	99.05	99.14	99.07	99.14	99.10	99.11	99.16	99.09	99.03	99.12	99.13	99.13	99.12	99.11	99.12	99.09	99.06	99.05	99.11	99.07	99.10	99.11	99.11	99.08	99.12	99.12	99.10	99.12	99.13	99.11
37	93.22	93.08	93.10	93.26	93.19	93.19	92.51	93.10	93.24	92.97	93.33	93.11	93.16	93.11	93.20	93.15	93.18	93.16	93.01	93.19	93.31	93.06	93.29	93.15	93.28	93.15	93.30	93.20	93.13	93.16	93.14	93.14	93.21	93.12	93.05	93.18	93.19	93.30	93.11
38	89.90	89.88	89.87	89.87	89.96	89.93	89.20	90.00	89.82	89.93	89.94	89.72	89.98	89.91	89.99	89.80	89.85	89.92	89.96	90.05	90.02	89.99	90.01	89.84	89.97	89.74	89.90	89.79	89.97	90.11	90.05	89.89	89.86	89.95	89.90	89.93	90.09	90.03	89.96	90.00
39	96.98	96.99	97.07	96.85	97.02	97.05	96.73	97.01	96.96	97.00	97.00	97.07	97.05	96.92	96.96	97.11	97.03	97.04	97.08	97.03	97.05	96.94	97.05	97.05	97.01	96.96	97.04	97.00	96.96	97.04	97.02	96.99	97.02	96.87	97.00	96.96	97.05	96.98	97.02	97.03
40	88.52	88.59	88.45	88.37	88.37	88.39	87.54	88.29	88.33	88.47	88.55	88.45	88.27	88.46	88.46	88.19	88.20	88.41	88.45	88.34	88.47	88.37	88.35	88.48	88.38	88.37	88.31	88.45	88.51	88.42	88.18	88.41	88.33	88.21	88.36	88.48	88.46	88.66	88.42	88.46

Table 5: Accuracy (%) on CelebA validation set with all adversarial attributes.

Att	Acc.	Adversarial trained attributes (Atts)
1	94.65	7, 9, 13, 14, 19, 27, 28, 30, 37
2	86.90	1, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40
3	82.19	1, 7, 14, 28, 29, 35, 38
4	85.19	1, 2, 3, 5, 7, 9, 12, 18, 19, 20, 21, 22, 25, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 40
5	99.02	1, 2, 3, 4, 6, 7, 8, 9, 10, 12, 14, 15, 16, 17, 21, 24, 25, 26, 27, 28, 31, 32, 33, 34, 38, 40
6	95.96	1, 2, 3, 4, 7, 8, 10, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 27, 28, 29, 30, 31, 34, 35, 37, 38, 39, 40
7	83.56	2, 9, 12, 13, 15, 20, 22, 26, 29, 30, 32, 33, 34, 36, 37, 38, 40
8	84.14	7, 13, 29, 33
9	91.87	6, 7, 26, 30
10	95.97	5, 6, 7, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 28, 29, 30, 31, 32, 33, 34, 37, 38, 39, 40
11	96.69	5, 6, 7, 8, 15, 27
12	85.90	1, 2, 3, 4, 5, 6, 7, 9, 11, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 29, 30, 31, 32, 34, 35, 37, 38, 39, 40
13	93.09	3, 7, 14, 17, 22, 23, 28, 29, 32, 34
14	95.99	1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 15, 16, 17, 18, 19, 20, 22, 23, 24, 25, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 39, 40
15	96.90	5, 7, 9, 17, 34, 38
16	99.62	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 17, 18, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 40
17	97.08	1, 7, 11, 14, 15, 16, 18, 21, 22, 25, 28, 35, 39
18	98.08	4, 5, 7, 8, 14, 15, 16, 20, 22, 24, 25, 27, 32, 33, 35, 36
19	93.05	3, 5, 7, 8, 10, 14, 18, 24, 27, 31, 40
20	89.14	1, 2, 3, 5, 7, 8, 9, 10, 12, 15, 16, 17, 19, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40
21	99.11	1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 34, 36, 37, 38, 40
22	94.43	3, 7, 12, 18, 19, 23, 25, 33, 34, 35
23	96.57	1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24, 25, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 39, 40
24	93.85	1, 2, 3, 4, 6, 7, 8, 9, 12, 13, 14, 15, 16, 18, 21, 22, 23, 25, 26, 28, 29, 32, 33, 35, 36, 38, 39, 40
25	96.59	2, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 20, 21, 22, 23, 24, 27, 28, 29, 31, 32, 33, 34, 35, 36, 37, 38, 40
26	76.89	4, 7, 8, 9, 11, 16, 22, 33, 34, 37
27	96.95	2, 3, 4, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 22, 23, 24, 25, 26, 28, 29, 30, 31, 32, 34, 35, 37, 38, 40
28	77.95	3, 4, 7, 19, 21, 25, 30, 32, 33, 35, 37, 38
29	94.75	1, 2, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 20, 21, 22, 23, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34, 35, 37, 38, 39, 40
30	95.34	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40
31	97.66	5, 7, 8, 12, 18, 20, 21, 22, 24, 25, 26, 27, 28, 29, 32, 33, 34, 35, 36, 38, 40
32	93.83	1, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 17, 18, 19, 20, 21, 23, 24, 25, 26, 27, 28, 30, 31, 33, 34, 38, 39, 40
33	85.78	7, 9, 10, 21, 28, 34
34	87.38	1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16, 17, 18, 19, 25, 26, 27, 28, 29, 30, 31, 32, 35, 36, 37, 38, 40
35	92.47	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 26, 27, 28, 29, 31, 32, 33, 36, 37, 39, 40
36	99.13	1, 2, 3, 5, 7, 11, 13, 15, 16, 18, 19, 24, 26, 27, 28, 29, 30, 31, 32, 33, 34, 37, 40
37	93.23	2, 3, 7, 8, 10, 12, 13, 14, 16, 17, 18, 19, 20, 22, 24, 26, 29, 30, 31, 32, 34, 35, 36, 38, 40
38	89.95	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 27, 28, 29, 32, 33, 34, 35, 36, 39, 40
39	96.89	1, 2, 4, 5, 7, 8, 9, 10, 11, 14, 15, 22, 25, 26, 28, 29, 32, 34, 35, 36, 38
40	88.44	3, 4, 5, 6, 7, 8, 9, 12, 13, 16, 17, 18, 19, 20, 22, 23, 25, 26, 27, 28, 30, 31, 32, 33, 34, 35, 39
Mean	92.31	

Table 6: Accuracy (%) on DukeMTMC-attribute dataset. Neg Att is the attribute (Att) that has the highest accuracy drop when trained with multi-task learning. Sin. is the accuracy from single task training. The Mul is the accuracy from multi-task training with the Neg Att. Adv is the accuracy from adversarial training with the Neg Att. The attribute 14 does not have negative transfer with any other attribute so we show the accuracy from the single task training.

Att	Att name	Neg Att	Sin	Mul	Adv	Mul-Sin	Adv-Sin	Adv-Mul
1	Carrying backpack	13	81.67	80.71	81.95	-0.96	0.28	1.24
2	Carrying bag	9	82.63	81.28	82.27	-1.36	-0.36	0.99
3	Carrying handbag	11	94.12	93.63	93.96	-0.49	-0.15	0.33
4	Wearing boots	16	89.21	88.68	89.38	-0.54	0.17	0.7
5	Gender	9	85.48	84.28	84.54	-1.21	-0.95	0.26
6	Wearing hat	13	89.03	88.27	89.1	-0.76	0.07	0.83
7	Light-color shoes	6	91.81	91.23	91.52	-0.58	-0.29	0.29
8	Long-sleeve upper-body clothes	2	89.98	88.69	89.8	-1.29	-0.19	1.11
9	Black lower-body clothes	1	78.42	77.97	79.7	-0.44	1.29	1.73
10	White lower-body clothes	8	94.17	93.7	94.2	-0.47	0.02	0.5
11	Red lower-body clothes	8	98.87	98.75	98.95	-0.12	0.08	0.2
12	Gray lower-body clothes	13	91.33	90.81	91.91	-0.52	0.58	1.1
13	Blue lower-body clothes	4	80.37	79.61	80.92	-0.76	0.55	1.31
14	Green lower-body clothes	-	99.64	-	-	-	-	-
15	Brown lower-body clothes	8	97.75	97.34	97.76	-0.4	0.01	0.42
16	Black upper-body clothes	18	81.85	81.18	82.41	-0.67	0.55	1.23
17	White upper-body clothes	8	95.14	94.92	95.57	-0.22	0.43	0.65
18	Red upper-body clothes	7	97.03	96.56	97.19	-0.47	0.16	0.63
19	Purple upper-body clothes	21	99.64	99.45	99.6	-0.19	-0.03	0.15
20	Gray upper-body clothes	16	89.6	89.3	90.31	-0.29	0.71	1.01
21	Blue upper-body clothes	5	94.3	93.83	94.01	-0.47	-0.29	0.18
22	Green upper-body clothes	9	97.36	97.13	97.61	-0.23	0.25	0.48
23	Brown upper-body clothes	10	97.94	97.76	98.08	-0.19	0.14	0.32
	Mean		91.19	90.64	91.32	-0.55	0.13	0.68