Conditional Variance Regularization for Domain Shift Robustness in Transfer Learning

DAS in Data Science Capstone Project
Spring Semester 2024

Marius Tresoldi
mariust@student.ethz.ch
D-INFK

Supervisors:
Dr. Simon Dirmeier
Prof. Dr. Fernando Perez-Cruz

Zürich, April 19, 2024

Abstract

Conditional Variance Regularization (CoRe) is a recently developed technique used to train domain shift robust neural networks in computer vision, that is, models that retain high performance if the distribution of image properties such as rotation, position, size and image quality differs significantly between training and test data. The central idea is to augment a fraction of the training data with ID information that signifies the identity of the subject captured by the image, for example the name of the person in case of people. One can then differentiate between latent 'core' and 'style' features, where the style features correspond to the above image properties against which robustness is to be achieved. Assuming a specific causal inference framework, style features will exhibit much higher variance if conditioned on label Y and ID than core features. Consequently, the original formulation of CoRe achieves domain shift robustness by introducing a regularization term based on the variance of the predicted logits conditioned on (Y, ID). In this work, a modification of CoRe is presented in which the penalty is applied directly to the learned latent features instead. Using MNIST data it is shown that this Conditional Variance of Representation (CVR) yields comparable performance as the original Conditional Variance of Prediction (CVP) if the convolutional layers are sufficiently deep and the latent feature dimension does not significantly exceed the number of non-trivial (Y, ID) groups. It is demonstrated using CelebA data that representations learned with CVR can be transferred to another learning task such that the new predictor will also be domain shift robust without any need for further regularization or ID data collection. Hence, CVR is more suitable for deep pretrained computer vision models commonly used in transfer learning than CVP. Since CoRe can capture implicit style features, such models could benefit as they are currently guarded only with respect to explicit ones.

Contents

1. Introduction

One of the standard assumptions when training a neural network is that the test data is drawn from the same distribution as the samples the model is trained and validated on. In practice this assumption is often not satisfied, in which case one speaks of a domain shift. Training networks which exhibit robustness under such domain shifts remains a challenge in modern deep learning. A particularly instructive real-world example is provided by Badgeley et al. 2019, where a neural network is trained on radiograph data to predict hip fractures. People with hip fractures tend to be sent to the emergency room, where imaging machines with for example different image quality might be used than in a non-emergency hospital setting. Thus, instead of learning to predict a fracture based on the geometric properties of the hip bone alone, the network can misuse the latent features introduced by the hospital admission process by detecting which machine was used for imaging. When deploying such a predictor in a hospital, where for instance all machines are of the same type, performance will be significantly worse than during validation.

Recently, Heinze-Deml & Meinshausen 2021 introduced a new technique for training domain shift robust models called Conditional Variance Regularization (CoRe) relying on a causal inference framework based on the work of Gong et al. 2016, which is especially suited for computer vision tasks. As mentioned in the original CoRe paper, a particularly interesting line of future work is the application of the method to deeper models such as Inception or ResNet (Szegedy et al. 2015 and He et al. 2016). These models are commonly used for transfer learning in computer vision, that is, the representations learned by the convolutional layers of these models are reused and applied to a different task by fine-tuning the classification layer, thus significantly saving computer resources during training (see Zhuang et al. 2020 for a comprehensive review).

The goal of this project is to introduce a modification of the original CoRe formulation that is arguably more suitable for transfer learning. A brief formal introduction, the original CoRe formulation and the proposed modification are presented in section 2.1. Sections 2.2 and 3.1 present an experiment based on MNIST data that aims to directly compare the performance of the various CoRe variants. Finally, sections 2.3 and 3.2 apply the CoRe modification to a transfer learning task with CelebA data. A JAX/Flax implementation (Bradbury et al. 2018 and Heek et al. 2023) can be found on the following github repository: TreMarEd/Cond_Var_Regularization.

2. Methods

2.1. Domain Shift Robustness and CoRe

This section briefly provides some formalism to motivate the notion of domain shift robustness and CoRe. For a more thorough discussion, see Heinze-Deml & Meinshausen 2021.

The causal graph governing the data generation mechanism assumed henceforth is shown in figure 1. The tuple (X, Y, ID) is deemed to be observable, where X is the feature image, Y the label to be anti-causally predicted and ID is an identifier that represents the identity of the subject of which X is an image. For instance, if X contains faces of people, ID could be the name of the person displayed and Y could be a binary variable indicating the presence of eyeglasses. All these variables are causally influenced by an unobservable domain variable D. In the above eyeglass example, D could for instance represent the camera model used to capture the image and a domain shift would indicate a switch to a different camera model with different image quality. X is directly causally generated from two types of latent features termed 'core' features C and 'style' features S, the difference being that the style features have an additional causal influence from the domain D that is not mediated by the tuple (Y, ID), but some intermediary variable Δ. Note that Δ is mainly introduced by Heinze-Deml & Meinshausen 2021 to simplify mathematical proofs and one could draw a direct dependence between D and S. In the above eyeglass example, S might correspond to the resolution of the camera D and C might contain geometric information about eyeglasses worn by the person ID if Y=1. As style features contain unmediated information on D, it is their presence in the learned representations of a predictor that prevent domain shift robustness. C and S are distinguishable through the properties of their distributions if conditioned on (Y, ID): As opposed to $C|(Y, ID)$, $S|(Y, ID)$ will have additional volatility introduced by D, on which it is not conditioned. Thus, $S|(Y, ID)$ is likely to have bigger variance than $C|(Y, ID)$. This property will be exploited when defining CoRe below.

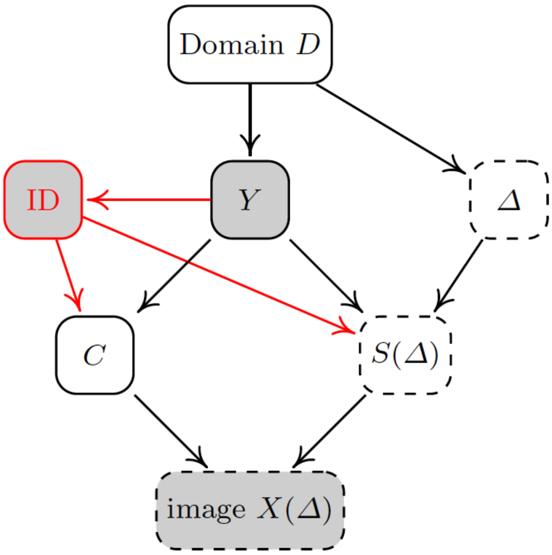

Figure 1: Causal graph governing the assumed data generation mechanism. Figure taken from Heinze-Deml & Meinshausen 2021.

Assume (Y, ID, S) to be distributed according to F_0 in the training and validation data. This implies that additionally to the usual features X and labels Y, it is required to gather ID data for each sample. The following class of distributions is now defined for $\epsilon > 0$, with $\mathcal{D}$ being some metric or divergence:

$$\mathcal{F}_\epsilon(F_0) = \{F : \mathcal{D}(F_0, F) \leq \epsilon\} \tag{1}$$

One can then define the risk relevant for domain shift robustness for some loss l and a predictor f_θ parametrized by $\theta \in \mathbb{R}^d$ as:

$$\mathcal{R}_\epsilon[f_\theta] = \sup_{F \in \mathcal{F}_\epsilon(F_0)} \mathbb{E}_F[l(Y, f_\theta(X)] \tag{2}$$

The overarching goal of domain shift robustness is then to solve:

$$\arg\min_\theta \mathcal{R}_\epsilon[f_\theta] \tag{3}$$

CoRe solves this problem by instead optimizing the following risk with conditional variance penalty $\mathcal{C}_\theta$ and regularization parameter λ:

$$\arg\min_\theta(\mathbb{E}_{F_0}[l(Y, f_\theta(X)] + \lambda\mathcal{C}_\theta) \tag{4}$$

$\mathcal{C}_\theta$ is defined by exploiting the above-mentioned properties of S and C by penalizing high variance conditioned on (Y, ID). In the original formulation $\mathcal{C}_\theta$ takes one of the following forms assuming f_θ to map to K prediction logits:

- Conditional Variance of Prediction (CVP): $\mathcal{C}_\theta = \mathbb{E}_{Y,ID\sim F_0}[\operatorname{tr}\operatorname{Cov}[f_\theta(X)|Y, ID]]$
- Conditional Variance of Loss (CVL): $\mathcal{C}_\theta = \mathbb{E}_{Y,ID\sim F_0}[\operatorname{Var}[l(Y, f_\theta(X)|Y, ID]]$

The empirical risk is then estimated via the unbiased sample covariance, such that an (Y, ID) group with only one element in the training data is defined to not contribute to the regularization term. Heinze-Deml & Meinshausen 2021 show that provided certain assumptions, solving problem (4) is equivalent to solving problem (3) to first order. Further, they show experimentally that CVP and CVL yield comparable results, such that in the following only CVP will be considered.

In this work, CVP is modified by assuming the predictor to be decomposable into an encoder e_Φ mapping images to their representations and a classifier $c_{\Phi'}$ mapping a representation to the predicted logits, such that $f_{(\Phi,\Phi')}(x) = c_{\Phi'}(e_\Phi(x))$. Conditional Variance of Representation (CVR) is then defined by applying the regularization directly to the learned representations instead of the predicted logits:

$$\mathcal{C}_\Phi = \mathbb{E}_{Y,ID\sim F_0}[\operatorname{tr}\operatorname{Cov}[e_\Phi(X)|Y, ID]] \tag{5}$$

Given that the conditional variance behavior described above is inherently a property of the latent features and only indirectly inherited by the predicted logits, it is argued here, that CVR is the canonical choice for regularization. Moreover, as will be shown in section 3.2 and as opposed to CVP, CVR allows to learn core features exclusively, such that the learned representations can be transferred to a different learning task while retaining domain shift robustness in the new predictor without additional need for regularization or ID data collection.

2.2. Comparison of CVP and CVR on MNIST Data

The experimental setup presented in section 5.5 of Heinze-Deml & Meinshausen 2021 relying on the classification of MNIST handwritten digit images (Deng 2012) is adopted to directly compare the behavior of CVP and CVR. The model architectures used here are different than in the original experiment, though the results therein can be reproduced up to 1% with the present JAX/Flax implementation. The idea is to artificially introduce two different domains with respect to which robustness is to be achieved: the first domain is given by the non-rotated digits in the original data set, while the second domain contains digits randomly rotated by an angle uniformly distributed in [35°, 70°].

Training data consists of 10'000 original data points, of which c=200 are randomly chosen and artificially augmented by said rotation, such that the final training set has n=10'200. The ID variable of an augmented sample is set to the same value as for the original data point. Thus, in the context of CoRe, there are exactly 200 non-trivial (Y, ID) groups, that is, (Y, ID) groups which consist of more than one data point and hence generally contribute positively to $\mathcal{C}_\theta$. The remaining 9'800 trivial (Y, ID) groups consist of an individual data point and do not contribute to the CoRe penalty. The validation set is structured the same way. A specific architecture is trained for $\lambda \in \{0.1, 1, 10, 100\}$ and the parameter and training epoch with the smallest validation loss is selected. Two test sets with n=10'000 are defined without any augmented data, each corresponding to one of the domains: test set 1 is comprised of non-rotated and test set 2 of rotated digits. Training and test set 2 are visualized in figure 2.

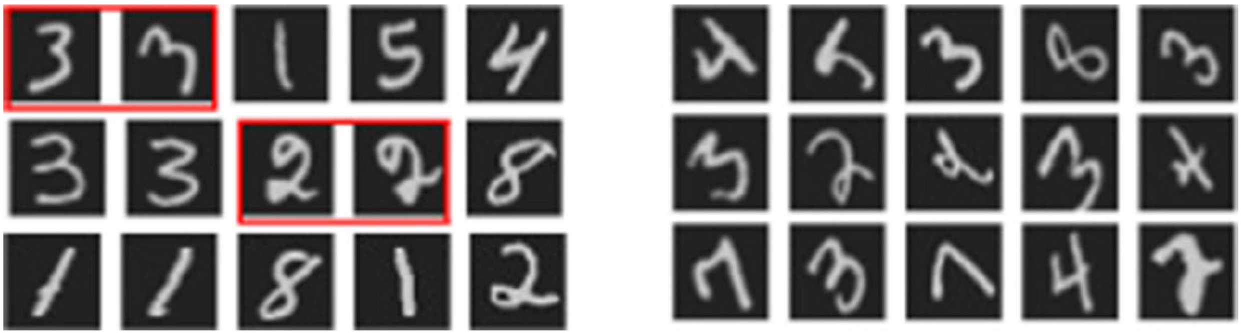

Figure 2: Left: visualization of the MNIST training data with n=10'200, where 200 of the original images are augmented with their rotated counterparts (red). Right: visualization of the domain shifted MNIST test 2 data with n=10'000, where all images are rotated by a random amount. Figure taken from Heinze-Deml & Meinshausen 2021.

The goal is to compare the performance of no regularization, CVP and CVR based on this data set along a wide variety of model architectures as parametrized by the number of convolutional layers L and the number of learned latent features q: the first architecture has low model complexity with $L = 2$ and $q = 144$, the second model has larger complexity with $L = 4$ and a small amount of latent features $q = 16$ and the final model also has $L = 4$ but a much bigger number of latent features $q = 3136$. Detailed model architectures can be found in appendix A. All architectures are trained for 30 epochs, with a learning rate of 0.003 and Adam optimizer (Kingma & Ba 2015). The experiment is carried out for 5 different RNG seeds over which the final results are averaged.

During training the efficient estimation of $\mathcal{C}_\theta$ requires efficient determination of (Y, ID) group membership for each data point in a batch. Further, a technical challenge is posed by CoRe requiring each (Y, ID) group to be fully contained in exactly one training batch per epoch. These challenges are handled here with the following batch generation scheme for batch size b: for each batch the first $b - 2d$ data points will stem from trivial and the last $2d$ data points from non-trivial groups, d being the fixed number of non-trivial groups per batch. The factor 2 comes from all non-trivial groups in the experiment having size 2. Members of the same group are consecutive in the batch. All architectures are trained with $b = 102$, $d = 2$ and 100 batches. With this scheme, batch generation still constitutes a computational bottleneck in the CoRe implementation presented here, such that batches are not reshuffled after each epoch. The experiment described above has been carried out for an individual RNG seed with reshuffling after each epoch without resulting in significantly different results.

2.3. Transfer Learning with CVR on CelebA Data

The goal of the experiment is to assess whether domain shift robustness is retained if CVR regularized latent features are transferred to a different learning task. CelebA data (Liu et al. 2015) is used, that is, images of celebrity faces. The setup is analogous to section 5.3 of Heinze-Deml & Meinshausen 2021, which however can not be reproduced here, as the exact class balance in the training data is not reported.

The prediction problem concerns the binary classification of one of the following four facial hair labels: beard, mustache, goatee and sideburns. These labels were chosen because of their relative similarity, making them ideal candidates for susccessful transfer learning. Again, two domains are artificially introduced with respect to which robustness is to be achieved. In domain 1 the images of people without the relevant facial hair label (Y=0) have normal image quality, while images of people with the relevant facial hair (Y=1) have artificially degraded image quality. In domain 2 this relationship is inverted. All images are resized to 64x48x3 and for degradation ImageMagick is used (https://imagemagick.org/) by sampling the image quality from $\mathcal{N}(30, 100)$. For all datasets 25% of the images have Y=1. Training data is subject to domain 1, has size n=20'400 and consists of: 15'000 Y=0 data points with original image quality, 5'000 Y=1 data points with degraded image quality, of which 400 are chosen at random to be augmented with their original quality counterpart. The augmented data points receive the same ID as their counterparts. Validation data is structured the same, but scaled down to n=5'100. Test sets 1 and 2 have n=5'000 without any augmented data and correspond to domain 1 and 2 respectively.

For each facial hair label the following three models are trained (detailed architectures can be found in appendix A): an unregularized model, a CVR-regularized model with $\lambda = 500$ and an unregularized model using only the transferred, CVR-regularized beard latent features and a fully connected softmax classification layer. The experiment is carried out for 3 different RNG seeds over which the results are averaged. The same batch generation scheme as described in section 2.2 is applied, using 200 batches with $b = 102$, $d = 2$, a learning rate of 0.005 and Adam optimizer.

3. Results and Discussion

3.1. Comparison of CVP and CVR on MNIST Data

Table 1 shows the results of the comparison of CVP and CVR on the MNIST data set using different model architectures, with L representing the number of convolutional layers and q being the number of learned latent features. For all architectures the unregularized models are not domain shift robust: test set 1 and 2 accuracies differ by 25% to 30% because the predictor is not sufficiently trained on rotated digits. While CVP slightly reduces test set 1 performance, it also increases domain shift robustness by raising test set 2 accuracy between 13% and 18% throughout all architectures.

The picture is more varied for CVR. For the architecture with only two convolutional layers no regularizing effect is discernible, as the test set 2 accuracy is even worse than in the unregularized case. This difference between CVP and CVR can be explained by the fact that CVP has two degrees of freedom to achieve a domain shift robust predictor: i) not learning style features in the convolutional layers and ii) ignoring learned style features in the classification layer. CVR on the other hand, can only make use of the former. With only two convolutional layers, the model is not complex enough to learn core features exclusively, such that CVR fails, while CVP learns to ignore the remaining style features in the classification layer to achieve domain shift robustness. This makes CVP unsuitable for transfer learning, as the transferred representations might generally still contain style features. In the case of 4 convolutional layers and a low number of latent features ($q = 16$), the convolutional layers are complex enough to learn core features exclusively and consequently, CVR achieves comparable domain shift robustness as its CVP counterpart.

For 4 convolutional layers and a high number of latent features ($q = 3136$) CVR does have a regularizing effect, which is however significantly smaller than for CVP. This is likely attributable to the circumstance that with only 200 augmented data points CVR effectively has only 200 samples available to learn to regularize 3136 features, while CVP is performed on the 10 output logits only. Hence, for CVR training to be stable, the number of non-trivial (Y, ID) groups should be of similar order of magnitude as q.

Generally, CVR has a less adverse effect on test set 1 accuracy than CVP. This is likely due to CVR not directly interfering with the classification layer during training, where the data is dominated by the test set 1 domain.

	L=2, q=144		**L=4, q=16**		**L=4, q=3136**	
domain	test 1	test 2	test 1	test 2	test 1	test 2
non-regularized	97.0	68.8	97.4	72.9	98.2	78.3
CVP	96.0	86.7	97.2	85.2	95.2	88.1
CVR	97.3	65.1	97.6	84.0	98.8	83.4

Table 1: Accuracies in % of various models on MNIST test set 1 and 2. The domain of test set 1 is not shifted, meaning digits are non-rotated, while in test set 2 they are rotated. L signifies the number of convolutional layers used, while q is the dimension of the learned representation. For each architecture the first model is unregularized, the second CVP-regularized and the third CVR-regularized.

3.2. Transfer Learning with CVR on CelebA Data

Table 2 shows the results of the transfer learning experiment on the CelebA data set concerning various facial hair labels. The unregularized models are not domain shift robust: test set 1 and 2 accuracies differ between 20% and 30% because the predictor learns to misuse the spurious correlation between presence of facial hair and image quality. CVR regularization reduces the test set 1 performance by 3% to 4% but significantly enhances domain shift robustness by increasing test set 2 performance by 10% to 20%. This successful training with CVR is in accordance with the findings of section 3.1 as the the convolutional layers are relatively complex ($L = 4$) and the number of latent features has similar order of magnitude as the number of non-trivial (Y, ID) groups ($q = 72$, $c = 400$).

The unregularized transfer models, which make use of the pretrained regularized beard representations, achieve similar performance on both test set 1 and test set 2 as their explicitly regularized counterparts. This implies that the regularized beard model exclusively learned domain shift robust core features that generalize well for the task of detecting facial hair, resulting in successful transfer learning.

	beard		**mustache**		**goatee**		**sideburns**	
domain	test1	test2	test1	test2	test1	test2	test1	test2
non-regularized	94	66	92	63	92	73	92	68
CVR λ=500	90	84	88	83	89	84	89	83
transfer	-	-	88	85	89	85	89	85

Table 2: Accuracies in % of various models on CelebA facial hair labels on test set 1 and test set 2. The domain of test set 1 is not shifted, meaning people with the respective facial hair have degraded and people without have normal image quality. This is reversed for the domain shifted test set 2. The first model is unregularized and the second is CVR-regularized. The third model is unregularized and makes use of the transferred representations of the regularized beard model.

4. Summary and Conclusion

A modification of CoRe for domain shift robustness in deep learning computer vision tasks was introduced, where the penalty is applied directly to the representations learned by the convolutional layers (CVR) instead of the logits predicted by the classification layer (CVP). By applying both methods to the MNIST data set, it was shown that CVR achieves similar performance as CVP if the convolutional layers are sufficiently complex and the number of latent features to be learned is of similar order of magnitude as the number of non-trivial (Y, ID) groups in the training data. Using various facial hair labels in the CelebA data set, it was shown that as opposed to CVP, CVR can be used to exclusively learn latent core features which when transferred to another learning task result in domain shift robust predictors without any need for explicit regularization or additional ID data collection. This makes CVR more suitable than CVP for pretrained computer vision models commonly used in transfer learning, such as ResNet or Inception. CoRe might benefit such models, as they are currently guarded only with respect to explicitly predefined style features and not implicit ones as captured by CoRe (Heinze-Deml & Meinshausen 2021).

The transfer learning task on which the method was applied was relatively simple: only a small, resized sample was used and the labels were all centered on facial hair, such that the learned representations are likely not particularly general. Hence, in future work it would be interesting to apply this scheme to more complicated tasks with much deeper architectures comparable to the above-mentioned pretrained models commonly used in transfer learning.

Other lines of future work could encompass generalizing CVR to non-computer vision or regression tasks, proofing similar formal theorems as presented in Heinze-Deml & Meinshausen 2021 for CVP and applying CVR to an experimental setup with unknown style intervention as in section 5.3 of Heinze-Deml & Meinshausen 2021. Moreover, the need for each (Y, ID) group to be at each training epoch fully contained in exactly one batch makes efficient batch creation challenging and might be further improved upon, especially for the more realistic case of varying non-trivial (Y, ID) group size.

References

Badgeley, M. A., Zech, J. R., Oakden-Rayner, L., Glicksberg, B. S., Liu, M., Gale, W., McConnell, M. V., Percha, B., Snyder, T. M., & Dudley, J. T. (2019). Deep learning predicts hip fracture using confounding patient and healthcare variables. *npj Digital Medicine*, *2*(1), 31.
URL `https://doi.org/10.1038/s41746-019-0105-1`

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., & Zhang, Q. (2018). JAX: composable transformations of Python+NumPy programs.
URL `http://github.com/google/jax`

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, *29*(6), 141–142.

Gong, M., Zhang, K., Liu, T., Tao, D., Glymour, C., & Schölkopf, B. (2016). Domain adaptation with conditional transferable components. In *JMLR workshop and conference proceedings*, vol. 48.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Heek, J., Levskaya, A., Oliver, A., Ritter, M., Rondepierre, B., Steiner, A., & van Zee, M. (2023). Flax: A neural network library and ecosystem for JAX.
URL `http://github.com/google/flax`

Heinze-Deml, C., & Meinshausen, N. (2021). Conditional variance penalties and domain shift robustness. *Machine Learning*, *110*, 303–348.

Kingma, D., & Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

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

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2020). A comprehensive survey on transfer learning. In *Proceedings of the IEEE*.

Appendices

A. Model Architectures

	input dim.	conv. layer	conv. stride	class. layer	activation
MNIST L=2, q=144	28x28x1	5x5x16, 5x5x16, 2x2 avg. pool	2	fully connected softmax	ReLU
MNIST L=4, q=16	28x28x1	5x5x16, 5x5x16, 5x5x16, 5x5x16, 2x2 avg. pool	2	fully connected softmax	ReLU
MNIST L=4, q=3136	28x28x1	5x5x16, 5x5x16, 5x5x16, 5x5x16, 2x2 avg. pool	1	fully connected softmax	ReLU
CelebA	64x48x3	4x3x16, 4x3x16, 4x3x16, 4x3x16,	2	fully connected softmax	leaky ReLU

Table 3: Specification of the detailed architectures of all relevant models.