Weekly Report I
for Laboratory Research

University of Houston

May 18, 2018

Outline

Outline

Dual-Agent GANs

Dual-Agent GANs for Photorealistic and Identity Preserving Profile Face Synthesis [1]

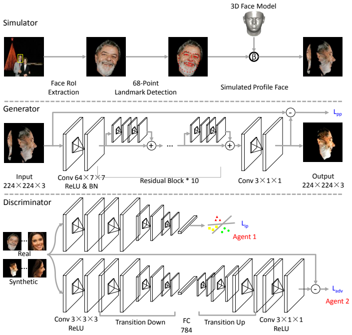


Figure 1: Dual-Agent GANs architecture.

- Using a 3D model to transform a face into the side view.
- Use GAN to compensate the blurred details of the side view (synthesis) faces.

Dual-Agent GANs

Dual-Agent GANs for Photorealistic and Identity Preserving Profile Face Synthesis

The whole loss function

Minimize the discriminator loss $\mathscr{L}_{D_\phi}$ and generator loss $\mathscr{L}_{G_\phi}$ alternatively.

$$\begin{aligned} \mathscr{L}_{D_\phi} &= \mathscr{L}_{\mathrm{adv}} + \lambda_1 \mathscr{L}_{\mathrm{ip}}, \\ \mathscr{L}_{G_\phi} &= \left(-\mathscr{L}_{\mathrm{adv}} + \lambda_1 \mathscr{L}_{\mathrm{ip}}\right) + \lambda_2 \mathscr{L}_{\mathrm{pp}}. \end{aligned} \tag{1}$$

The adversarial loss

The adversarial loss (Wasserstein distance [2]) computes the difference between the distributions of real faces and synthesis faces.

$$\begin{aligned} \mathscr{L}_{\mathrm{adv}} &= \sum_j |y_j - D_\phi(y_j)| - k_t \sum_i |\tilde{x}_i - D_\phi(\tilde{x}_i)|, \\ k_{t+1} &= k_t + \alpha \left(\gamma \sum_j |y_j - D_\phi(y_j)| - \sum_i |\tilde{x}_i - D_\phi(\tilde{x}_i)| \right). \end{aligned} \tag{2}$$

Dual-Agent GANs

Dual-Agent GANs for Photorealistic and Identity Preserving Profile Face Synthesis

The whole loss function

Minimize the discriminator loss $\mathscr{L}_{D_\phi}$ and generator loss $\mathscr{L}_{G_\phi}$ alternatively.

$$\begin{aligned} \mathscr{L}_{D_\phi} &= \mathscr{L}_{\text{adv}} + \lambda_1 \mathscr{L}_{\text{ip}}, \\ \mathscr{L}_{G_\phi} &= \left(-\mathscr{L}_{\text{adv}} + \lambda_1 \mathscr{L}_{\text{ip}}\right) + \lambda_2 \mathscr{L}_{\text{pp}}. \end{aligned} \tag{1}$$

The identity loss

The identity loss preserves the identity of the same face.

$$\begin{aligned} \mathscr{L}_{\text{ip}} = & \frac{1}{N}\sum_j -\left(Y_j \log(D_\phi(y_j)) + (1 - Y_j)\log(1 - D_\phi(y_j))\right) \\ & + \frac{1}{N}\sum_i -\left(Y_i \log(D_\phi(\tilde{x}_i)) + (1 - Y_i)\log(1 - D_\phi(\tilde{x}_i))\right). \end{aligned} \tag{2}$$

Dual-Agent GANs

Dual-Agent GANs for Photorealistic and Identity Preserving Profile Face Synthesis

The whole loss function

Minimize the discriminator loss $\mathscr{L}_{D_\phi}$ and generator loss $\mathscr{L}_{G_\phi}$ alternatively.

$$\begin{aligned}\mathscr{L}_{D_\phi} &= \mathscr{L}_{\mathrm{adv}} + \lambda_1 \mathscr{L}_{\mathrm{ip}}, \\ \mathscr{L}_{G_\phi} &= \left(-\mathscr{L}_{\mathrm{adv}} + \lambda_1 \mathscr{L}_{\mathrm{ip}}\right) + \lambda_2 \mathscr{L}_{\mathrm{pp}}.\end{aligned} \tag{1}$$

The pixel-wise loss

The pixel-wise loss preserves the shape from the original synthesis face.

$$\begin{aligned}\mathscr{L}_{\mathrm{pp}} &= \sum_j |x_i - \tilde{x}_i|, \\ \tilde{x}_i &= G_\phi(x).\end{aligned} \tag{2}$$

Dual-Agent GANs
Results

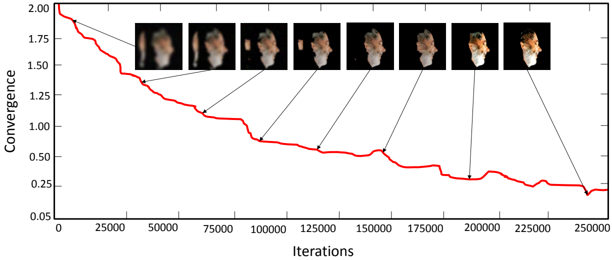
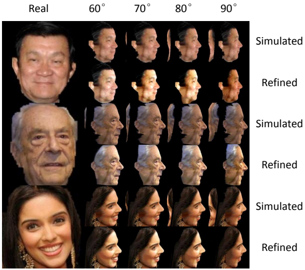
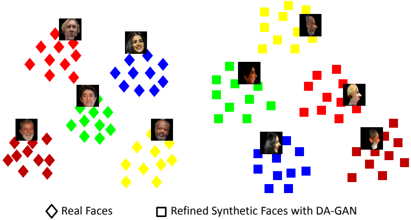


Figure 2: Quality of refined results w.r.t. the network convergence measurement.

Dual-Agent GANs
Results



(a) Refined results of DA-GAN.



(b) Feature space of real faces and DA-GAN synthetic faces.

Figure 2: Qualitative analysis of DA-GAN.

Outline

Transferable Learning

Label Efficient Learning of Transferable Representations across Domains and Tasks [3]

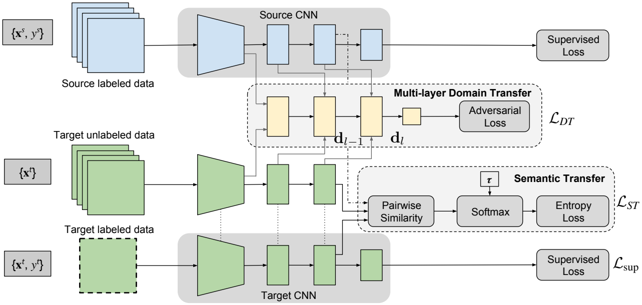


Figure 3: Transferable Learning architecture.

- Use a multi-layer domain discriminator to make the distribution of unsupervised learning branch coherent to supervised one.
- Use a similarity function to transform the label in source domain into that in target domain.

Transferable Learning

Label Efficient Learning of Transferable Representations across Domains and Tasks

The whole loss function

$$\mathcal{L} = \mathcal{L}_{\text{sup}} + \alpha \mathcal{L}_{DT} + \beta \mathcal{L}_{ST}. \quad (3)$$

The supervised learning loss

The supervised learning loss which is from the target domain is defined in the most traditional way.

$$\mathcal{L}_{\text{sup}} = \mathbb{E}_{\mathscr{X}^{\mathscr{T}},\, \mathscr{Y}^{\mathscr{T}}} \left(E^t(\mathbf{x}^t) - y_t \right). \quad (4)$$

Transferable Learning

Label Efficient Learning of Transferable Representations across Domains and Tasks

The whole loss function

$$\mathscr{L} = \mathscr{L}_{\text{sup}} + \alpha \mathscr{L}_{DT} + \beta \mathscr{L}_{ST}. \tag{3}$$

The domain discriminative loss

The domain discriminative loss is comprised by the logarithmic differences from losses in source and target domains.

$$\begin{aligned} \mathscr{L}_{DT} = &-\mathbb{E}_{\mathscr{X}^{\mathscr{S}}}\left(\log(\mathbf{d}_l^{s})\right) - \mathbb{E}_{\mathscr{X}^{\hat{\mathscr{T}}}}\left(\log(1-\mathbf{d}_l^{\tilde{t}})\right) \\ &-\mathbb{E}_{\mathscr{X}^{\mathscr{S}}}\left(\log(1-\mathbf{d}_l^{s})\right) - \mathbb{E}_{\mathscr{X}^{\hat{\mathscr{T}}}}\left(\log(\mathbf{d}_l^{\tilde{t}})\right). \end{aligned} \tag{4}$$

Transferable Learning

Label Efficient Learning of Transferable Representations across Domains and Tasks

The whole loss function

$$\mathscr{L} = \mathscr{L}_{\text{sup}} + \alpha \mathscr{L}_{DT} + \beta \mathscr{L}_{ST}. \tag{3}$$

The semantic transfer loss

The semantic transfer loss is based on calculating the similarities between samples from different domains.

$$\begin{aligned} \mathscr{L}_{ST} = \mathscr{L}_{ST}\left(\mathscr{X}^{\tilde{\mathscr{T}}}, \mathscr{X}^{\mathscr{S}}\right) + \mathscr{L}_{ST,\,\text{sup}}\left(\mathscr{X}^{\mathscr{T}}\right) \\ + \mathscr{L}_{ST,\,\text{unsup}}\left(\mathscr{X}^{\tilde{\mathscr{T}}}, \mathscr{X}^{\mathscr{T}}\right). \end{aligned} \tag{4}$$

Transferable Learning

Label Efficient Learning of Transferable Representations across Domains and Tasks

The semantic transfer loss

The semantic transfer loss is based on calculating the similarities between samples from different domains.

$$\mathcal{L}_{ST} = \mathcal{L}_{ST}\left(\mathcal{X}^{\tilde{\mathcal{T}}},\, \mathcal{X}^{\mathcal{S}}\right) + \mathcal{L}_{ST,\,\text{sup}}\left(\mathcal{X}^{\mathcal{T}}\right) + \mathcal{L}_{ST,\,\text{unsup}}\left(\mathcal{X}^{\tilde{\mathcal{T}}},\, \mathcal{X}^{\mathcal{T}}\right). \tag{3}$$

The semantic transfer loss (cross domain)

Use similarity to calculate the cross domain entropy between the source and the unsupervised target.

$$\mathcal{L}_{ST}\left(\mathcal{X}^{\tilde{\mathcal{T}}},\, \mathcal{X}^{\mathcal{S}}\right) = \mathbb{E}_{\mathcal{X}^{\tilde{\mathcal{T}}}}\left(H\left(\frac{\sigma}{\tau} v_s(\tilde{\mathbf{x}}^t)\right)\right), \quad [v_s(\tilde{\mathbf{x}}^t)]_i = \psi(\tilde{\mathbf{x}}^t,\, \mathbf{x}_i^s). \tag{4}$$

Transferable Learning

Label Efficient Learning of Transferable Representations across Domains and Tasks

The semantic transfer loss

The semantic transfer loss is based on calculating the similarities between samples from different domains.

$$\mathscr{L}_{ST} = \mathscr{L}_{ST}\left(\mathscr{X}^{\tilde{\mathscr{T}}},\ \mathscr{X}^{\mathscr{S}}\right) + \mathscr{L}_{ST,\ \text{sup}}\left(\mathscr{X}^{\mathscr{T}}\right) + \mathscr{L}_{ST,\ \text{unsup}}\left(\mathscr{X}^{\tilde{\mathscr{T}}},\ \mathscr{X}^{\mathscr{T}}\right). \tag{3}$$

The semantic transfer loss (supervised)

Define the centroid of each class as $c_i^{\mathscr{T}}$, then we have.

$$\mathscr{L}_{ST,\ \text{sup}}\left(\mathscr{X}^{\mathscr{T}}\right) = -\mathbb{E}_{\mathscr{X}^t}\left(\log\left(\text{softmax}\left(v_t(\mathbf{x}^t)\right)\right)\right), \quad [v_t(\mathbf{x}^t)]_i = \psi\left(\mathbf{x}^t,\ c_i^{\mathscr{T}}\right). \tag{4}$$

Transferable Learning

Label Efficient Learning of Transferable Representations across Domains and Tasks

The semantic transfer loss

The semantic transfer loss is based on calculating the similarities between samples from different domains.

$$\mathcal{L}_{ST} = \mathcal{L}_{ST}\left(\mathcal{X}^{\tilde{\mathcal{T}}},\ \mathcal{X}^{\mathcal{S}}\right) + \mathcal{L}_{ST,\ \mathrm{sup}}\left(\mathcal{X}^{\mathcal{T}}\right) + \mathcal{L}_{ST,\ \mathrm{unsup}}\left(\mathcal{X}^{\tilde{\mathcal{T}}},\ \mathcal{X}^{\mathcal{T}}\right). \tag{3}$$

The semantic transfer loss (unsupervised)

Use similarity to calculate the cross domain entropy between unsupervised loss and supervised one.

$$\mathcal{L}_{ST,\ \mathrm{unsup}}\left(\mathcal{X}^{\tilde{\mathcal{T}}},\ \mathcal{X}^{\mathcal{T}}\right) = \mathbb{E}_{\mathcal{X}^{\tilde{\mathcal{T}}}}\left(H\left(\frac{\sigma}{\tau} v_t(\tilde{\mathbf{x}}^t)\right)\right), \quad [v_t(\tilde{\mathbf{x}}^t)]_i = \psi\left(\tilde{\mathbf{x}}^t,\ c_i^{\mathcal{T}}\right). \tag{4}$$

Dual-Agent GANs

Transferable Learning

Theory

Results

Clustering with deep learning

De-animation

Reference

Transferable Learning

Results

Figure 4: Transfer the learned representation on SVHN digits 0-4 (left) to MNIST digits 5-9 (right).

Table 1: Qualitative analysis of transfered learning model, where **k** is the number of used labels.

Method	k=2	k=3	k=4	k=5
Target only	0.642 ± 0.026	0.771 ± 0.015	0.801 ± 0.010	0.840 ± 0.013
Fine-tune	0.612 ± 0.020	0.779 ± 0.018	0.802 ± 0.016	0.830 ± 0.011
Matching nets	0.469 ± 0.019	0.455 ± 0.014	0.566 ± 0.013	0.513 ± 0.023
Fine-tuned matching nets	0.645 ± 0.019	0.755 ± 0.024	0.793 ± 0.013	0.827 ± 0.011
Ours: fine-tune + adv.	0.702 ± 0.020	0.800 ± 0.013	0.804 ± 0.014	0.831 ± 0.013
Ours: full model ($\gamma = 0.1$)	$\mathbf{0.917} \pm \mathbf{0.007}$	$\mathbf{0.936} \pm \mathbf{0.006}$	$\mathbf{0.942} \pm \mathbf{0.006}$	$\mathbf{0.950} \pm \mathbf{0.004}$

Outline

Clustering with deep learning

Deep Subspace Clustering Networks [4]

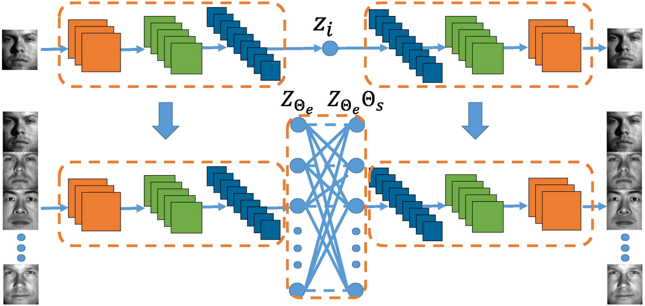


Figure 5: Deep Subspace Clustering Networks' architecture.

- The basic structure is autoencoder.
- Insert the self-expressive (SE) problem between the encoder and decoder.

Clustering with deep learning

Deep Subspace Clustering Networks

The whole loss function

Minimize the combined loss of autoencoder's and SE's.

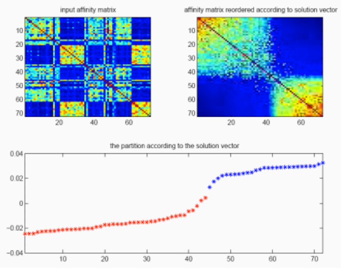
$$\mathscr{L} = \frac{1}{2}\|\mathbf{X} - D(E(\mathbf{X}))\|_F^2 + \lambda_1\|\mathbf{C}\|_p + \frac{\lambda_2}{2}\|E(\mathbf{X}) - E(\mathbf{X})\mathbf{C}\|_F^2,$$
$$\text{s.t. } \operatorname{diag}(\mathbf{C}) = \mathbf{0}. \tag{5}$$

Traditional SE problem

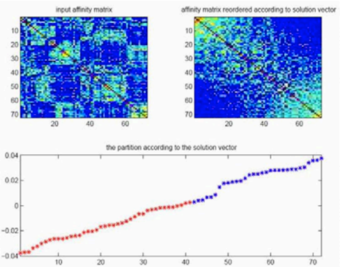
Assuming that we have N samples with a length of L, then all samples comprise a matrix $\mathbf{X}$ with a size of $L \times N$,

$$\min_{\mathbf{C}} \|\mathbf{C}\|_p + \frac{\lambda}{2}\|\mathbf{X} - \mathbf{X}\mathbf{C}\|_F^2,$$
$$\text{s.t. } \operatorname{diag}(\mathbf{C}) = \mathbf{0}, \tag{6}$$

Clustering with deep learning
Results



Good similarity measure



Poor similarity measure

Figure 6: An example of clustering by using affinity matrix.

Clustering with deep learning
Results

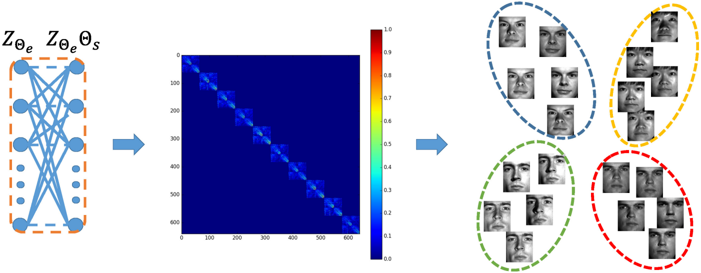


Figure 6: Using the middle part of the network, i.e. the SE model, we could construct the affinity matrix for clustering.

Outline

De-animation

Learning Physical Intuition of Block Towers by Example [6]

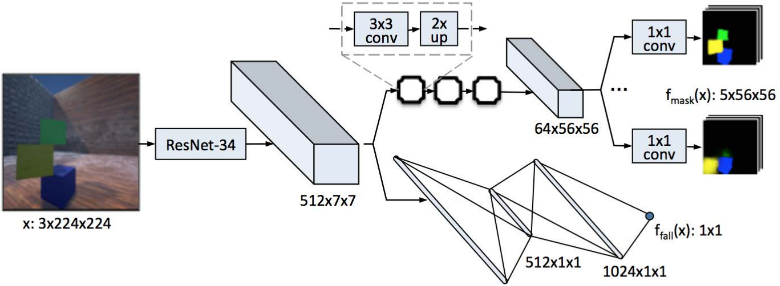


Figure 7: The PhysNet architecture.

- This network is adapted from DeepMask [5]. The input is the whole image rather than a local patch.
- Use multiple kernels to produce multiple frames.
- Replace the score value with the prediction value.

De-animation

Learning to See Physics via Visual De-animation [7]

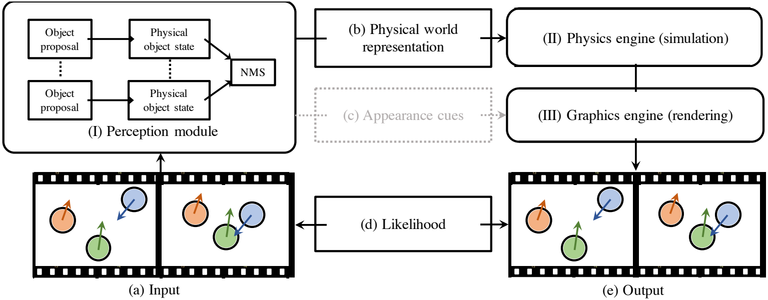
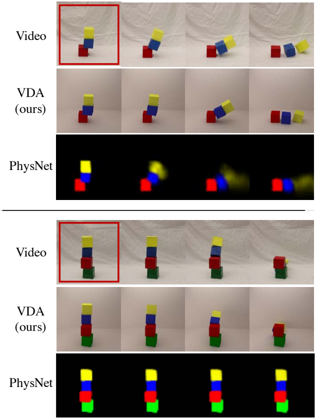


Figure 8: The visual de-animation architecture.

- An autoencoder-lite structure. While the decoder is a physical model (from physical states to objects) and the encoder is CNN (from objects to physical states).
- The decoder, i.e. the physical engine could be differentiable or not.

De-animation
Results



(a) Our reconstruction and prediction results given a single frame (marked in red). From top to bottom: ground truth, our results, results from Lerer et al. [2016].

Methods	# Blocks			Mean
	2	3	4	
Chance	50	50	50	50
Humans	67	62	62	64
PhysNet	66	66	73	68
GoogLeNet	70	70	70	70
VDA (init)	73	74	72	73
VDA (joint)	75	76	73	75
VDA (full)	76	76	74	75

(b) Accuracy (%) of stability prediction on the blocks dataset

Methods	2	3	4	Mean
PhysNet	56	68	70	65
GoogLeNet	70	67	71	69
VDA (init)	74	74	67	72
VDA (joint)	75	77	70	74
VDA (full)	76	76	72	75

(c) Accuracy (%) of stability prediction when trained on synthetic towers of 2 and 4 blocks, and tested on all block tower sizes.

Figure 9: Compare the performances of PhysNet and VDA.

Outline

Reference I

- J. Zhao, L. Xiong, P. Karlekar Jayashree, J. Li, F. Zhao, Z. Wang, P. Sugiri Pranata, P. Shengmei Shen, S. Yan, and J. Feng, "Dual-agent gans for photorealistic and identity preserving profile face synthesis," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 66–76. [Online]. Available: http://papers.nips.cc/paper/6612-dual-agent-gans-for-photorealistic-and-identity-preserving-profile pdf
- D. Berthelot, T. Schumm, and L. Metz, "BEGAN: boundary equilibrium generative adversarial networks," *CoRR*, vol. abs/1703.10717, 2017. [Online]. Available: http://arxiv.org/abs/1703.10717
- Z. Luo, Y. Zou, J. Hoffman, and L. Fei-Fei, "Label Efficient Learning of Transferable Representations across Domains and Tasks," *ArXiv e-prints*, Nov. 2017.

Reference II

- P. Ji, T. Zhang, H. Li, M. Salzmann, and I. D. Reid, "Deep subspace clustering networks," *CoRR*, vol. abs/1709.02508, 2017. [Online]. Available: http://arxiv.org/abs/1709.02508
- P. H. O. Pinheiro, R. Collobert, and P. Dollár, "Learning to segment object candidates," *CoRR*, vol. abs/1506.06204, 2015. [Online]. Available: http://arxiv.org/abs/1506.06204
- A. Lerer, S. Gross, and R. Fergus, "Learning physical intuition of block towers by example," *CoRR*, vol. abs/1603.01312, 2016. [Online]. Available: http://arxiv.org/abs/1603.01312
- J. Wu, E. Lu, P. Kohli, B. Freeman, and J. Tenenbaum, "Learning to see physics via visual de-animation," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 153–164. [Online]. Available: http://papers.nips.cc/paper/6620-learning-to-see-physics-via-visual-de-animation.pdf