TransGaGa: Geometry-Aware Unsupervised Image-to-Image Translation
Appendix

Wayne Wu[1] Kaidi Cao[2] Cheng Li[1] Chen Qian[1] Chen Change Loy[3]

[1]SenseTime Research [2]Stanford University
[3]Nanyang Technological University

{wuwenyan, chengli, qianchen}@sensetime.com kaidicao@cs.stanford.edu
ccloy@ntu.edu.sg

Abstract

In this appendix, more details of the training and network architecture are provided. We also demonstrate more qualitative results due to the limit of space in the main paper. In addition, we discuss the limitation of our method.

1. Training Details

We train our model in two main steps, *i.e.*, separated-training and joint-training.

Separated-training. As described in [16, 5], unsupervised landmark detection is challenging and many schemas need to be leveraged to regularize the training. In our experiments, we also find it hard to train the geometry estimator $E^g_\cdot$ together with other components at the very start. Thus, we train the conditional VAE network in X/Y domain separately as shown in Fig. 1. The loss for separated-training is defined as:

$$\mathcal{L}_{\text{separ}} = \mathcal{L}_{\text{CVAE}} + \mathcal{L}_{\text{prior}} \tag{1}$$

The conditional VAE network consists with an unsupervised geometry estimator $E^g_\cdot(;\pi)$, a geometry encoder $E^c_\cdot(;\theta)$ which embeds the landmark heatmaps into the latent space $C_\cdot$, an appearance encoder $E^a_\cdot(;\phi)$ which embeds the appearance information into the latent space $A_\cdot$, and a decoder $D_\cdot(;\theta) : C_\cdot \times A_\cdot \rightarrow X/Y$, which maps the latent space back to the image space. Inspired by [7, 13, 2], we model $p_\theta(x|g,z)$ as a parametric Laplace and $q_\phi(z|x,g)$ as a parametric Gaussian distribution. The parameters can be estimated by $E^c_\cdot(;\theta)$, $E^a_\cdot(;\phi)$ and $D_\cdot(;\theta)$ respectively. Thus, we implement the conditional VAE loss as:

$$\begin{aligned}\mathcal{L}_{\text{CVAE}}(x,\theta,\phi) = & KL(q_\phi(z|x,g)||p_\theta(z|g)) \\ & + \mathbb{E}_{q_\phi(z|x,g)}[\log p_\theta(x|g,z)]\end{aligned} \tag{2}$$

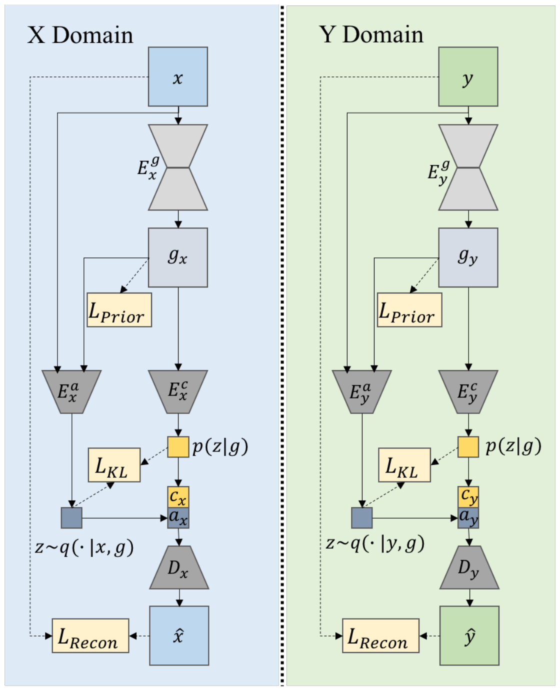

Figure 1: **Separated training.** $x \in \mathbb{R}^{256\times256\times3}$: input image. $g_x \in \mathbb{R}^{256\times256\times30}$: landmark heatmaps. $d_x \in \mathbb{R}^{1\times60}$: landmark coordinates. $c_x \in \mathbb{R}^{1\times256}$: geometry latent code. $a_x \in \mathbb{R}^{1\times256}$: appearance latent code. $\hat{x} \in \mathbb{R}^{256\times256\times3}$: generated image. E^g_x: geometry estimator. E^c_x: geometry encoder. E^a_x: appearance encoder. D_x: decoder for X domain. In practice, we take only x as the input of E^a_x and it can be found little impact to the performance. Components are defined similarly in Y domain.

where the first term is the Kullback-Leibler divergence $\mathcal{L}_{\text{KL}}$. Following [2], the second term can be implemented with a reconstruction loss. Then, the loss can be formulated as:

$$\mathcal{L}_{\text{CVAE}}(x,\theta,\phi) = KL(q_\phi(z|x,g)||p_\theta(z|g)) \\ +\|\hat{x}-x\|_1 + \sum_l \|\psi_l(\hat{x}) - \psi_l(x)\|_1 \tag{3}$$

where $\hat{x}$ is $D_\cdot(E^c_\cdot(g_\cdot), E^a_\cdot(x))$ and ψ_l is the feature obtained from l-th layer of a VGG-19 model [12] pre-trained on ImageNet. Using the reparametrization trick [7], these networks can be trained end-to-end. In the supervised scenario, L_{CVAE} can encourage the network to learn a complementary representation of geometry and appearance as demonstrated in [2]. However, in our unsupervised scenario, there is no guarantee that $E^g_\cdot(;\pi)$ can distill effective geometry information. Inspired by the recent development of unsupervised landmark detection [15, 14, 16, 5], we introduce a prior loss to constrain the learning of geometry estimator:

$$\mathcal{L}\text{prior} = \sum_{i\neq j} \exp(-\frac{||g^i - g^j||^2}{2\sigma^2}) + \text{Var}(g) \tag{4}$$

where g^i is the $i-$th channel of the heatmap g, σ is a normalization factor and set to be 2 in our experiments. The first term is a *Separation Loss*, which encourages each heatmap to sufficiently cover the object of interest. The second term is a *Concentration Loss*, which encourages the variance of activations g to be small so that it could concentrate at a single location. Together with the prior loss $\mathcal{L}\text{prior}$, reconstruction loss in $\mathcal{L}_{\text{CVAE}}$ and equivariance constraint [16, 5], we can learn reasonable and effective landmarks unsupervisedly, which is necessary for the learning of disentanglement of geometry and appearance. As shown in Fig. 1, the networks in the separate domain (X/Y) can be trained end-to-end with $\mathcal{L}_{\text{separ}}$.

Joint-training. Once the networks in Fig. 1 are trained separately within X/Y domain, the model yet has the ability of disentanglement within domain. However, it so far does not have the ability of transformation and pose-preserving. Thus, we further train the model across-domain jointly. As shown in Fig. 2, the networks with dotted borders are frozen (*i.e.*, E^g_x, E^c_x, E^a_x, E^g_y, E^c_y and E^a_y) and other networks are trained end-to-end.

We train four transformers (*i.e.*, $\Phi^a_{x\to y}$, $\Phi^a_{y\to x}$, $\Phi^g_{x\to y}$, $\Phi^g_{y\to x}$) for appearance and geometry transformation respectively. For appearance transformation, transformers are defined in the appearance latent code $a_\cdot$. However, for geometry transformation, instead of defining transformers in the geometry latent code $c_\cdot$, we propose to transform geometry cues in the well-defined landmark heatmaps $g_\cdot$ for their pure-distilled geometry information. As discussed in Sec. 3.4 in the main paper, since we found it is hard to learn a transfer between unsupervised learned geometry heatmaps $g_\cdot$ directly, we perform geometry transformation in the landmark coordinates space. To this end, we extract the coordinates of each landmark from the heatmaps

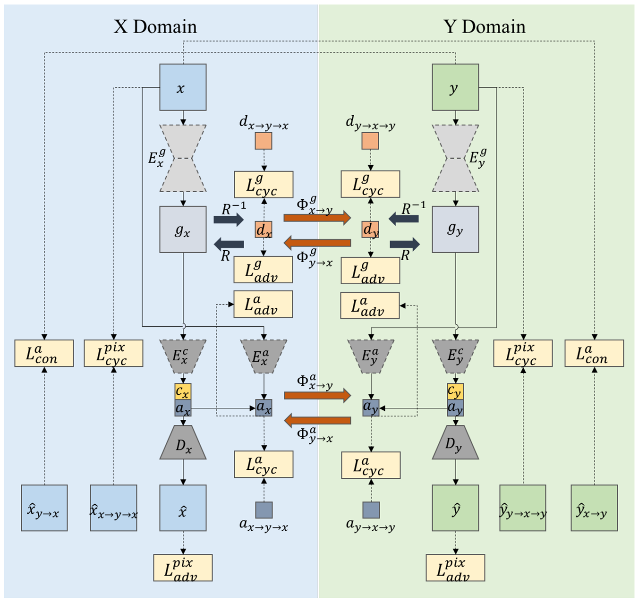

Figure 2: **Joint training.** $x \in \mathbb{R}^{256\times256\times3}$: input image. $g_x \in \mathbb{R}^{256\times256\times30}$: landmark heatmaps. $d_x \in \mathbb{R}^{1\times60}$: landmark coordinates. $c_x \in \mathbb{R}^{1\times256}$: geometry latent code. $a_x \in \mathbb{R}^{1\times256}$: appearance latent code. $\hat{x} \in \mathbb{R}^{256\times256\times3}$: generated image. E^g_x: geometry estimator. E^c_x: geometry encoder. E^a_x: appearance encoder D_x: decoder for X domain. $\Phi^a_{x\to y}$: appearance transformer. $\Phi^g_{x\to y}$: geometry transformer. Components are defined similarly in Y domain. R^{-1}: re-normalisation operator for converting landmark heatmaps to landmark coordinates. R^{-1}: the inverse operation of R^{-1}.

directly with a re-normalisation operator R^{-1}. R^{-1} is a differentiable operator that can transfer landmark heatmaps $g_\cdot \in \mathbb{R}^{256\times256\times30}$ to landmark coordinates $d_\cdot \in \mathbb{R}^{1\times60}$ as defined in [5], where R can do the inverse operation. The joint loss can be formulated as:

$$\mathcal{L}_{\text{joint}} = \mathcal{L}^a_{\text{con}} + \mathcal{L}^a_{\text{cyc}} + \mathcal{L}^g_{\text{cyc}} + \mathcal{L}^{pix}_{\text{cyc}} \\ + \mathcal{L}^a_{\text{adv}} + \mathcal{L}^g_{\text{adv}} + \mathcal{L}^{pix}_{\text{adv}} \tag{5}$$

The first term is the proposed cross-domain appearance consistency loss. Other terms are cycle-consistency losses and adversarial losses corresponding to the appearance, geometry and pixel space respectively. We formulate the definition of all these terms below.

Cross-domain appearance consistency loss:

$$\mathcal{L}^a_{\text{con}} = \|\zeta(x) - \zeta(D_y(c_{x\to y}, a_{x\to y}))\| \\ + \|\zeta(y) - \zeta(D_x(c_{y\to x}, a_{y\to x}))\| \tag{6}$$

where ζ is the *Gram matrix* [4, 6] calculated with a pre-trained VGG-16 [12] network, $c_{x\to y}$ equals to $E^c_y(\Phi^g_{x\to y}(g_x))$, which is the geometry code transformed from X to Y, $a_{x\to y}$ equals to $\Phi^a_{x\to y}(a_x)$, which is the appearance code transformed from X to Y, and $D_y(,)$ refers to decoder of Y domain.

Cycle-consistency loss in appearance space:

$$\begin{aligned}\mathcal{L}^a_{\text{cyc}} =& \mathbb{E}_{x\sim X}[||a_{x\to y\to x} - a_x||_1] \\ &+ \mathbb{E}_{y\sim Y}[||a_{y\to x\to y} - a_y||_1]. \end{aligned} \quad (7)$$

where $a_{x\to y\to x}$ equals to $\Phi^a_{y\to x}(\Phi^a_{x\to y}(a_x))$ and $a_{y\to x\to y}$ equals to $\Phi^a_{x\to y}(\Phi^a_{y\to x}(a_y))$.

Cycle-consistency loss in geometry space:

$$\begin{aligned}\mathcal{L}^g_{\text{cyc}} =& \mathbb{E}_{x\sim X}[||d_{x\to y\to x} - d_x||_1] \\ &+ \mathbb{E}_{y\sim Y}[||d_{y\to x\to y} - d_y||_1]. \end{aligned} \quad (8)$$

where $d_{x\to y\to x}$ equals to $\Phi^g_{y\to x}(\Phi^g_{x\to y}(d_x))$ and $d_{y\to x\to y}$ equals to $\Phi^g_{x\to y}(\Phi^g_{y\to x}(d_y))$.

Cycle-consistency loss in pixel space:

$$\begin{aligned}\mathcal{L}^{pix}_{\text{cyc}} =& \mathbb{E}_{x\sim X}[||\hat{x}_{x\to y\to x} - x||_1] \\ &+ \mathbb{E}_{y\sim Y}[||\hat{y}_{y\to x\to y} - y||_1] \end{aligned} \quad (9)$$

where $\hat{x}_{x\to y\to x}$ equals to $D_x(E^c_x(g_{x\to y\to x}), a_x)$ and $\hat{y}_{y\to x\to y}$ equals to $D_y(E^c_y(g_{y\to x\to y})$. $g_{x\to y\to x}$ is the landmark heatmaps calculated with $R(d_{x\to y\to x})$ and similarly $g_{y\to x\to y}$ is calculated with $R(d_{y\to x\to y})$.

Adversarial loss in appearance space of domain Y:

$$\begin{aligned}\mathcal{L}^a_{\text{adv}} =& \mathbb{E}_{a_x\sim q(a_x)}[\log(1 - S^a_y(\Phi^a_{x\to y}(a_x)))] \\ &+ \mathbb{E}_{a_y\sim q(a_y)}[\log(S^a_y(a_y))] \end{aligned} \quad (10)$$

where S^a_y is a discriminator that tries to distinguish between translated appearance latent codes and real appearance latent codes in Y domain. The discriminator S^a_x and the adversarial loss are defined similarly in domain X.

Adversarial loss in geometry space of domain Y:

$$\begin{aligned}\mathcal{L}^g_{\text{adv}} =& \mathbb{E}_{d_x\sim p(d_x)}[\log(1 - S^g_y(\Phi^g_{x\to y}(d_x)))] \\ &+ \mathbb{E}_{d_y\sim p(d_y)}[\log(S^g_y(d_y))] \end{aligned} \quad (11)$$

where S^g_y is a discriminator that tries to distinguish between translated landmark coordinates and real landmark coordinates in Y domain. The discriminator S^g_x and the adversarial loss are defined similarly in domain X.

Adversarial loss in pixel space of domain Y:

$$\begin{aligned}\mathcal{L}^{pix}_{\text{adv}} =& \mathbb{E}_{d_x\sim p(d_x), a_y\sim q(a_y)}[\log(1 - S^{pix}_y(D_y(c_{x\to y}, a_y)))] \\ &+ \mathbb{E}_{y\sim Y}[\log(S^{pix}_y(y))] \end{aligned} \quad (12)$$

where $c_{x\to y}$ equals to $E^c_y(R(\Phi^g_{x\to y}(d_x)))$, S^{pix}_y is a discriminator that tries to distinguish between translated images and real images in Y domain. The discriminator S^{pix}_x and the adversarial loss are defined similarly in domain X.

Total loss. Combining the losses in separated-training and joint-training, the full loss function of our method can be defined as:

$$\begin{aligned}\mathcal{L}_{\text{total}} =& \lambda_0\mathcal{L}_{\text{CVAE}} + \lambda_1\mathcal{L}_{\text{prior}} + \lambda_2\mathcal{L}^a_{\text{con}} + \lambda_3\mathcal{L}^a_{\text{cyc}} \\ &+ \lambda_4\mathcal{L}^g_{\text{cyc}} + \lambda_5\mathcal{L}^{pix}_{\text{cyc}} + \lambda_6\mathcal{L}^a_{\text{adv}} + \lambda_7\mathcal{L}^g_{\text{adv}} + \lambda_8\mathcal{L}^{pix}_{\text{adv}} \end{aligned} \quad (13)$$

We first perform separated-training for 40 epochs. Then, joint-training is done for 20 epochs. In all experiments, we use a batch size of 8 and set the loss weights to $\lambda_0 = 10$, $\lambda_1 = 1$, $\lambda_2 = 1$, $\lambda_3 = 0.1$, $\lambda_4 = 0.1$, $\lambda_5 = 1$, $\lambda_6 = 0.1$, $\lambda_7 = 0.1$, $\lambda_8 = 1$. We train all of the models use the Adam [7] optimizer with $(\beta_1, \beta_2) = (0.5, 0.999)$ and an initial learning rate of 0.0001. The learning rate is decreased by half every $100,000$ iterations.

2. Network Architecture Details

We use Stack-Hourglass network [10] for the geometry estimator E^g_x. For the mapping from g_x to $\hat{x}$ (E^c_x and D_x with skip-connection), we use the UNet architecture [11] provided by [17]. A same architecture of E^c_x is adopted for the appearance encoder E^a_x. The details of E^a_x (E^c_x) are shown in Fig. 3. The details of the decoder D_x is shown in Fig. 4. For the transformer $\Phi^a_{x\leftrightarrow y}$ ($\Phi^g_{x\leftrightarrow y}$) and the discriminator S^a_x (S^g_x), we use a simple 4-layer fully-connection network followed with ReLU, as shown in Fig. 5. Note that for $\Phi^g_\cdot$, the input (output) dimension is 16, rather than 60 for the use of PCA embedding. For pixel level adversarial loss, we use the discriminator provided by [9]. Architectures are defined same in Y domain.

Type	Kernel Size	Output Channels	Output Size
Input	N/A	3	256
Convolution	4	64	128
LeaklyReLU+Conv+IN	4	128	64
LeaklyReLU+Conv+IN	4	256	32
LeaklyReLU+Conv+IN	4	256	16
LeaklyReLU+Conv+IN	4	256	8
LeaklyReLU+Conv+IN	4	256	4
LeaklyReLU+Conv+IN	4	256	2
LeaklyReLU+Conv+IN	4	256	1
Conv	1	256	1

Figure 3: **Architecture details.** Architecture of encoder $E^a_\cdot$ and $E^c_\cdot$.

Type	Kernel Size	Output Channels	Output Size
Conv	1	256	1
LeaklyReLU+DeConv+IN	4	256	2
LeaklyReLU+DeConv+IN	4	256	4
LeaklyReLU+DeConv+IN	4	256	8
LeaklyReLU+DeConv+IN	4	256	16
LeaklyReLU+DeConv+IN	4	256	32
LeaklyReLU+DeConv+IN	4	128	64
LeaklyReLU+DeConv+IN	4	64	128
LeaklyReLU+DeConv+Tanh	4	3	256

Figure 4: **Architecture details.** Architecture of encoder $D_\cdot$.

Type	Output Channels	Output Size
Input	256	1
Fully-Connection+ReLU	512	1
Fully-Connection+ReLU	1024	1
Fully-Connection+ReLU	512	1
Fully-Connection	256	1

(a) Appearance transformer Φ^a and discriminator S^a

Type	Output Channels	Output Size
Input	16	1
Fully-Connection+ReLU	64	1
Fully-Connection+ReLU	256	1
Fully-Connection+ReLU	64	1
Fully-Connection	16	1

(b) Geometry transformer $\Phi^g$and discriminator S^g

Figure 5: **Architecture details.** Architecture of $\Phi^a_\cdot$, $S^a_\cdot$, $\Phi^g_\cdot$ and $S^g_\cdot$.

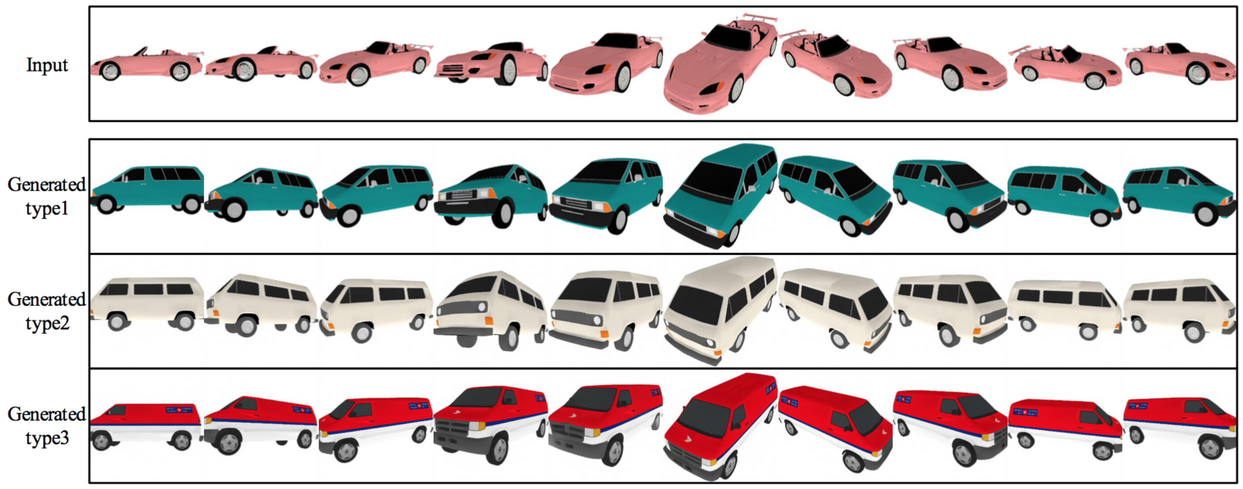

Figure 6: **Qualitative results.** Results of rendered sports car $\rightarrow$ commercial vehicle task.

3. More Qualitative Results

We demonstrate more results of our experiments on the rendered bear, wolf [18], sports car and commercial vehicle [3], which was not shown in the main paper due to the space limit. In Fig. 7, we show the task of bear $\leftrightarrow$ wolf. In Fig. 6, we show the task of sports car $\rightarrow$ commercial vehicle. Each sports car in different pose is translated to three types of commercial vehicles respectively by taking corresponding appearance references.

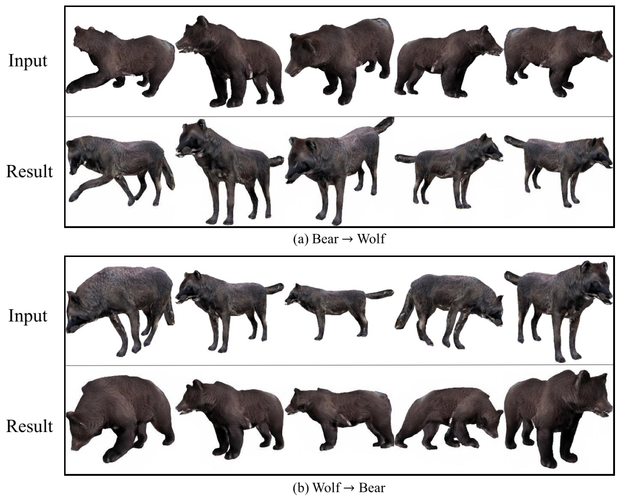

Figure 7: **Qualitative results.** Results of rendered bear $\leftrightarrow$ wolf task.

4. Limitation

Although our method can achieve compelling results in many translation tasks with large domain geometry variations, the results are far from uniformly positive in some extremely unconstrained and noisy scenarios. To push the limits of our method, we collect several categories of objects in ImageNet [1] and COCO [8] dataset, *e.g.*, car and bus, cow and horse, *etc*. These images from in-the-wild datasets are extremely unconstrained for the large appearance variations together with large geometry variations. For example, images of bus have different views, scales and appearance and some buses are even part-missing. Fig. 8 shows several typical failure cases. We observed that the main cause of the failure of our method is lying on the failure of unsupervised geometry estimator. Handling translation in extremely unconstrained scenarios is an important problem for future work.

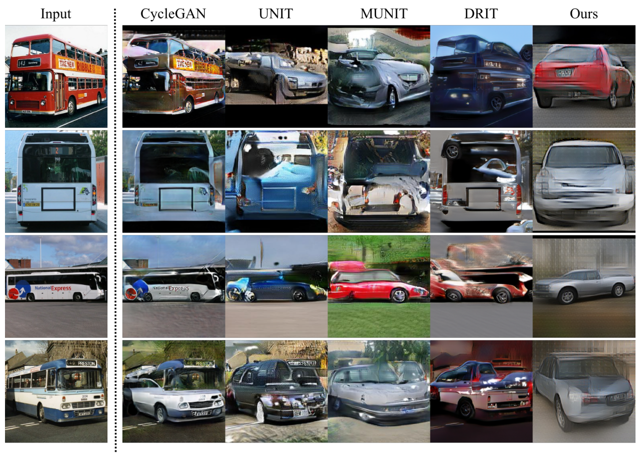

Figure 8: **Limitation.** Failure cases in bus $\rightarrow$ car task on ImageNet.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[2] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. In *CVPR*, 2018.

[3] Sanja Fidler, Sven J. Dickinson, and Raquel Urtasun. 3d object detection and viewpoint estimation with a deformable 3d cuboid model. In *NIPS*, 2012.

[4] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *NIPS*, 2015.

[5] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Conditional image generation for learning the structure of visual objects. In *NeurIPS*, 2018.

[6] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[7] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[8] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014.

[9] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *NIPS*, 2017.

[10] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[12] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[13] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *NIPS*, 2015.

[14] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object frames by dense equivariant image labelling. In *NIPS*, 2017.

[15] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *ICCV*, 2017.

[16] Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee. Unsupervised discovery of object landmarks as structural representations. In *CVPR*, 2018.

[17] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *ICCV*, 2017.

[18] Silvia Zuffi, Angjoo Kanazawa, and Michael J. Black. Lions and tigers and bears: Capturing non-rigid, 3d, articulated shape from images. In *CVPR*, 2018.