Multi-digit Image Synthesis Using Recurrent Conditional Variational Autoencoder

Haoze Sun, Weidi Xu, Chao Deng and Ying Tan
Key Laboratory of Machine Perception(MOE), Peking University
Department of Machine Intelligence, School of Electronics
Engineering and Computer Science, Peking University, Beijing, 100871, China
Email: pkucissun@foxmail.com, {wead_hsu, cdspace678, ytan}@pku.edu.cn

***Abstract*—In the field of deep neural networks, several generative methods have been proposed to address the challenges from generative and discriminative tasks, e.g., natural language process, image caption and image generation. In this paper, a conditional recurrent variational autoencoder is proposed for multi-digit image synthesis. This model is capable of generating multi-digit images from the given number sequences and retaining the generalisation ability to recover different types of background. Our method is evaluated on SVHN dataset and the experimental results show it succeeds to generate multi-digit images with various styles according to the given sequential inputs. The generated images can also be easily identified by both human beings and convolutional neural networks for digit classification.**

I. Introduction

Deep neural networks have seen huge progress in several major domains, e.g., computer vision [9], speech recognition [8] and natural language process [2]. However most of those models are discriminative and need lots of labelled data. In recent years, several advanced deep generative models have been proposed, e.g., restricted boltzmann machines (RBMs) [9], variational autoencoders (VAEs) [7, 12, 13, 17] and generative adversarial networks (GANs) [4, 18]. These models greatly improved the performance of deep generative networks.

In contrast to conventional generative models, conditional generative models can generate data while keeping several certain attributions of given labels (e.g., object category, colour). Kingma et al.[13, 20] proposed a conditional variational autoencoder to successfully separate the image style and content information. Springenberg et al. [18] proposed a categorical generative model based on generative adversarial network. This approach is based on an objective function that trades-off mutual information between observed samples and their prediction. These methods can generate particular data samples with given input.

Conditional generative models are somewhat related to multi-modal models. Typically, given a meaningful images, several image-caption models [11, 14, 19] can generate general captions conditioned on given images. These models are extensively studied over computer vision and natural language process communities.

However, although powerful generative models are represented, image generation method conditioned on sequential labels has been out of sight. Recently proposed methods, e.g., conditional variational autoencoders [13, 20] and CatGAN [18], originally deal with simple single categorical input but can not generate image data conditioned on sequential input. This motivates us to build a model that is capable of generating general images given label sequence.

The aim of this paper is to generate SVHN-like multi-digit images using sequential number labels. *Caption-image* model [15] was recently proposed to generate general images given captions. This *caption-image* model is modified it for multi-digit image synthesis. The model receives label information using attention mechanism [1] at each time-step. Hence the drawing model can modify image canvas by different numbers iteratively. Attention mechanism allows it to choose where to focus on during the generation process. Useful information, i.e., selected number is passed to conditional DRAW model while others remain unused.

The contributions of this paper are:

1) A new conditional generation framework is proposed for multi-digit image synthesis.
2) Different from [15], the recurrent encoding layer for sequential labeled inputs is replaced with an attention-based model to select raw input features, which is found better for multi-digit image synthesis.

The rest of the paper is organized as follows. In section II, we introduce the related works. In section III, our conditional generative model is presented in detail. In section IV we obtain both quantitive results and qualitative analysis. In section V, we conclude the paper with a discussion.

II. Related Works

This section introduces several related deep generated models, i.e., restricted boltzmann machines, variational autoencoders and generative adversarial networks and their extensions. Variational autoencoder and its extensions will be descirbed in detail.

A. Deep Generative Models

Restricted Boltzmann Machine (RBM) is one of the most successfully models in deep neural learning field. It is widely used in various applications, however it suffers from costly posterior distribution inference which needs to take expensive MCMC steps. Generative adversarial network (GAN) is another novel model for image generation tasks. The model

utilise two networks to compete with each other: one for image generation and the other tries to tell if the image is generated by the first network or from dataset.

Recently, variaional autoencoder (VAE) have drawn a lot of attentions due to its impressive results reported in [12] and [7]. With a top-down generative network and a bottom-up recognition network, the model is trained to maximize the variational lower bound of data likelihood.

B. Conditional Variational Autoencoder

Since standard variational autoencodes[12] are pure generative models, many looked for conditional generative models. Kingma et al. [13] proposed a conditional generative model which can utilise label feature and generate images with certain characteristics. Specifically it adds another label factor into probabilistic graphical model and reformulates the variational lower bound with this additional factor. Kingma et al. [13] firstly proposed this conditional variational autoencoder and Yan et al. [20] extended it with a powerful convolutional neural network for complex image generation. Both works show the effectiveness of variational autoencoders.

C. Recurrent Variational Autoencoder

Considering that single step generation is difficult in handling various hidden information, Gregor et al. [7] extended the conventional variational autoencoder structure to recurrent form. Their model draws a picture with multiple steps of modification. DRAW network converts both encoder and decoder of conventional VAE into recurrent networks so that it can handle long hidden variable sequence. In short, DRAW decides at each time-step "where to read" and "where to write" as well as "what to write". With help of dynamic spatial attention mechanism this model can achieve the best generative performance so far on mnist dataset.

While many approaches have been proposed for unconditional generation or simple category-based conditional generation, in this paper we propose an image generative model conditioned on given sequential labels.

III. Conditional Generative Model

The model is divided into two main parts due to its complexity: 1) a variational autoencoder network whose basic structure is similar with Conditional DRAW network [15]. 2) an attention-based sequence processing network. The architecture is sketched in figure 1.

A. Conditional DRAW Network

At first, some basic conceptions of conditional variational autoencoder (CVAE) is described and then we present conditional DRAW network, a recurrent form of CVAE. In this paper, we use the notation $b = W(a)$ to denote a linear weight matrix with bias from vector a to vector b for simplicity .

Given dataset $X = \{x_1, x_2, ..., x_N\}$, the variational autoencoder aims to maximize the loglikelihood of all datapoints, the following variational lower bound is optimized:

$$\log p_\theta(x) \geq E_{q_\phi}[\log p_\theta(x|z)] - KL[q_\phi(z|x)||p_\theta(z)] \quad (1)$$

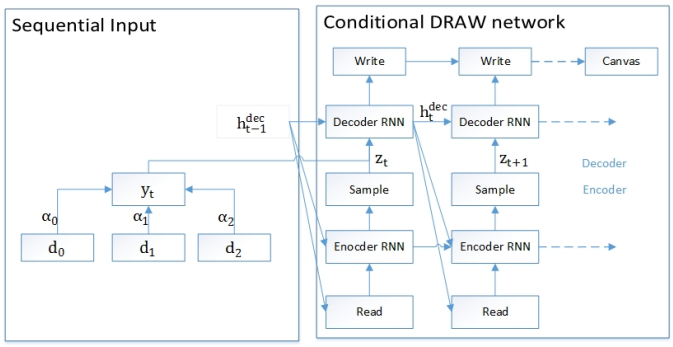

Fig. 1. **Left**: This is the sketch of sequential input structure. Sequential input represented as vectors is imported and the output is weighted over all elements in sequence using attention method. **Right**: 1) At each time-step data is encoded by enocoder RNN 2) A sample z_t from prior $p(z_t|z_{1:t-1}, y_{1:t-1})$ is passed to recurrent decoder nework, which modifies part of the canvas matrix. The output of decoder RNN computes the approximate posterior over $z_{1:T}$ and $y_{1:T}$.

The first item performs reconstruction approximation while the second one acts as a regulariser.

In the context of conditional model, additional input y is given and hence this equation simply extends to equation 2.

$$\log p_\theta(x|y) \geq E_{q_\phi}(\log p_\theta(x|z, y)) - D_{KL}(q_\phi(z|x, y)||p_\theta(z|y)) \quad (2)$$

where inference model q_ϕ and ground prior is conditioned on given input y.

In original DRAW network hidden variable z_t at each step is sampled from standard Gaussian $\mathcal{N}(0, I)$. However in conditional DRAW network this distribution is transformed to base on previous hidden variables and input y_t. Since previous hidden variable h^{dec}_{t-1} in recurrent decoder contains all information from previous z_t and y_t, the mean and variance of this prior distribution over z_t are given by:

$$p(z_t|z_{1:t-1}, y_{1:t-1}) = \mathcal{N}(\mu(h^{dec}_{t-1}), \sigma(h^{dec}_{t-1})) \quad (3)$$

$$\mu(h^{dec}_{t-1}) = W_\mu(h^{dec}_{t-1}) \quad (4)$$

$$\sigma(h^{dec}_{t-1}) = exp(W_\sigma(h^{dec}_{t-1})) \quad (5)$$

Where $W_\mu \in R^{n\times m}, W_\sigma \in R^{n\times m}$ are parameters learned during training, n is the dimension of vector h^{dec}_{t-1} and m is the dimension of $\mathcal{N}$.

In generation process canvas c_t is gradually modified at each time-step by sequence z_t and y_t. Given initial h^{dec}_0 and $y_{1:T}$, generation goes as:

$$z_t \sim p(z_t|z_{1:t-1}, y_{1:t-1}) = \mathcal{N}(\mu(h^{dec}_{t-1}), \sigma(h^{dec}_{t-1})) \quad (6)$$

$$h^{dec}_t = RNN^{dec}(h^{dec}_{t-1}, [z_t, y_t]) \quad (7)$$

$$c_t = c_{t-1} + write(h^{dec}_t) \quad (8)$$

All above are similar with DRAW network [7] except for additional input y_t and conditional priori distribution.

At each time step, we recompute h^{dec} using recurrent network according to equations above. Here we simply concatenates z_t and y_t as single-time input. z_t is sampled from prior distribution and y_t is generated from sequential input part which we will describe in detail in next sub-section III-B. In DRAW network all recurrent networks are long short-term memory network (LSTM) [3, 6]. LSTM network is capable of capturing long term dependency while mitigating annoying gradient vanishing problem occurring in traditional recurrent networks. Important information will be saved in memory units on the fly. Adopting LSTM network long-time-step drawing becomes possible.

The DRAW network carefully designed a method to dynamically write content to canvas at each step. In equation 8 $write$ function receives h_t^{dec} as input parameter and then outputs a five-dimension array specifying where to write. This array leads to the horizontal and vertical Gaussian filters F_X and F_Y and finally transforms generated image-patch $W_{write}(h_t^{dec})$ into current canvas c_{t-1}:

$$write(h_t^{dec}) = F_Y(h_t^{dec})W_{write}(h_t^{dec})F_X(h_t^{dec}) \tag{9}$$

After T time-steps final canvas c_T is generated. Finally generated image is given by:

$$x \sim p(x|y, z_{1:T}) = Bern(\sigma(c_T)) \tag{10}$$

The second part is inference network which models posterior distribution $q_\phi(z|x, y)$. Similar to DRAW network, the inference network produces an approximate posterior $q(z_{1:T}|x, y)$ where x is data to be generated and y is given labels. By using $read$ function the DRAW network is able to gradually read patches of image step by step.

$$\hat{x}_t = x - \sigma(c_{t-1}) \tag{11}$$

$$r_t = read(x_t, \hat{x}_t, h_{t-1}^{dec}) \tag{12}$$

$$h_t^{enc} = RNN^{enc}(h_{t-1}^{enc}, [r_t, h_{t-1}^{dec}]) \tag{13}$$

$$q(z_t|x, y, z_{1:t-1}) = \mathcal{N}(\mu(h_t^{enc}), \sigma(h_t^{enc})) \tag{14}$$

Where $\hat{x_t}$ is the error image to make up, h_t^{enc} is the hidden output of encoder recurrent network, h_t^{dec} is the hidden output of aforementioned decoder recurrent network.

Similarly, h_0^{enc} is the initial parameter to be learned during training. According to equations above h_t^{enc} is given by previous hidden state and r_t (generated using origin image x and error image $\hat{x}_t$) and previous hidden state from decoder network. Since h_{t-1}^{dec} depends on sampled variables $z_{1:t-1}$ and conditional input sequence $y_{1:t-1}$, h_t^{enc} depends on z and y which satisfies the required form $q_\phi(z|x, y)$ in equation 2. $read$ function acts like the inversion of $write$ function. While $write$ maps generated patch to the entire canvas $read$ tries to extract single patch both in x and $\hat{x}_t$ by generating two other Gaussian filters.

B. Attention-based Sequential Input

In this section, we present how sequential input is modelled to generate y_t at each time step. Here we give the network flexibility to choose which part of input sequence to focus on by itself. Given a input sequence $d = (d_0, d_1, ..., d_L)$ with length L and each element in d is a b dimension vector depicting digit label, y_t is produced by standard soft attention mechanism. This mechanism is carefully studied in [1] and is widely used in neural translation and image caption models [19]. The conditional input y_t is generated as follows.

$$y_t = \sum_{i=1}^{L} \alpha_{t,i} d_i \tag{15}$$

$$\alpha_{t,i} = \frac{exp(e_{t_i})}{\sum_{j=1}^{L} exp(e_{t_j})} \tag{16}$$

$$e_{t,i} = v_a^T tanh(W_a h_{t-1}^{dec} + U_a d_i) \tag{17}$$

Where v_a, W_a, U_a are parameters in soft attention model. Since $U_a d_i$ does not depend on t, we can pre-compute it to minimize the computational cost. Because h_{t-1}^{dec} represents the conditional DRAW network and carries necessary information about $z_{1:t-1}$ and $y_{1:t-1}$, we use it as a single factor for calculating attention weight α.

A simple alternative is to use recurrent network and provide the last hidden layer output to conditional DRAW network at each time-step. However in experiments we found it tends to remember the digits together and generate all digits simultaneously. Attention-based method deals this with ability to choose the conditional label dynamically.

C. Gradient-based Optimization for Variational Bounds

While conditional variational autoencoder tries to optimize the variational lower bound in equation 2, this model tries to optimize an objective function that consists of a sequence steps of regularization. Formally the equation extends to:

$$\begin{aligned} L =& E_{q(z_{1:T}|x,y)}(\log p(x|y, z_{1:T}) \\ & - \sum_{t=1}^{T} D_{KL}(q(z_t|z_{1:t-1}, x, y)||p(z_t|z_{1:t-1}, y))) \end{aligned} \tag{18}$$

Note that there is an expectation here and it is impossible to iterate all $z_{1:T}$ in practice, it is easier to sample several hidden variables and then perform stochastic gradient descend method directly:

$$\begin{aligned} L \approx& \frac{1}{L} \sum_{l=1}^{L} (\log p(x|y, z_{1:T}^l) \\ & - \sum_{t=1}^{T} D_{KL}(q(z_t|z_{1:t-1}^l, x, y)||p(z_t|z_{1:t-1}^l, y))) \end{aligned} \tag{19}$$

During the training, variable z is sampled with the reparameterization trick proposed in [12]. An auxiliary variable ϵ is first sampled from standard Gaussian distribution and

TABLE I
HYPER-PARAMETERS FOR OUR MODEL. SYMBOL # MEANS THE DIMENSION OF THE VECTOR.

parameters	LSTM #h	#z	#y	b	k
rnnDRAW	400	100	50	10	10
seqDRAW	400	100	20	20	10
seqDRAW-cur	400	100	20	20	10

$z \sim \mathcal{N}(\mu, \sigma^2)$ can be expressed as $z = \mu + \sigma\epsilon$. L times of sampling are used to approximate variable z:

$$E_{\mathcal{N}(\mu,\sigma^2)}[f(z)] \approx \frac{1}{L}\sum_{l=1}^{L} f(\mu + \sigma\epsilon) \tag{20}$$

Since all operations used here are differentiable and hence the objective function can be optimized using gradient-based methods. Detailed derivation about D_{KL} of two Gaussian distribution with different means and variations is stated in appendix VI.

D. Curriculum Learning

To deal with the variable length of sequences, the total number of time-step T should be different for each input sequence. We assume that shorter sequence implies less image complexity and hence needs fewer iteration steps during image generation. In the implementation, we simply set $T = kL$, where k is const. By using this setting and attention mechanism our model can generate digit canvas sequentially.

Curriculum learning is often used in some comprehension-related tasks when difficult concepts are hard to be inferred directly but can be obtained by gradually learning from simple to complex data [21, 22]. In our model we gradually increase the difficulty for generation. The dataset is first partitioned into subsets according to sequence length L. Then the model is trained with gradually increased data length, i.e., after training the subset of $L = 1$, the model is feed with the subset of $L = 2$ and the rest can be done in the same manner.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

Our model was trained with images from Street View House Numbers (SVHN) [16]. The SVHN dataset has 33k raw images for training and 13k images for testing. Each data sample has information about digit labels and digit bounding boxes (position and size). In training set, the percentage of images with 1 digit is about 16%, the percentage of images with 2 digits is about 54%, and 26% for length 3, the rest for length 4.

Resolution of original images are different from each other, so the original data is preprocessed into 32x64 size of grey images. Rather than stretching characters to fill the whole image as in [5] we first count the digits length and then carefully align each digit from right to left. Several preprocessed images are shown in figure 2. These images have various kinds of backgrounds and that increases the difficulty in generation tasks.

Fig. 2. Several data samples after pre-processing.

A. Quantitive Analysis

To demonstrate the generation ability of our model we analyse the reconstructed images from test dataset. One obvious measurement is the reconstruction likelihood of images. However this measurement has shown high variation in our experiments and is not able to intuitively show the quality of generated images. Hence we use one alternative measurement here, i.e., instead of demonstrating likelihood of dataset, a classification model is employed to obtain the generating precision. We trained a convolutional neural network for digit classification proposed by Goodfellow et al. [5]. This model jointly determines the digits length and the label of each digits. We used our preprocessed data to train this model and achieved about 78% accuracy in testset.

Here we compare the results of three models. For the baseline model, we simply replace the sequential input model with a recurrent network and use the output of last hidden layer as the input value for all time-steps in conditional DRAW network. This baseline model is denoted as *rnnDRAW*. The second model is described in section III but without the curriculum learning, namely *seqDraw*. The last one is the model with curriculum learning, i.e., *seqDRAW-cur*. The hyper-parameters for these models are listed in table I. All the models take $k = 10$ steps to reconstruct one digit. Note that in *seqDRAW* and *seqDRAW-cur* label vectors and $10d$ position vectors are concatenated together and hence results in $20d$ label vectors.

The classification results are shown in table II. Although the classification accuracy of images with more than two digits is not very high, it can be used to evaluate the performance. The low accuracy is somehow due to that 1) the generated images are vague. According to [10] variational autoencoders tend to generate vague images while generative adversarial nets do not. 2) the noise in images.

We can tell from table II that the *seqDRAW-cur* performs better comparing to other two models. This indicates that simple sequence input model with attention mechanism is capable of feeding conditional DRAW network with more useful information.

TABLE II
CLASSIFICATION ACCURACIES

length	1	2	3
rnnDRAW	81.25%	7.55%	6.60%
seqDRAW	81.02%	7.66%	6.50%
seqDRAW-cur	**81.65%**	**7.82%**	**6.80%**

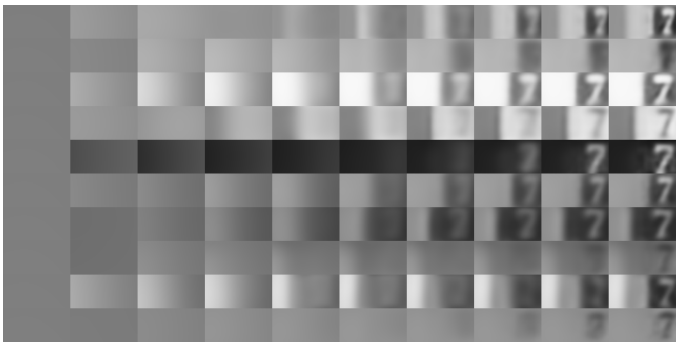

Fig. 3. Single-digit image generated given number 7 by *seqDRAW-cur*

B. Qualitative Analysis

Here we demonstrate several generation processes. We randomly sampled multi-digit several sequences with different length and then generate images conditioned on these sequences. Figures 3,4,5 are generated using *seqDRAW-cur* with sequence length L from 1 to 3. Each row represents one generation run for certain sequence and the images from left to right are sampled uniformly from all kL time-steps.

The generated images have various kinds of styles while having the same digit content. Since the hidden variable z_t is not deterministic in each run, the gray scale of backgrounds and digits are diverse. However the digits during each generation process keep the same gray scale. This indicates that our model is able to generate images with certain sequence while keeping some global features like image style.

Figure 6 is generated using *rnnDRAW*, it is more blurry and all of the digits come out almost simultaneously. This is because at each time-step, single recurrent network without attention mechanism tends to extract information from the whole given sequence but not focus on one digit. The results also show that the generalisation ability of *seqDRAW-cur* is better than *rnnDRAW*.

V. CONCLUSION

In this paper, a conditional generative model was proposed to address multi-digit image synthesis tasks. This model combines the conditional variational autoencoders and attention-based sequential inputs to achieve the conditional sequence generation ability. To deal with the difficulty arising from various input length, curriculum learning method is used to adaptively select the total time-steps for generation process. The evaluations on SVHN dataset well demonstrate that our model is able to generate multi-digit images conditioned on the given sequential inputs while having different backgrounds.

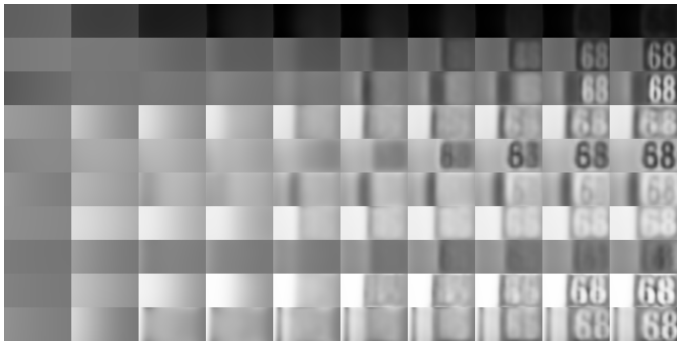

Fig. 4. Two-digits image generated given numbers 68 by *seqDRAW-cur*

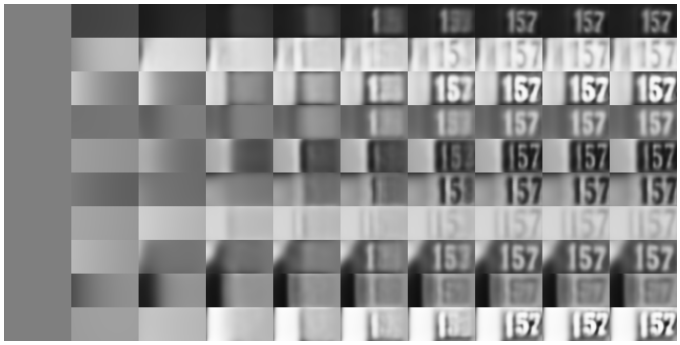

Fig. 5. Three-digits image generated given numbers 157 by *seqDRAW-cur*

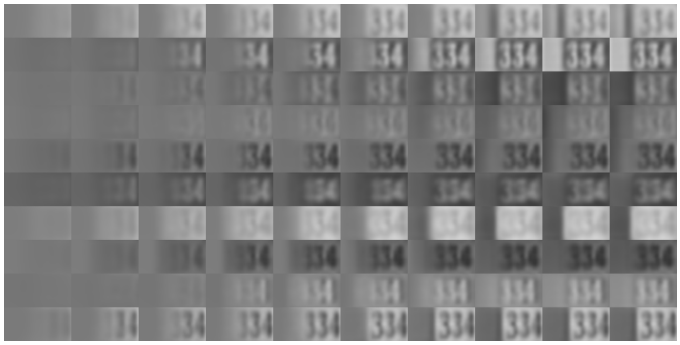

Fig. 6. Three-digits image generated given numbers 334 by *rnnDRAW*

ACKNOWLEDGMENT

This work was supported by National Key Basic Research Development Plan (973 Plan) Project of China under grant no. 2015CB352302, and partially supported by the Natural Science Foundation of China (NSFC) under grant no. 61375119 and Beijing Natural Science Foundation (4162029).

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model.

The Journal of Machine Learning Research, 3:1137–1155, 2003.

[3] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[5] Ian J Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks. *arXiv preprint arXiv:1312.6082*, 2013.

[6] Alex Graves et al. *Supervised sequence labelling with recurrent neural networks*, volume 385. Springer, 2012.

[7] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. DRAW: A recurrent neural network for image generation. In *ICML*, volume 37 of *JMLR Proceedings*, pages 1462–1471. JMLR.org, 2015.

[8] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29 (6):82–97, 2012.

[9] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[10] Ferenc Huszár. How (not) to train your generative model: Scheduled sampling, likelihood, adversary? *arXiv preprint arXiv:1511.05101*, 2015.

[11] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *arXiv preprint arXiv:1412.2306*, 2014.

[12] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[13] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[14] Lin Ma, Zhengdong Lu, Lifeng Shang, and Hang Li. Multimodal convolutional neural networks for matching image and sentence. *arXiv preprint arXiv:1504.06063*, 2015.

[15] Elman Mansimov, Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Generating images from captions with attention. *arXiv preprint arXiv:1511.02793*, 2015.

[16] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5. Granada, Spain, 2011.

[17] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pages 3465–3473, 2015.

[18] Jost Tobias Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390*, 2015.

[19] Kelvin Xu, Jimmy Ba, Ryan Kiros, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2015.

[20] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. Attribute2image: Conditional image generation from visual attributes. *arXiv preprint arXiv:1512.00570*, 2015.

[21] Jimei Yang, Scott E Reed, Ming-Hsuan Yang, and Honglak Lee. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In *Advances in Neural Information Processing Systems*, pages 1099–1107, 2015.

[22] Wojciech Zaremba and Ilya Sutskever. Learning to execute. *arXiv preprint arXiv:1410.4615*, 2014.

VI. Appendix

The variational lower bound contains KL term which can be integrated analytically. Here we give the solution when the posterior $q_\phi(z|\cdot) = \mathcal{N}(\mu_1, \sigma_1^2)$ and the prior $p_\theta(z|\cdot) = \mathcal{N}(\mu_2, \sigma_2^2)$. Let J be the dimensionality of z.

$$\begin{aligned}\int q_\phi(z) \log p_\theta(z) dz &= \int \mathcal{N}(z; \mu_1, \sigma_1^2) \log \mathcal{N}(z; \mu_2, \sigma_2^2) dz \\ &= -\frac{1}{2} \sum_j \log(2\pi\sigma_2^2) - \\ &\quad \frac{1}{2} \sum_j ((\mu_2 - \mu_1)^2 + \frac{\sigma_1^2}{\sigma_2^2})\end{aligned}$$

And:

$$\begin{aligned}\int q_\phi(z) \log q_\phi(z) dz &= \int \mathcal{N}(z; \mu_1, \sigma_1^2) \log \mathcal{N}(z; \mu_1, \sigma_1^2) dz \\ &= -\frac{1}{2} \sum_j \log(2\pi\sigma_1^2) - \frac{J}{2}\end{aligned}$$

Therefor:

$$\begin{aligned}D_{KL}(q\phi(z|\cdot)||p_\theta(z|\cdot)) =& \frac{1}{2}(\sum_j (\mu_1 - \mu_2)^2 \\ &+ \sum_j (\sigma_1/\sigma_2)^2 - \sum_j \log(\sigma_1/\sigma_2)^2 - J)\end{aligned}$$