Supplementary Material: Channel-Recurrent Autoencoding for Image Modeling

Wenling Shang
University of Amsterdam
wshang@uva.nl

Kihyuk Sohn
NEC Labs
ksohn@nec-labs.com

Yuandong Tian
Facebook AI Research
yuandong@fb.com

A. Mathematical Intuition on Model Design

We provide more details on the mathematics that motivate our proposed channel-recurrent architecture. Particularly, our architecture attempts to imitate the effects of having a non-diagonal multivariate Gaussian approximated posterior and prior, which is too computationally expensive to achieve exactly in practice when the latent dimension is big [3]. In practice, to draw a sample from a non-diagonal multivariate Gaussian, one first draw from the standard multivariate Gaussian and operate the following procedures are performed:

$$z = A\epsilon + \mu,\ AA^\top = \Sigma, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

Therefore, to imitate the variance transformation via multiplication of A, we apply an LSTM layer on the variance inference path to allow all latent variables to communicate with one another and similarly to imitate the inverse of that, we apply another LSTM layer on the generation path. To imitate the mean addition, on the mean inference path, we integrate an element-wise addition layer and on the generation path, we add another element-wise addition layer after the LSTM layer.

B. Ablation Studies on MNIST

We ablate our model architecture on MNIST, by comparing models including standard VAE, cVAE, crVAEs with channel recurrency on the inference path only, the generation path only and on both paths. We also conduct control experiments to search the optimal number of recurrent channels on crVAE.

The negative log-likelihood (NLL) of MNIST has been used as a benchmark to evaluate probabilistic generative models. NLL is approximated via importance sampling [13], i.e. for each image from the statistically binarized MNIST test set [9], we sample from its approximated posterior 10K times. We keep the same latent dimension for all the models. The results obtained with our and a number of previously proposed generative models are summarized in Table 1. The baseline, standard VAE [8], obtains 82.04 NLL. As expected, cVAE shows worse performance (83.12), because it can not well capture globally coherent patterns. Among crVAE models, positioning the recurrency on either the inference path or the generation path marginally improves upon the standard VAE and cVAE. The most substantial improvement happens when the recurrency is used both for inference and generation paths, achieving 80.84 NLL.

To assess the impact of the number of recurrent time steps, we train crVAEs with $T = 4, 8, 16$ while maintaining other hyperparameters. Table 1 shows that $T = 8$ achieves the best NLL. We conjecture that including too many latent channels at one step ($T = 4$) burdens the recurrent layers to establish meaningful intra-block connections, whereas too few latent channels ($T = 16$) limits the expressive power of the LSTM.

Our crVAE is generalizable and compatible with many other advanced training methods of directed graphical models. For example, by integrating k-sample importance weighted estimation [1], we can improve the NLL to 80.02; a hierarchical crVAE with 2 stochastic layers reduces NLL to 79.65

C. Details of Network Architecture

C.1. Stage1 Generation

Please refer to the paper repository (https://github.com/WendyShang/crVAE) for Stage1 model architecture details.

Model	specifications	NLL	Model	specifications	NLL
VAE	–	82.04	crVAE	$T = 4$	81.09
cVAE	–	83.12	crVAE	$T = 4$	80.84
crVAE	@inference	81.86	crVAE	$T = 16$	81.35
crVAE	@generation	81.71	crIWAE	–	80.02
crVAE	both	80.84	crVAE	2 stochastic layers	79.65

Table 1: Ablation studies on MNIST comparing VAE, cVAE, and crVAE for different configurations.

C.2. Stage2 for Birds

The Stage2 generation network for Birds follows similar architecture as in [15], but without the text encoding parts. We have uploaded the pretrained Stage2 Birds models to the repository.

C.3. Stage2 for CelebA

For Stage2 generation of crVAE-GAN on CelebA dataset, we found the original stackGAN [15] architecture does not perform well. Instead, we replace nearest neighbor upsampling followed by 3×3 convolution with a deconvolution layer, i.e. spatial full convolution of kernel 4×4 and stride 2, padding 1. In the case of Stage2 generation of VAE-GAN, we found that the generation performance is sensitive to the generator architecture and we only reduce the spatial dimension once from 64×64 to 32×32 instead of downsampling to spatial dimension 16×16.

We also have uploaded the pretrained Stage2 CelebA models to the repository.

C.4. Stage2 for LSUN

We integrated more modifications over the model from [15] to generate 224×224 LSUN bedroom images and uploaded the pretrained LSUN model to the repository.

D. Implementation Details

Modification of VAE-GAN objective from [11]. We empirically found that reconstructing the last convolutional layer of the discriminator to maximize $\mathbb{E}_{q_\phi(z|x)}\big[\log p_\theta(x|z)\big]$ as proposed in [11] leads to more instability in training comparing to reconstructing at the pixel-level. Hence, throughout our work, we reconstruct at the pixel level for both VAE-GAN and crVAE-GAN.

Data Augmentation. For Birds and CelebA, random horizontal flipping is used during training as data augmentation. For LSUN, random cropping and horizontal flipping are used.

Optimization. Our models are optimized with ADAM [6], where we set $\epsilon = 1 \times 10^{-8}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$. For the discriminator, we use a variation of ADAM with additional thresholding [10]: if the classification accuracy of the discriminator for a batch consisting of half real and half fake images is over 90%, we do not update the model parameters of the discriminator.

Initial Learning Rate. For Stage1 models, Birds has initial learning rate 0.0003, CelebA 0.003 for VAE-GAN and 0.001 for crVAE-GAN, LSUN 0.0003 for VAE-GAN and 0.0001 for crVAE-GAN. For all Stage2 networks, we use 0.0002 as initial learning rate.

Objectives. Recall the VAE objective:

$$\mathcal{L}_{\text{VAE}} = \underbrace{-\mathbb{E}_{q_\phi(z|x)}\big[\log p_\theta(x|z)\big]}_{\mathcal{L}_{\text{recon}}} + \underbrace{D_{\text{KL}}(q_\phi(z|x)\|p(z))}_{\mathcal{L}_{\text{prior}}} \geq -\log(p(x)). \tag{1}$$

Conventionally, for RGB images, $p_\theta(x|z)$, a diagonal multivariate Gaussian, is assumed to have fixed variance σ and only the mean values are estimated which correspond to pixel values. We follow this convention and apply MSE loss at the pixel level, which we find effective even without feature-level reconstruction, and defer σ to be adjusted through weighting the $\mathcal{L}_{\text{prior}}$, i.e. the KL-divergence term, by introducing an additional hyperparamter α. Recall that we weigh the KL-divergence term

differently for different time steps when training crVAE, i.e. for the first 3 time steps with α_1 and the rest α_2. Moreover, the adversarial loss is weighted by hyperparameter β. Together, we have the following training objective for RGB images:

$$\mathcal{L}_{\text{VAE-GAN}} = \mathcal{L}_{\text{recon}} + \alpha\mathcal{L}_{\text{prior}} + \beta\mathcal{L}_{\text{D}}, \tag{2}$$

where

$$\mathcal{L}_{\text{D}} = -\log(D(x)) - \log(1 - D(\text{gen}(z))), z \sim p(z), \text{ or } z \sim q_\phi(z|x).$$

It is worth mentioning that the generated samples, which are decoded from z sampled from the prior $p(z)$, and the reconstructed samples, which are decoded from z sampled from the approximated posterior $q_\phi(z|x)$, are both used for training with adversarial loss. Specifically, a mini-batch for discriminator update is assembled with 50% real examples (half), 25% generated samples (a quarter) and 25% reconstructed samples. Also, note that $\mathcal{L}_{\text{recon}}$ is divided not only by the batch size, but also by the channel, width and height of the images for implementation convenience, while $\mathcal{L}_{\text{prior}}$ or $\mathcal{L}_{\text{D}}$ are divided by the batch size only. This explains why the optimal α or β values are considerably smaller than one.

α and β For Birds, VAE-GAN is trained with $\alpha = 0.0002$ and $\beta = 0.0125$ and crVAE-GAN $\alpha_1 = 0.0003$, $\alpha_2 = 0.0002$ and $\beta = 0.0125$; for CelebA, VAE-GAN is trained with $\alpha = 0.0003$ and $\beta = 0.01$ and crVAE-GAN $\alpha_1 = 0.0003$, $\alpha_2 = 0.0002$ and $\beta = 0.01$; for LSUN, VAE-GAN is trained with $\alpha = 0.0002$, $\beta = 0.025$ and crVAE-GAN $\alpha_1 = 0.0003$, $\alpha_2 = 0.0002$, $\beta = 0.0125$.

MI Objectives. In the main text, for crVAE-GAN, we introduce a regularization term based on maximization of mutual information between the generation $\tilde{x}$ and its sampled latent variable z. We now mathematically justify and illustrate the formulation of the auxiliary objective:

$$\begin{aligned}
I(z, \tilde{x}) &= H(z) - H(z|\tilde{x}) \\
&= \mathbb{E}_{\tilde{x}\sim p_\theta(x|z)}\left[\mathbb{E}_{\tilde{z}\sim p(z|\tilde{x})}\left[\log p(\tilde{z}|\tilde{x})\right]\right] + H(z) \\
&\geq \mathbb{E}_{\tilde{x}\sim p_\theta(x|z)}\left[\mathbb{E}_{\tilde{z}\sim p(z|\tilde{x})}\left[\log q(\tilde{z}|\tilde{x})\right]\right] + H(z) \\
&= \mathbb{E}_{z\sim p(z),\tilde{x}\sim p_\theta(x|z)}\left[\mathbb{E}_{\tilde{z}\sim p(z|\tilde{x})}\left[\log q_\psi(\tilde{z}|\tilde{x})\right]\right] + H(z) \\
&= \mathbb{E}_{z\sim p(z),\tilde{x}\sim p_\theta(x|z)}\left[\log q_\psi(z|\tilde{x})\right] + H(z),
\end{aligned}$$

where $q_\psi(\tilde{z}|\tilde{x})$ is a variational approximation to $p(\tilde{z}|\tilde{x})$, parametrized by a neural network which shares the same encoding path with the discriminator. The last equality transformation comes from Lemma 5.1 in [2]. Similarly as done in [2], we bypass the optimization of $H(z)$ by treating it as a constant and leverage the reparametrization trick to sample from z. As in the case for the adversarial loss, when given the ground truth x, we sample z from its approximated posterior $q_\phi(z|x)$, otherwise sample from prior $p(z)$. Therefore, the overall objective for the auxiliary task becomes:

$$\mathcal{L}_{\text{MI}} = \mathbb{E}_{z\sim p(z),\tilde{x}\sim p_\theta(x|z)}\left[\log q_\psi(z|\tilde{x})\right] + \mathbb{E}_{x\sim X, z\sim q_\phi(z|x),\tilde{x}\sim p_\theta(x|z)}\left[\log q_\psi(z|\tilde{x})\right]$$

Recall that in the case of RGB images, the formulation $p_\theta(x|z)$ estimates the mean of the Gaussian distribution while fixing variance. Therefore, in practice, we do not perform additional sample from $p_\theta(x|z)$ but directly take the mean. Also, we assume $q_\psi(z|\tilde{x})$ to be a diagonal multivariate Gaussian with fixed σ. In this way, the auxiliary objective boils down to L^2 reconstruction of z from $\tilde{x}$. Finally, as explained in the main text, we found empirically reconstructing z does not work as well as reconstructing the transformed latent variable $u = \text{LSTM}(z)$, thus we set the objective to be the latter. Now the overall objective for crVAE-GAN becomes

$$\mathcal{L}_{\text{VAE-GAN}} = \mathcal{L}_{\text{recon}} + \alpha_1\mathcal{L}^1_{\text{prior}} + \alpha_2\mathcal{L}^2_{\text{prior}} + \beta\mathcal{L}_{\text{D}} + \kappa\mathcal{L}_{\text{MI}},$$

where for Birds, $\kappa = 0.02$, for CelebA, $\kappa = 0.01$ and for LSUN, $\kappa = 0.01$.

Stage2 Objective. The objective for the discriminator during Stage2 training is to maximize:

$$\mathcal{L}_D = \mathbb{E}_{s\sim S}\left[\log(D(s))\right] + \mathbb{E}_{z\sim p(z),\tilde{x}\sim p_\theta(x|z)}\left[\log(1 - D(G(\tilde{x}))\right] + \mathbb{E}_{z\sim q_\phi(z|x),\tilde{x}\sim p_\theta(x|z)}\left[\log(1 - D(G(\tilde{x}))\right].$$

The loss for the generator is to minimize

$$\mathcal{L}_G = \mathbb{E}_{z\sim p(z),\tilde{x}\sim p_\theta(x|z)}\left[\log(1 - D(G(\tilde{x}))\right] + \mathbb{E}_{z\sim q_\phi(z|x),\tilde{x}\sim p_\theta(x|z)}\left[\log(1 - D(G(\tilde{x}))\right] + \gamma\mathbb{E}_{z\sim q_\phi(z|x),\tilde{x}\sim p_\theta(x|z)}\left[\|f(S(x)) - f(G(\tilde{x}))\|_2^2\right],$$

where $S(x)$ is the ground truth upsampled x and f represent feature extraction via some pre-trained deep CNNs, namely VGG11 for Birds and CelebA and ResNet50 for LSUN. For the experiments without perceptual loss, $\gamma = 0$, otherwise $\gamma = 1$ for Birds and CelebA, and $\gamma = 10$ for bedroom.

Image Completion Hyperparameters. Recall the image completion objective:

$$\min_z \left[\|\hat{x} \odot m - x \odot m\|_2^2 + \gamma \log \mathcal{N}(z; 0, \mathbf{I}) + \tau \log(1 - D(\hat{x})\right].$$

We use $\gamma = 0.00001$ and $\tau = 0.003$.

E. More Generated Examples without Cherry-Picking.

(a) 64×64 VAE-GAN for Birds

(b) 128×128 VAE-GAN for Birds without Perceptual Loss

(c) 128×128 VAE-GAN for Birds with Perceptual Loss

Figure 1: Birds dataset with VAE-GAN.

(a) 64×64 crVAE-GAN for Birds

(b) 128×128 crVAE-GAN for Birds without Perceptual Loss

(c) 128×128 crVAE-GAN for Birds with Perceptual Loss

Figure 2: Birds dataset with crVAE-GAN.

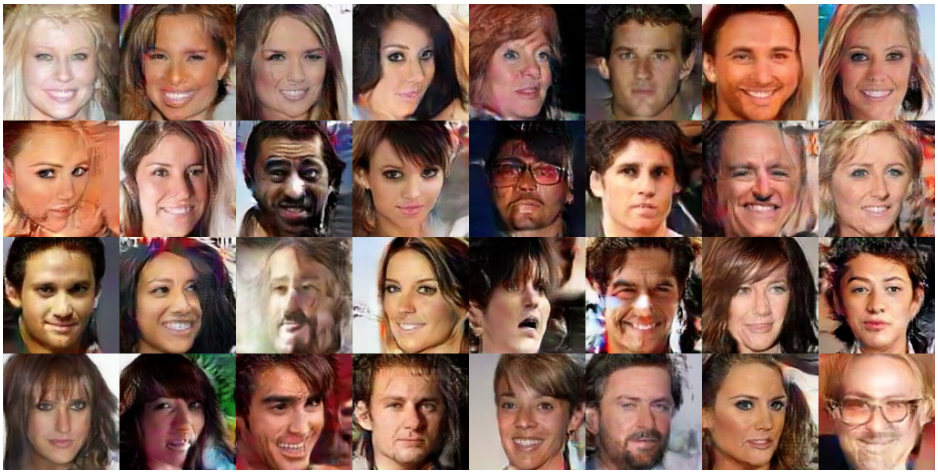
(a) 64×64 VAE-GAN for CelebA

(b) 128×128 VAE-GAN for CelebA without Perceptual Loss

(c) 128×128 VAE-GAN for CelebA with Perceptual Loss

Figure 3: CelebA dataset with VAE-GAN.

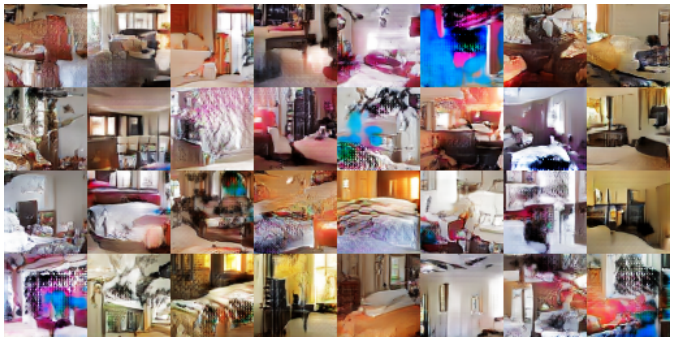
(a) 64×64 crVAE-GAN for CelebA



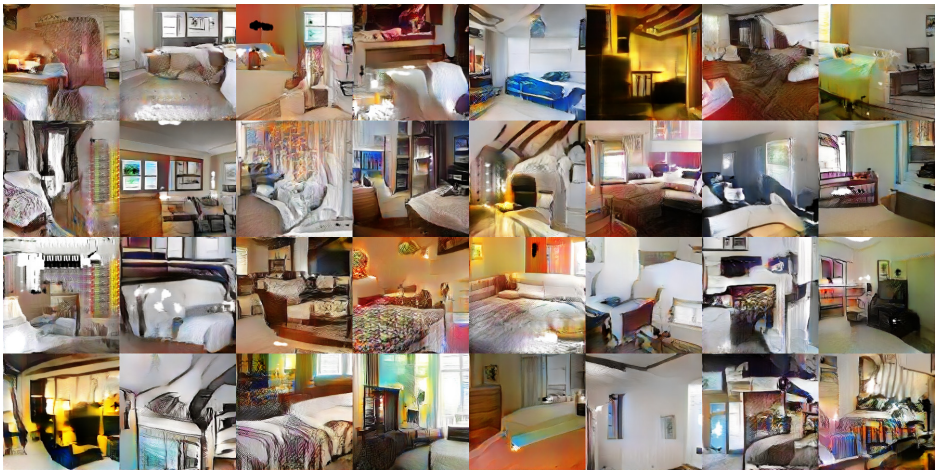
(b) 128×128 crVAE-GAN for CelebA without Perceptual Loss

(c) 128×128 crVAE-GAN for CelebA with Perceptual Loss

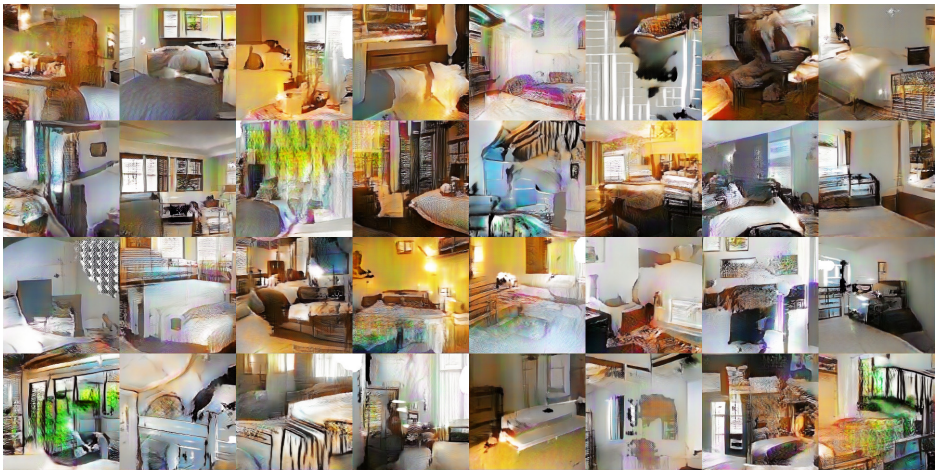
Figure 4: CelebA dataset with crVAE-GAN.



(a) 64×64 VAE-GAN for LSUN



(b) 224×224 VAE-GAN for LSUN without Perceptual Loss

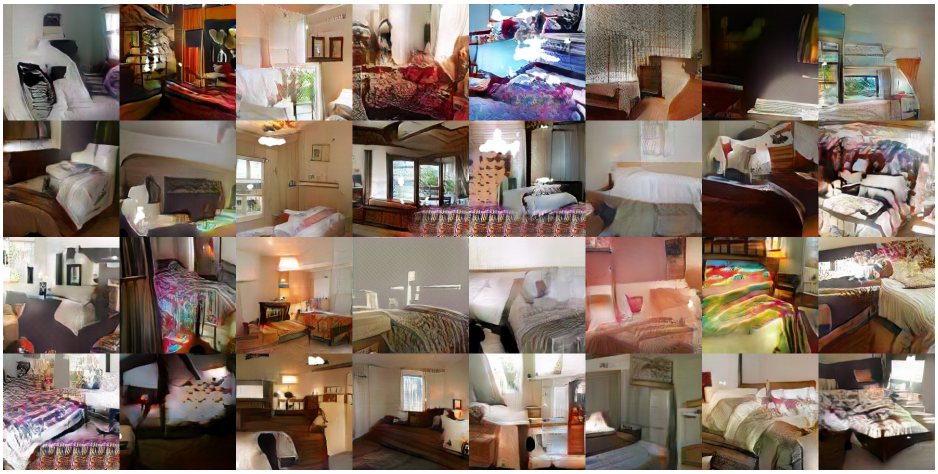


(c) 224×224 VAE-GAN for LSUN with Perceptual Loss

Figure 5: LSUN dataset with VAE-GAN. Due to the size limit of the Supplementary Materials, 224 images are compressed to jpg format.

(a) 64×64 crVAE-GAN for LSUN

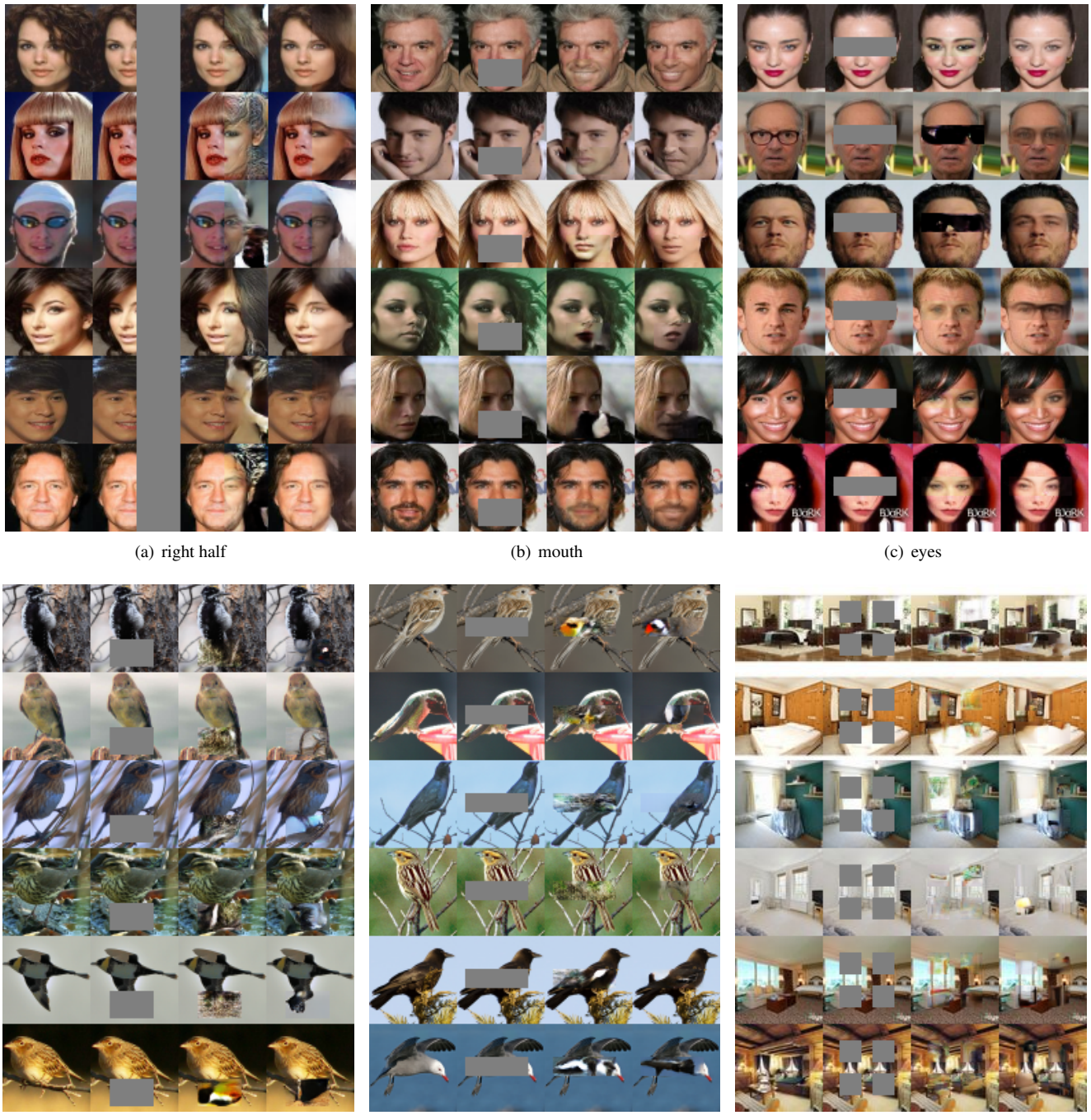
(b) 128×128 crVAE-GAN for LSUN without Perceptual Loss



(c) 128×128 crVAE-GAN for LSUN with Perceptual Loss

Figure 6: LSUN dataset with crVAE-GAN. Due to the size limit of the Supplementary Materials, 224 images are compressed to jpg format.

F. More Image Completion Examples.



(a) right half (b) mouth (c) eyes

(d) lower (e) upper (f) blocks

Figure 7: Each block from left to right: ground truth; occluded; VAE-GAN; crVAE-GAN.

References

[1] Y. Burda, R. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. *ICLR*, 2015. 1

[2] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, 2016. 3

[3] J. Duchi. Derivations for linear algebra and optimization. *Berkeley, California*, 2007. 1

[4] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. In *ICML*, 2015.

[5] K. Gregor, I. Danihelka, A. Mnih, C. Blundell, and D. Wierstra. Deep autoregressive networks. In *ICML*, 2014.

[6] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. 2

[7] D. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improving variational inference with inverse autoregressive flow. In *NIPS*, 2016.

[8] D. Kingma and M. Welling. Auto-encoding variational bayes. In *ICLR*, 2013. 1

[9] H. Larochelle. Binarized mnist dataset. http://www.cs.toronto.edu/~larocheh/public/datasets/binarized_mnist, 2015. 1

[10] A. B. L. Larsen and S. K. Sønderby. Generating faces with torch. http://torch.ch/blog/2015/11/13/gan.html, 2016. 2

[11] A. B. L. Larsen, S. K. Sonderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. In *ICML*, 2016. 2

[12] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *ICML*, 2015.

[13] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014. 1

[14] T. Salimans, D. Kingma, and M. Welling. Markov chain monte carlo and variational inference: Bridging the gap. In *ICML*, 2015.

[15] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv*, 2016. 2