Guided Image Generation with Conditional Invertible Neural Networks

Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, Ullrich Köthe
Visual Learning Lab Heidelberg

Abstract

In this work, we address the task of natural image generation guided by a conditioning input. We introduce a new architecture called conditional invertible neural network (cINN). The cINN combines the purely generative INN model with an unconstrained feed-forward network, which efficiently preprocesses the conditioning input into useful features. All parameters of the cINN are jointly optimized with a stable, maximum likelihood-based training procedure. By construction, the cINN does not experience mode collapse and generates diverse samples, in contrast to e.g. cGANs. At the same time our model produces sharp images since no reconstruction loss is required, in contrast to e.g. VAEs. We demonstrate these properties for the tasks of MNIST digit generation and image colorization. Furthermore, we take advantage of our bi-directional cINN architecture to explore and manipulate emergent properties of the latent space, such as changing the image style in an intuitive way.

Code and appendix available at
`github.com/VLL-HD/FrEIA`
Correspondence to
`lynton.ardizzone@iwr.uni-heidelberg.de`

Figure 1: Diverse colorizations, which our network created for the same grayscale image. One of them shows ground truth colors, but which? Solution at the bottom of the page.

1 Introduction

Generative adversarial networks (GANs) produce ever larger and more realistic samples [21, 3]. Hence they have become the primary choice for a majority of image generation tasks. As such, their conditional variants (cGANs) would appear to be the natural tool for conditional image generation as well, and they have successfully been applied in many scenarios [29, 32]. Despite numerous improvements, significant expertise and computational resources are required to find a training configuration for large GANs that is stable, and produces diverse images. A lack in diversity is especially common when the condition itself is an image, and special precautions have to be taken to avoid mode collapse.

Conditional variational autoencoders (cVAEs) do not suffer from the same problems. Training is generally stable, and since every data point is assigned a region in latent space, sampling yields the full variety of data seen during training. However cVAEs come with drawbacks of their own: The assumption of a Gaussian posterior on the decoder side implies an L2 reconstruction loss, which is known to cause blurriness. In addition, the partition of the latent space into diagonal Gaussians leads to either mode-mixing issues or regions of poor sample quality [23]. There has also been some success in combining aspects of both approaches for certain tasks, such as [17, 44, 33].

We propose a third approach, by extending Invertible Neural Networks (INNs, [8, 22, 1]) for the task of *conditional* image generation, by adding conditioning inputs to their core building blocks. INNs are neural networks which are by construction bijective, efficiently invertible, and have a tractable Jacobian determinant. They represent transport maps between the input distribution $p(\mathbf{x})$ and a prescribed, easy-to-sample-from latent distribution $p(\mathbf{z})$. During training, the likelihood of training samples from $p(\mathbf{x})$ is maximized in latent space, while at inference time, $\mathbf{z}$-samples can trivially be transformed back to the data domain. Previously, INNs have been used successfully for unconditional image generation, e.g. by [8] and [22].

Unconditional INN training is related to that of VAEs, but it compensates for some key disadvantages: Firstly,

since reconstructions are perfect by design, no reconstruction loss is needed, and generated images do not become blurry. Secondly, each $\mathbf{x}$ maps to exactly one $\mathbf{z}$ in latent space, and there is no need for posteriors $p(\mathbf{z} \,|\, \mathbf{x})$. This avoids the VAE problem of disjoint or overlapping regions in latent space. In terms of training stability and sample diversity, INNs show the same strengths as autoencoder architectures, but with superior image quality. We find that these positive aspects apply to conditional INNs (cINNs) as well.

One limitation of INNs is that their design restricts the use of some standard components of neural networks, such as pooling and batch normalization layers. Our conditional architecture alleviates this problem, as the conditional inputs can be preprocessed by a *conditioning network* with a standard feed-forward architecture, which can be learned jointly with the cINN to greatly improve its generative capabilities. We demonstrate the qualities of cINNs for conditional image generation, and uncover emergent properties of the latent space, for the tasks of conditional MNIST generation and diverse colorization of ImageNet.

Our work makes the following contributions:

- We propose a new architecture called conditional invertible neural network (cINN), which combines an INN with an unconstrained feed-forward network for conditioning. It generates diverse images with high realism and thus overcomes limitations of existing approaches.

- We demonstrate a stable, maximum likelihood-based training procedure for jointly optimizing the parameters of the INN and the conditioning network.

- We take advantage of our bidirectional cINN architecture to explore and manipulate emergent properties of the latent space. We illustrate this for MNIST digit generation and image colorization.

2   RELATED WORK

Conditional Generative Modeling. Modern generative models learn to transform noise (usually sampled from multivariate Gaussians) into desired target distributions. Methods differ by the model-family these transformations are picked from and by the losses determining optimal solutions.

Conditional generative adversarial networks (cGANs) [31] train a pair of neural networks: a *generator* transforms a pair of conditioning and noise vectors to images, and a *discriminator* penalizes unrealistic looking images. The conditioning information is either concatenated to the noise [31], or fed into the network via conditional batch-norm layers [9, 15, 33]. Ensuring diversity of the generated images (for fixed conditioning) appears to be challenging in this approach. Recent BigGANs [3] successfully address this problem by using very large networks and batch sizes, but require parallel training on

up to 512 TPUs. PacGANs [30] employ augmented discriminators, which evaluate entire batches of real or generated images together rather than one image at a time. CausalGANs [25] train two additional discriminator networks, called "labeler" and "anti-labeler", with the latter explicitly penalizing the lack of diversity. Pix2pix [17] addresses the important special case when the target is conditioned on an image in a different modality, e.g. to generate satellite images from maps. In addition to the discriminator loss, it minimizes the L1 distance between generated and ground-truth targets using a paired training set, which contains corresponding images from both modalities. This leads to impressive image quality, but lack of diversity seems to be an especially hard problem in this case. In contrast, our method does not need explicit precautions to promote diversity.

Bidirectional architectures augment generator networks with complementary encoder networks that learn the generator's inverse and enable reconstruction losses, which exploit cycle consistency requirements. Conditional variational autoencoders (cVAEs) [38] replace all distributions in a standard VAE [24] by the appropriate conditional distributions, and are trained to minimize the evidence lower bound (ELBO loss). Since variational distributions are typically Gaussian, the reconstruction penalty is equivalent to squared loss, resulting in rather blurry generated images. This is avoided by AGE networks [39] and CycleGANs [43], which combine standard cGAN discriminators with L1 reconstruction loss in the data domain, and bidirectional conditional GANs [20], which extend the GAN discriminator to act on the distributions in data and latent space jointly. SPADE [33], building upon pix2pix and pix2pixHD [40], augments cGANs with additional VAE encoders to shape the latent space such that diversity is ensured.

Instead of enforcing bijectivity through cycle losses, invertible neural networks are bidirectional by design, since encoder and generator are realized by forward and backward processing within a single bijective model. We focus on architectures whose forward and backward pass require the same computational effort. The coupling layer designs pioneered by NICE [7] and RealNVP [8] emerged as very powerful and flexible model families under this restriction. Using additive coupling layers, i-RevNets [19] demonstrated that the lack of information reduction from data space to latent space does not cause overfitting. The Glow architecture [22] combines affine coupling layers with invertible 1x1 convolutions and achieves impressive attribute manipulations (e.g. age, hair color) in generated faces images. This approach was recently generalized to video [27].

Thanks to tractable Jacobian determinants, the coupling layer architecture enables maximum likelihood training [7, 8], but experimental comparisons with other training methods are inconclusive so far. For instance, [5] found minimization of an adversarial loss to be superior to maximum likelihood training in RealNVPs, [36] trained i-

RevNets in the same manner as adversarial auto-encoders, i.e. with a discriminator acting in latent rather than data space, and Flow-GANs [11] performed best using bidirectional training, a combination of maximum likelihood and adversarial loss. On the other hand, maximum likelihood training worked well within Glow [22], and i-ResNets [2] could even be trained with approximated Jacobian determinants. In this work we reinforce the view that high-quality generative models can be trained by maximum likelihood loss alone. To the best of our knowledge, we are the first to apply the coupling layer design for *conditional* generative models, with the exception of [1], who use it to compute posteriors for (relatively small) inverse problems, but do not consider image generation.

Colorization. State-of-the-art regression models for colorization produce visually near-perfect images [16], but do not account for the ambiguity inherent in this inverse problem. To address this, models would ideally define a conditional distribution of plausible color images for a given grayscale input, instead of just returning a single "best" solution.

Popular existing approaches for diverse colorization predict per-pixel color histograms from a U-Net [42] or from hypercolumns of an adapted VGG network [28]. However, sampling from these local histograms independently can not lead to a spatially consistent colorization, requiring additional heuristic post-processing steps to avoid artefacts.

In terms of generative models, both VAEs [6] and cGANs [17, 4] have been proposed for the task. However, their solutions do not reach the quality of the regression-based models, and cGANs in particular often lack diversity. To compensate, modifications and extensions to generative approaches have been developed, such as auto-regressive models [12] and CRFs [34]. However, these methods are computationally very expensive and often unable to scale to realistic image sizes.

Conceptually closest to our proposed method is the work of [39], where an encoder network maps color information to a latent space and a generator network learns the inverse transform, both conditioned on the grayscale image. Their experiments however are limited to a data set with only cars, and just three latent dimensions, leading to global, but no local diversity.

In contrast to the above, our flow-based cINN generates diverse colorizations in one standard feed-forward pass. It models the distribution of all pixels jointly, and allows for meaningful latent space manipulations.

3 METHOD

Our method is an extension of the affine coupling block architecture established in [8]. There, each network block splits its input $\mathbf{u}$ into two parts $[\mathbf{u}_1, \mathbf{u}_2]$ and applies affine transformations between them that have strictly upper or lower triangular Jacobians:
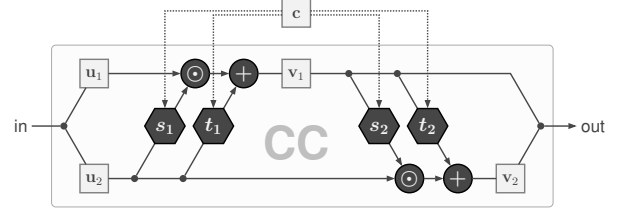

Figure 2: One conditional affine coupling block (CC).

$$
\begin{aligned}
\mathbf{v}_1 &= \mathbf{u}_1 \odot \exp\big(s_1(\mathbf{u}_2)\big) + t_1(\mathbf{u}_2) \\
\mathbf{v}_2 &= \mathbf{u}_2 \odot \exp\big(s_2(\mathbf{v}_1)\big) + t_2(\mathbf{v}_1) \, .
\end{aligned} \tag{1}
$$

The outputs $[\mathbf{v}_1, \mathbf{v}_2]$ are concatenated again and passed to the next coupling block. The internal functions s_j and t_j can be represented by arbitrary neural networks, and are only ever evaluated in the forward direction, even when the coupling block is inverted:

$$
\begin{aligned}
\mathbf{u}_2 &= \big(\mathbf{v}_2 - t_2(\mathbf{v}_1)\big) \oslash \exp\big(s_2(\mathbf{v}_1)\big) \\
\mathbf{u}_1 &= \big(\mathbf{v}_1 - t_1(\mathbf{u}_2)\big) \oslash \exp\big(s_1(\mathbf{u}_2)\big) \, .
\end{aligned} \tag{2}
$$

As shown in [8], the logarithm of the Jacobian determinant for such a coupling block is simply the sum of s_1 and s_2 over image dimensions.

3.1 CONDITIONAL INVERTIBLE TRANSFORMATIONS

We adapt the design of Eqs. (1) and (2) to produce a conditional version of the coupling block. Because the subnetworks s_j and t_j are never inverted, we can concatenate conditioning data $\mathbf{c}$ to their inputs without losing the invertibility, replacing $s_1(\mathbf{u}_2)$ with $s_1(\mathbf{u}_2, \mathbf{c})$ etc. Our conditional coupling block design is illustrated in Fig. 2.

In general, we will refer to a cINN with network parameters θ as $f(\mathbf{x}; \mathbf{c}, \theta)$, and the inverse as $g(\mathbf{z}; \mathbf{c}, \theta)$. For any fixed condition $\mathbf{c}$, the invertibility is given as

$$
f^{-1}(\,\cdot\,; \mathbf{c}, \theta) = g(\,\cdot\,; \mathbf{c}, \theta). \tag{3}
$$

3.2 MAXIMUM LIKELIHOOD TRAINING OF CINNs

By prescribing a probability distribution $p_Z(\mathbf{z})$ on latent space Z, the model f assigns any input $\mathbf{x}$ a probability, dependent on both the network parameters θ and the conditioning $\mathbf{c}$, through the change-of-variables formula:

$$
p_X(\mathbf{x}; \mathbf{c}, \theta) = p_Z\big(f(\mathbf{x}; \mathbf{c}, \theta)\big) \left| \det\left(\frac{\partial f}{\partial \mathbf{x}}\right) \right| . \tag{4}
$$

Here, we use the Jacobian matrix $\partial f / \partial \mathbf{x}$. We will denote the determinant of the Jacobian, evaluated at some training sample $\mathbf{x}_i$, as $J_i \equiv \det\big(\partial f / \partial \mathbf{x}|_{\mathbf{x}_i}\big)$. Bayes' theorem gives us the posterior over model parameters as $p(\theta; \mathbf{x}, \mathbf{c}) \propto p_X(\mathbf{x}; \mathbf{c}, \theta) \cdot p_\theta(\theta)$. Our goal is to find network parameters that maximize its logarithm, i.e. we minimize the loss

$$
\mathcal{L} = \mathbb{E}_i \big[-\log\big(p_X(\mathbf{x}_i; \mathbf{c}_i, \theta)\big)\big] - \log\big(p_\theta(\theta)\big), \tag{5}
$$

which is the same as in classical Bayesian model fitting.

Inserting Eq. (4) with a standard normal distribution for $p_Z(\mathbf{z})$, as well as a Gaussian prior on the weights θ with $1/2\sigma_\theta^2 \equiv \tau$, we obtain

$$\mathcal{L} = \mathbb{E}_i\left[\frac{\|f(\mathbf{x}_i; \mathbf{c}_i, \theta)\|_2^2}{2} - \log|J_i|\right] + \tau\|\theta\|_2^2 . \quad (6)$$

The latter term represents L2 weight regularization, while the former is the *maximum likelihood loss*.

Training a network with this loss yields an estimate of the maximum likelihood network parameters $\hat{\theta}_{\mathrm{ML}}$. From there, we can perform conditional generation for a fixed $\mathbf{c}$ by sampling $\mathbf{z}$ and using the inverted network g: $\mathbf{x}_{\mathrm{gen}} = g(\mathbf{z}; \mathbf{c}, \hat{\theta}_{\mathrm{ML}})$, with $\mathbf{z} \sim p_Z(\mathbf{z})$.

Training with the maximum likelihood method makes it virtually impossible for mode collapse to occur: If any mode in the training set has low probability under the current guess $p_X(\mathbf{x}; \mathbf{c}, \theta)$, the corresponding latent vectors will lie far outside the normal distribution p_Z and receive big loss from the first L2-term in Eq. (6). In contrast, the discriminator of a GAN only supplies a weak signal, proportional to the mode's relative frequency in the training data, so that the generator is not penalized much for ignoring a mode completely.

3.3 CONDITIONING NETWORK

In complex settings, we expect that higher-level features of $\mathbf{c}$ need to be extracted for the conditioning to be effective, e.g. global semantic information from an image as in Section 4.2. In such cases, feeding the condition $\mathbf{c}$ directly into the cINN would place an unreasonable burden on the s and t networks, as higher-level features would have to be re-learned in each coupling block.

To address this issue, we introduce an additional feed-forward *conditioning network* h, which transforms the condition $\mathbf{c}$ to some intermediate representation $\tilde{\mathbf{c}} = h(\mathbf{c})$, and replace $\mathbf{c}_i$ in Eq. (6) with $\tilde{\mathbf{c}}_i = h(\mathbf{c}_i)$. The network h can be pretrained, e.g. by using features from a VGG architecture trained for image classification. Alternatively or additionally, h can be trained jointly with the cINN by propagating gradients from the maximum likelihood loss through the conditioning $\tilde{\mathbf{c}}$. In this case, the conditioning network will learn to extract features which are particularly useful for embedding the cINN inputs $\mathbf{x}$ into latent variables $\mathbf{z}$.

3.4 IMPORTANT DETAILS

For cINNs to match the performance of well-established architectures for conditional generation, we introduce a number of minor modifications and adjustments to the architecture and training procedure. With these adaptions, our training setup is very stable and converges every time. Ablation results are presented in Sec. 4.4.

Noise as data augmentation. We add a small amount of noise to the inputs $\mathbf{x}$ as part of the standard data augmentation. This helps to smooth out quantization artifacts in
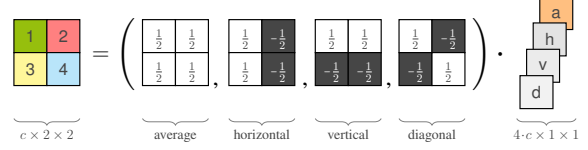

Figure 3: Haar wavelet downsampling reduces spatial dimensions & separates lower frequencies (a) from high (h,v,d).

the input, and prevents sparse gradients when large parts of the image are completely flat (as e.g. in MNIST).

Soft clamping of scale coefficients. We apply an additional nonlinear function to the scale coefficients s, of the form

$$s_{\mathrm{clamp}} = \frac{2\alpha}{\pi}\arctan\left(\frac{s}{\alpha}\right), \quad (7)$$

which yields $s_{\mathrm{clamp}} \approx s$ for $|s| \ll \alpha$ and $s_{\mathrm{clamp}} \approx \pm\alpha$ for $|s| \gg \alpha$. This prevents any instabilities stemming from exploding magnitude of the exponential $\exp(s_{\mathrm{clamp}})$. We find $\alpha = 1.9$ to be a good value for most architectures.

Initialization. Heuristically, we find that Xavier initialization [10] leads to stable training from the start. We experienced training instability when initial parameter values were too high. Similar to [22], we also initialize the last convolution in all s and t subnetworks to zero, so training starts from an identity transform.

Soft channel permutations. We use random orthogonal matrices to mix the information between the channels. This allows for more interaction between the two information streams $\mathbf{u}_1$, $\mathbf{u}_2$ in the coupling blocks. A similar technique was used in [22], but our matrices stay fixed throughout training and are guaranteed to be cheaply invertible.

Haar wavelet downsampling. All prior INN architectures use checkerboard patterns for reshaping to lower spatial resolutions. We find it helpful to instead perform downsampling with Haar wavelets [13], which essentially decompose images into an average pooling channel as well as vertical, horizontal and diagonal derivatives, see Fig. 3. The three derivative channels contain high resolution information which we can split off early, transforming only the remaining information further in later stages of the cINN. This also contributes to mixing the variables between layers, complementing the soft permutations. Similarly, [18] uses a discrete cosine transform as a final transformation in their INN, to replace global average pooling.

4 EXPERIMENTS

We present results and explore the latent space of our models for two conditional image generation tasks: MNIST digit generation and image colorization.
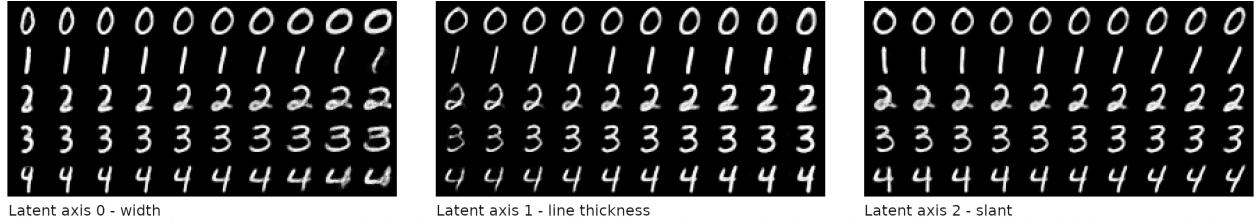
Latent axis 0 - width   Latent axis 1 - line thickness   Latent axis 2 - slant

Figure 4: Axes in our MNIST model's latent space, which linearly encode the style attributes width, thickness and slant.

4.1 Class-conditional generation for MNIST

As a first experiment, we perform simple class-conditional generation of MNIST digits. We construct a cINN of 24 coupling blocks using fully connected subnetworks s and t, which receive the conditioning directly as a one-hot vector (Fig. 5). No conditioning network h is used. For data augmentation we only add a small amount of noise to the images ($\sigma = 0.02$), as described in Section 3.4.

Samples generated by the model are shown in Fig. 6. We find that the cINN learns latent representations that are shared across conditions $\mathbf{c}$. Keeping the latent vector $\mathbf{z}$ fixed while varying $\mathbf{c}$ produces different digits in the same style. This property, in conjunction with our network's invertibility, can directly be used for style transfer, as demonstrated in Fig. 7. This outcome is not obvious – the trained cINN could also decompose into 10 essentially separate subnetworks, one for each condition. In this case, the latent space of each class would be structured differently, and inter-class transfer of latent vectors would be meaningless. The structure of the latent space is further illustrated in Fig. 4, where we identify three latent axes with interpretable meanings. Note that while the latent space is learned without supervision, we found the axes in a semi-automatic fashion: We perform PCA on the latent vectors of the test set, without the noise augmentation, and manually identify meaningful directions in the subspace of the first four principal components.

4.2 Diverse ImageNet colorization

For a more challenging task, we turn to colorization of natural images. The common approach for this task is to represent images in Lab color space and generate color channels $\mathbf{a}, \mathbf{b}$ by a model conditioned on the luminance channel $\mathbf{L}$.
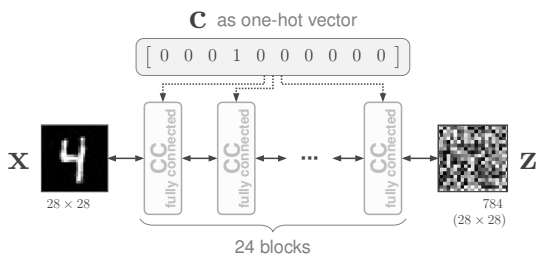

Figure 5: cINN model for conditional MNIST generation.

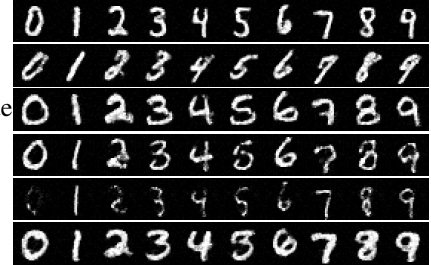Tidy
Slanted, narrow
Slanted left, wide
Messy
Faint
Bold

Figure 6: MNIST samples from our cINN conditioned on digit labels. All ten digits within one row $(0, \ldots, 9)$ were generated using the same latent code $\mathbf{z}$, but changing condition $\mathbf{c}$. We see that each $\mathbf{z}$ encodes a single style consistently across digits, while varying $\mathbf{z}$ between rows leads to strong differences in writing style.

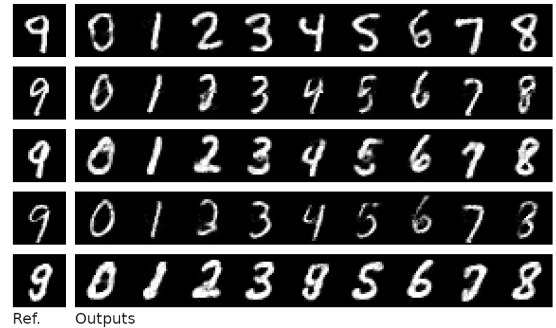Ref.   Outputs

Figure 7: To perform style transfer, we determine the latent code $\mathbf{z} = f(\mathbf{x}; \mathbf{c}, \theta)$ of a validation image *(left)*, then use the inverse network $g = f^{-1}$ with different conditions $\hat{\mathbf{c}}$ to generate the other digits in the same style, $\hat{\mathbf{x}} = g(\mathbf{z}; \hat{\mathbf{c}}, \theta)$.

We train on the ImageNet dataset [35], again adding low noise to the $\mathbf{a}, \mathbf{b}$ channels ($\sigma = 0.05$). As the color channels do not require as much resolution as the luminance channel, we condition on 256×256 pixel grayscale images, but generate 64×64 pixel color information. This is in accordance with the majority of existing colorization methods.

As with most generative INN architectures, we do not keep the resolution and channels fixed throughout the network, for the sake of computational cost. Instead, we use 4 resolution stages, as illustrated in Fig. 8. At each stage, the data is reshaped to a lower resolution and more channels, after which a fraction of the channels are split
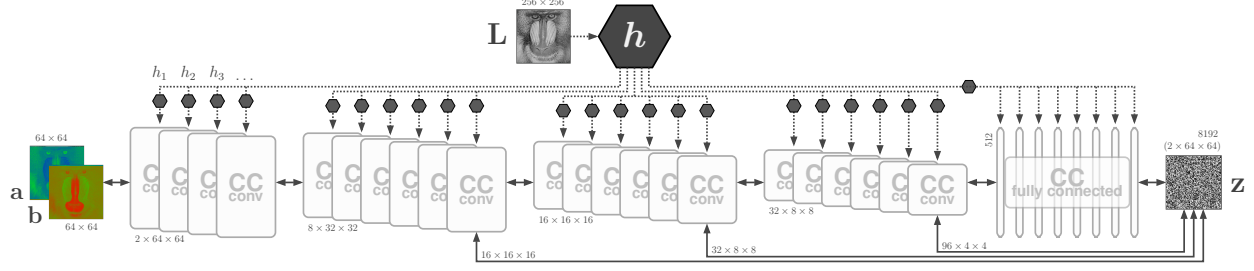
Figure 8: cINN model for diverse colorization. The conditioning network h consists of a truncated VGG [37] pretrained to predict colors on ImageNet, with separate convolutional heads $h_1, h_2, h_3, \ldots$ tailoring the extracted features to each individual conditional coupling block (CC). After each group of coupling blocks, we apply Haar wavelet downsampling (Fig. 3) to reduce the spatial dimensions and, where indicated by arrows, split off parts of the latent code $\mathbf{z}$ early.

off as one part of the latent code. As the high resolution stages have a smaller receptive field and less expressive power, the corresponding parts of the latent vector encode local structures and noise. More global information is passed on to the lower resolution sections of the cINN.

For the conditioning network h, we start with the same VGG-like architecture and pretraining as [42], i.e. we pre-train the network to classify each pixel of the gray image into color bins. By cutting off the network before the second-to-last convolution, we extract 256 feature maps of size 64×64 from the grayscale image $\mathbf{L}$. We then add independent heads on top of this for each conditional coupling block in the cINN, indicated by small hexagons in Fig. 8. Thus each coupling block k receives its own specialized conditioning $\tilde{\mathbf{c}}_i^{(k)} = h_k\big(h(\mathbf{c}_i)\big)$. Each head consists of up to five strided convolutions, depending on its required output resolution, and a batch normalization layer. The ablation study in Fig. 16 confirms that the conditioning network is necessary to capture semantic information.

We initially train the cINN and the h_k, keeping the parameters of the conditioning network h fixed, for $30\,000$ iterations. After this, we train both jointly until convergence, for 3 days on 3 Nvidia GTX1080 GPUs. The Adam optimizer is essential for fast convergence, and we lower the learning rate when the maximum likelihood loss levels off.

At inference time, we use joint bilateral upsampling [26] to match the resolution of the generated color channels $\hat{\mathbf{a}}, \hat{\mathbf{b}}$ to that of the luminance channel $\mathbf{L}$. This produces visually slightly more pleasing edges than bicubic upsampling, but has little to no impact on the results. It was not used in the quantitative results table, to ensure an unbiased comparison.

The cINN compares favourably to existing methods, as shown in Table 1, and has the best diversity and best-of-8 accuracy of the compared methods. The cGAN apparently ignores the latent code, and relies only on the condition. As a result, we do not measure any significant diversity, in line with results from [17].

In terms of FID score, the cGAN performs best, although its results do not appear more realistic to the human eye, cf. Fig. 13. This may be due to the fact that FID is sensitive to outliers, which are unavoidable for a truly diverse method (see Fig. 12), or because the discriminator loss implicitly optimizes for the similarity of deep CNN activations. The VGG classification accuracy of generative methods is decreased compared to CNN, because occasional outliers may lead to misclassification. Latent space interpolations and color transfer are shown in Figs. 14 and 15.

4.3 Diverse bedrooms colorization

To provide a simpler model for more in-depth experiments and ablations, we additionally train a cINN for colorization on the LSUN bedrooms dataset [41]. We use a smaller model than for ImageNet, and train the conditioning network jointly from scratch, without pretraining. Both the conditioning input, as well as the generated color channels have a resolution of 64×64 pixels. The entire model trains in under 4 hours on a single GTX 1080Ti GPU.

To our knowledge, the only diversity-enforcing cGAN architecture previously used for colorization is the colorGAN [4], which is also trained exclusively on the bedrooms dataset. Training the colorGAN for comparison, we find it requires over 24 hours to converge stably, after multiple restarts. The results are generally worse than those of the cINN, as shown in Fig. 9. While the resulting pixel-wise color variance is slightly higher for the colorGAN, it is not clear whether this captures the true variance, or whether it is due to unrealistically colorful outputs, such as in the second row in Fig. 9.

4.4 Ablation of training improvements

To demonstrate the improved stability and training speed through the improvements from Sec. 3.4, we perform ablations, see Fig. 10. The ablations for colorization were performed for the LSUN bedrooms task, due to training speed.

6

Metric	cINN	colorGAN
MSE best-of-8	**6.14**	6.43
Variance	33.69	**39.46**
FID	**26.48**	28.31

cINN   COLORGAN

Figure 9: Quantitative and qualitative comparison between smaller cINN and colorGAN on LSUN bedrooms. The metrics used are explained in Table 1.

We find that for stable training at Adam learning rates of 10^{-3}, the clamping and Haar wavelet downsampling are strictly necessary. Without these, the network has to be trained with much lower learning rates and more careful and specialized initialization, as used e.g. in [22]. Beyond this, the noise augmentation and permutations lead to the largest improvement in final result. The effect of the noise is more pronounced for MNIST, as large parts of the image are completely black otherwise. For natural images, dequantization of the data is likely to be the main advantage of the added noise. The initialization only improves the final result by a small margin, but also converges noticeably faster.

5 CONCLUSION AND OUTLOOK

We have proposed a conditional invertible neural network architecture which enables guided generation of diverse images with high realism. For image colorization, we believe that even better results can be achieved when employing latest tricks from large-scale GAN frameworks. Especially the non-invertible nature of the conditioning network make cINNs a suitable method for other computer vison tasks such as diverse semantic segmentation.

6 ACKNOWLEDGMENTS

Ablation	Final loss
Full model	-3.364
No noise aug.	-2.244
No Xavier init.	-3.341
No perm.	-3.198
No clamp	-0.808 (div.)

Ablation	Final loss
Full model	-2.732
No noise aug.	-2.701
No Xavier init.	-2.730
No perm.	-2.720
No clamp	0.107 (div.)
No Haar	-2.570 (div.)
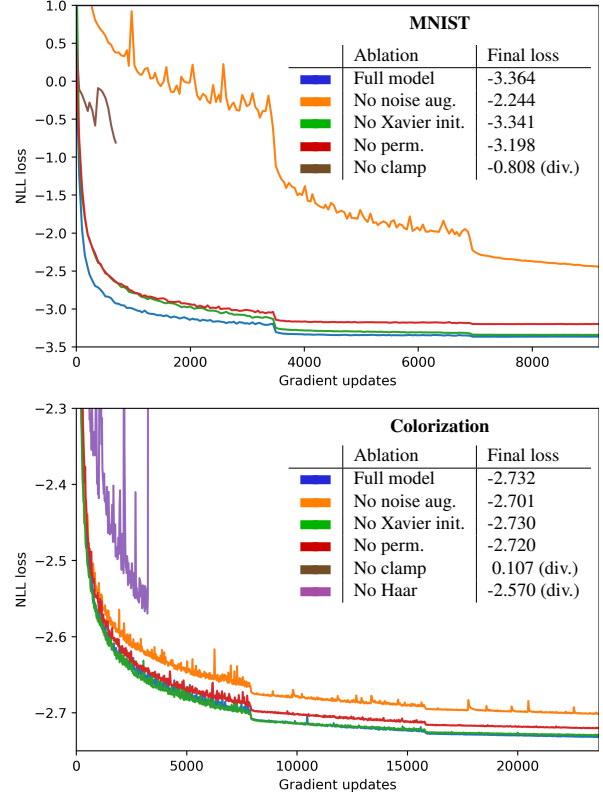
Figure 10: Training curves for each task, ablating the different improvements.

	cINN (ours)	VAE-MDN [6]	cGAN [17]	CNN [16]	BW	Ground truth
MSE best of 8	**3.53±0.04**	4.06±0.04	9.75±0.06	6.77 ±0.05	–	–
Variance	**35.2±0.3**	21.1±0.2	0.0±0.0	–	–	–
FID [14]	25.13±0.30	25.98±0.28	**24.41±0.27**	24.95±0.27	30.91 ± 0.27	14.69 ± 0.18
VGG top 5 acc.	85.00±0.48	85.00±0.48	84.62±0.53	**86.86±0.41**	86.02±0.43	91.66 ± 0.43

Table 1: Comparison of conditional generative models for diverse colorization. We additionally compare to a state-of-the-art regression model ('CNN', no diversity), and the grayscale images alone ('BW'). For each of 5k ImageNet validation images, we compare the best pixel-wise MSE of 8 generated colorization samples, the pixel-wise variance between the 8 samples as an approximation of the diversity, the Fréchet Inception Distance [14] as a measure of realism, and the top 5 accuracy of ImageNet classification performed on the colorized images, to check if semantic content is preserved by the colorization.
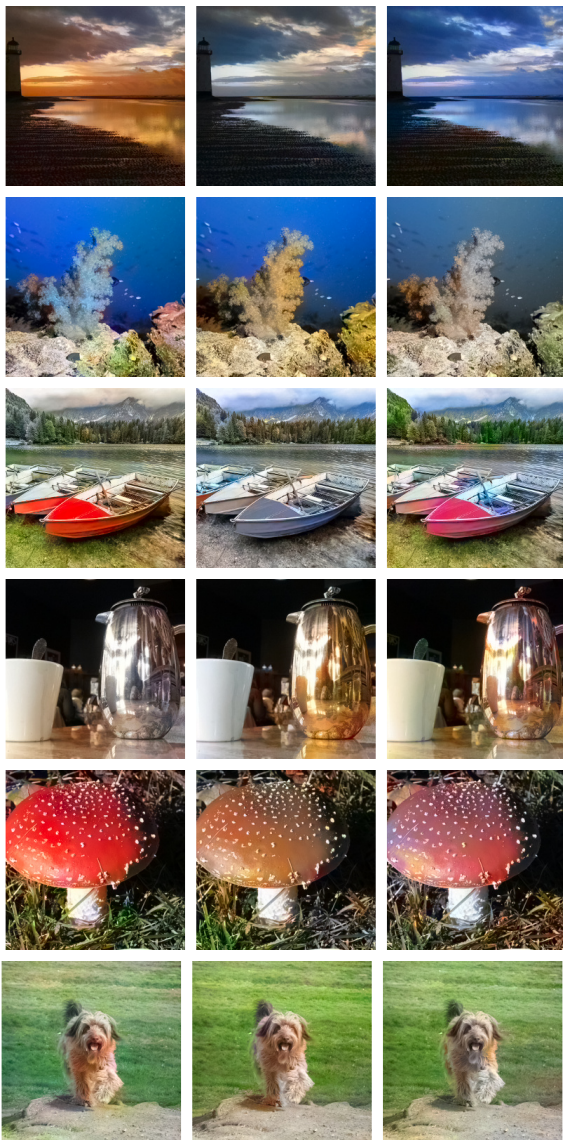

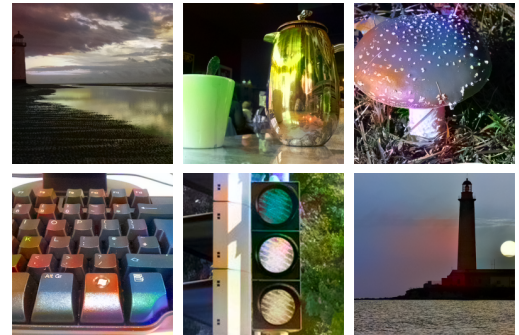Figure 11: Diverse colorizations produced by our cINN.

Figure 12: Failure cases of our method. *Top:* Sampling outliers. *Bottom:* cINN did not recognize an object's semantic class or the connectivity of occluded regions.
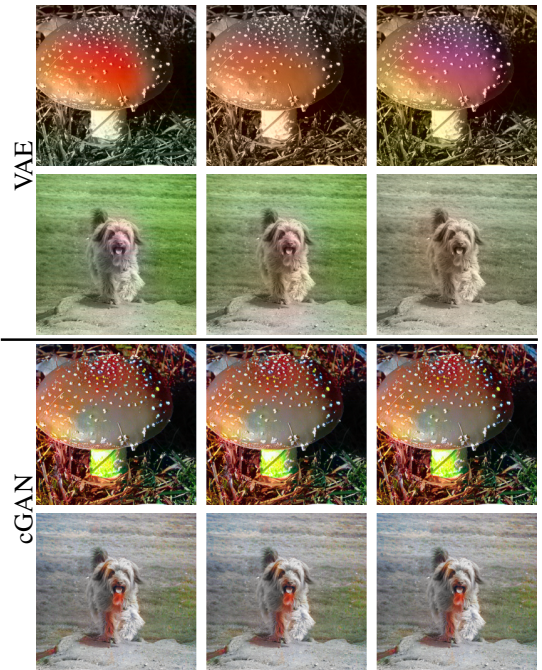

Figure 13: Alternative methods have lower diversity and lower quality, suffering from inconsistencies within objects, or color blurriness and bleeding (compare Fig. 11, bottom).
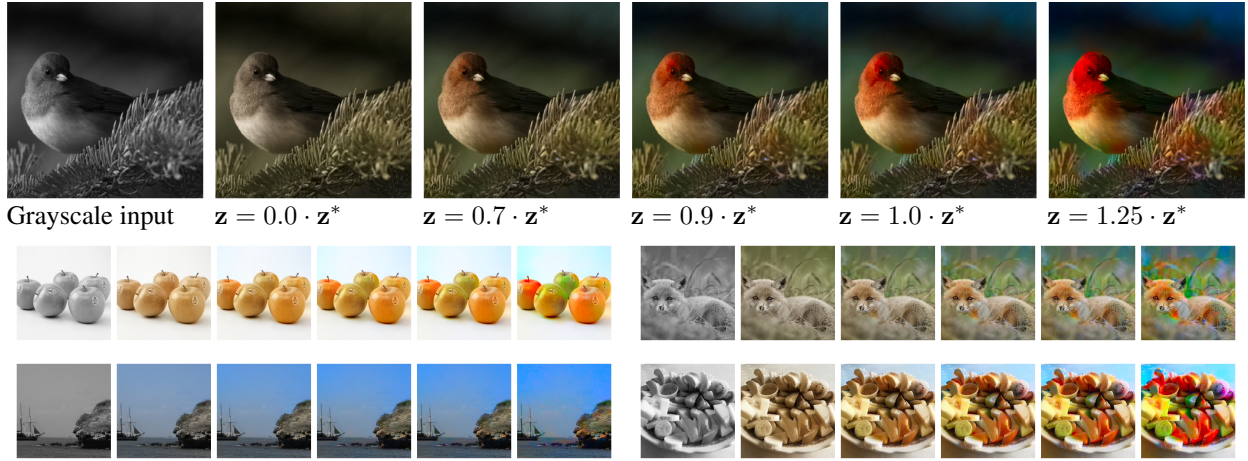
Figure 14: Effects of linearly scaling the latent code $\mathbf{z}$ while keeping the condition fixed. Vector $\mathbf{z}^*$ is "typical" in the sense that $\|\mathbf{z}^*\|^2 = \mathbb{E}[\|\mathbf{z}\|^2]$, and results in natural colors. As we move closer to the center of the latent space ($\|\mathbf{z}\| < \|\mathbf{z}^*\|$), regions with ambiguous colors become desaturated, while less ambiguous regions (e.g. sky, vegetation) revert to their prototypical colors. In the opposite direction ($\|\mathbf{z}\| > \|\mathbf{z}^*\|$), colors are enhanced to the point of oversaturation.

Figure 15: For color transfer, we first compute the latent vectors $\mathbf{z}$ for different color images $(\mathbf{L}, \mathbf{a}, \mathbf{b})$ *(top row)*. We then send the same $\mathbf{z}$ vectors through the inverse network with a new grayscale condition $\mathbf{L}^*$ *(far left)* to produce transferred colorizations $\mathbf{a}^*, \mathbf{b}^*$ *(bottom row)*. Differences between reference and output color (e.g. pink rose) can arise from mismatches between the reference colors $\mathbf{a}, \mathbf{b}$ and the intensity prescribed by the new condition $\mathbf{L}^*$.
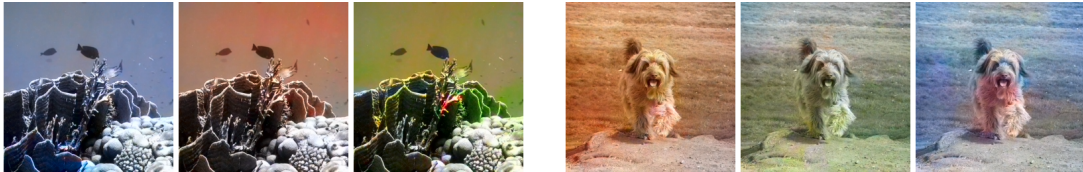

Figure 16: In an ablation study, we train a cINN using the grayscale image directly as conditional input, without a conditioning network h. The resulting colorizations largely ignore semantic content which leads to exaggerated diversity. More ablations are found in the appendix.

REFERENCES

[1] L. Ardizzone, J. Kruse, C. Rother, and U. Köthe. Analyzing inverse problems with invertible neural networks. In *Intl. Conf. on Learning Representations*, 2019. 1, 3

[2] J. Behrmann, D. Duvenaud, and J.-H. Jacobsen. Invertible residual networks. *arXiv:1811.00995*, 2018. 3

[3] A. Brock, J. Donahue, and K. Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *Intl. Conf. on Learning Representations*, 2019. 1, 2

[4] Y. Cao, Z. Zhou, W. Zhang, and Y. Yu. Unsupervised diverse colorization via generative adversarial networks. In *Joint Europ. Conf. on Machine Learning and Knowledge Discovery in Databases*, pages 151–166. Springer, 2017. 3, 6

[5] I. Danihelka, B. Lakshminarayanan, B. Uria, D. Wierstra, and P. Dayan. Comparison of maximum likelihood and GAN-based training of RealNVPs. *arXiv:1705.05263*, 2017. 2

[6] A. Deshpande, J. Lu, M.-C. Yeh, M. Jin Chong, and D. Forsyth. Learning diverse image colorization. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 6837–6845, 2017. 3, 8

[7] L. Dinh, D. Krueger, and Y. Bengio. NICE: Non-linear independent components estimation. *arXiv:1410.8516*, 2014. 2

[8] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using Real NVP. *arXiv:1605.08803*, 2016. 1, 2, 3

[9] V. Dumoulin, J. Shlens, and M. Kudlur. A learned representation for artistic style. In *Intl. Conf. on Learning Representations*, 2017. 2

[10] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proc. 13. Intl. Conf. Artificial Intelligence and Statistics*, pages 249–256, 2010. 4

[11] A. Grover, M. Dhar, and S. Ermon. Flow-GAN: combining maximum likelihood and adversarial learning in generative models. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 3

[12] S. Guadarrama, R. Dahl, D. Bieber, M. Norouzi, J. Shlens, and K. Murphy. Pixcolor: Pixel recursive colorization. *arXiv:1705.07208*, 2017. 3

[13] A. Haar. Zur Theorie der orthogonalen Funktionensysteme. *Mathematische Annalen*, 69(3):331–371, 1910. 4

[14] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017. 8

[15] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV'17*, pages 1501–1510, 2017. 2

[16] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics (TOG)*, 35(4):110, 2016. 3, 8

[17] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR'17*, pages 1125–1134, 2017. 1, 2, 3, 6, 8

[18] J.-H. Jacobsen, J. Behrmann, R. Zemel, and M. Bethge. Excessive invariance causes adversarial vulnerability. *arXiv preprint arXiv:1811.00401*, 2018. 4

[19] J.-H. Jacobsen, A. W. Smeulders, and E. Oyallon. i-RevNet: deep invertible networks. In *International Conference on Learning Representations*, 2018. 2

[20] A. Jaiswal, W. AbdAlmageed, Y. Wu, and P. Natarajan. Bidirectional conditional generative adversarial networks. *arXiv:1711.07461*, 2017. 2

[21] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. *arXiv:1710.10196*, 2017. 1

[22] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv:1807.03039*, 2018. 1, 2, 3, 4, 7

[23] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pages 4743–4751, 2016. 1

[24] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. *arXiv:1312.6114*, 2013. 2

[25] M. Kocaoglu, C. Snyder, A. G. Dimakis, and S. Vishwanath. CausalGAN: Learning causal implicit generative models with adversarial training. *arXiv:1709.02023*, 2017. 2

[26] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele. Joint bilateral upsampling. In *ACM Transactions on Graphics (ToG)*, volume 26, page 96. ACM, 2007. 6

[27] M. Kumar, M. Babaeizadeh, D. Erhan, C. Finn, S. Levine, L. Dinh, and D. Kingma. Videoflow: A flow-based generative model for video. *arXiv:1903.01434*, 2019. 2

[28] G. Larsson, M. Maire, and G. Shakhnarovich. Learning representations for automatic colorization. In *Europ. Conf. on Computer Vision*, pages 577–593. Springer, 2016. 3

[29] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Intl. Conf. on Computer Vision and Pattern Recognition*, pages 4681–4690, 2017. 1

[30] Z. Lin, A. Khetan, G. Fanti, and S. Oh. PacGAN: The power of two samples in generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 1498–1507, 2018. 2

[31] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv:1411.1784*, 2014. 2

[32] T. Miyato and M. Koyama. cGANs with projection discriminator. In *International Conference on Learning Representations*, 2018. 1

[33] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu. Semantic image synthesis with spatially-adaptive normalization. *arXiv:1903.07291*, 2019. 1, 2

[34] A. Royer, A. Kolesnikov, and C. H. Lampert. Probabilistic image colorization. In *British Machine Vision Conference (BMVC)*, 2017. 3

[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 5

[36] R. T. Schirrmeister, P. Chrabaszcz, F. Hutter, and T. Ball. Training generative reversible networks. *arXiv:1806.01610*, 2018. 2

[37] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 6

[38] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3483–3491. 2015. 2

[39] D. Ulyanov, A. Vedaldi, and V. Lempitsky. It takes (only) two: Adversarial generator-encoder networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2, 3

[40] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018. 2

[41] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 6

[42] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *Europ.Conf. on Computer Vision*, pages 649–666, 2016. 3, 6

[43] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV'17*, pages 2223–2232, 2017. 2

[44] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, pages 465–476, 2017. 1