MaCow: Masked Convolutional Generative Flow

Xuezhe Ma, Xiang Kong, Shanghang Zhang, Eduard Hovy
Carnegie Mellon University
Pittsburgh, PA, USA
xuezhem,xiangk@cs.cmu.edu, shanghaz@andrew.cmu.edu, hovy@cmu.edu

Abstract

Flow-based generative models, conceptually attractive due to tractability of the exact log-likelihood computation and latent-variable inference as well as efficiency in training and sampling, has led to a number of impressive empirical successes and spawned many advanced variants and theoretical investigations. Despite computational efficiency, the density estimation performance of flow-based generative models significantly falls behind those of state-of-the-art autoregressive models. In this work, we introduce masked convolutional generative flow (MaCow), a simple yet effective architecture for generative flow using masked convolution. By restricting the local connectivity to a small kernel, MaCow features fast and stable training along with efficient sampling while achieving significant improvements over Glow for density estimation on standard image benchmarks, considerably narrowing the gap with autoregressive models.

1 Introduction

Unsupervised learning of probabilistic models is a central yet challenging problem. Deep generative models have shown promising results in modeling complex distributions such as natural images (Radford et al., 2015), audio (Van Den Oord et al., 2016) and text (Bowman et al., 2015). Multiple approaches emerged in recent years, including Variational Autoencoders (VAEs) (Kingma and Welling, 2014), Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), autoregressive neural networks (Larochelle and Murray, 2011; Oord et al., 2016), and flow-based generative models (Dinh et al., 2014, 2016; Kingma and Dhariwal, 2018). Among these, flow-based generative models gained popularity for this capability of estimating densities of complex distributions, efficiently generating high-fidelity syntheses, and automatically learning useful latent spaces.

Flow-based generative models typically warp a simple distribution into a complex one by mapping points from the simple distribution to the complex data distribution through a chain of invertible transformations with Jacobian determinants that are efficient to compute. This design guarantees that the density of the transformed distribution can be analytically estimated, making maximum likelihood learning feasible. Flow-based generative models have spawned significant interests for improving and analyzing its algorithms both theoretically and practically, and applying them to a wide range of tasks and domains.

In their pioneering work, Dinh et al. (2014) first proposed Non-linear Independent Component Estimation (NICE) to apply flow-based models for modeling complex high-dimensional densities. RealNVP (Dinh et al., 2016) extended NICE with a more flexible invertible transformation to experiment with natural images. However, these flow-based generative models resulted in worse density estimation performance compared to state-of-the-art autoregressive models, and are incapable of realistic synthesis of large images compared to GANs (Karras et al., 2018; Brock et al., 2019). Recently, Kingma and Dhariwal (2018) proposed Glow as a generative flow with invertible 1 × 1 convolutions, which significantly improved the density estimation performance on natural images. Importantly, they demonstrated that flow-based generative models optimized towards the plain
likelihood-based objective are capable of generating realistic high-resolution natural images efficiently. Prenger et al. (2018) investigated applying flow-based generative models to speech synthesis by combining Glow with WaveNet (Van Den Oord et al., 2016). Ziegler and Rush (2019) adopted variational inference to apply generative flows to discrete sequential data. Unfortunately, the density estimation performance of Glow on natural images remains behind autoregressive models, such as PixelRNN/CNN (Oord et al., 2016; Salimans et al., 2017), Image Transformer (Parmar et al., 2018), PixelSNAIL (Chen et al., 2017) and SPN (Menick and Kalchbrenner, 2019). There is also some work (Rezende and Mohamed, 2015; Kingma et al., 2016; Zheng et al., 2017) trying to apply flow to variational inference.

In this paper, we propose a novel architecture of generative flow, masked convolutional generative flow (MACow), which leverages masked convolutional neural networks (Oord et al., 2016). The bijective mapping between input and output variables is easily established while the computation of the determinant of the Jacobian remains efficient. Compared to inverse autoregressive flow (IAF) (Kingma et al., 2016), MACow offers stable training and efficient inference and synthesis by restricting the local connectivity in a small “masked” kernel as well as large receptive fields by stacking multiple layers of convolutional flows and using rotational ordering masks (§3.1). We also propose a fine-grained version of the multi-scale architecture adopted in previous flow-based generative models to further improve the performance (§3.2). Experimenting with four benchmark datasets for images, CIFAR-10, ImageNet, LSUN, and CelebA-HQ, we demonstrate the effectiveness of MACow as a density estimator by consistently achieving significant improvements over Glow on all the three datasets. When equipped with the variational dequantization mechanism (Ho et al., 2019), MACow considerably narrows the gap of the density estimation with autoregressive models (§4).

2 Flow-based Generative Models

In this section, we first setup notations, describe flow-based generative models, and review Glow (Kingma and Dhariwal, 2018) as it is the foundation for MACow.

2.1 Notations

Throughout the paper, uppercase letters represent random variables and lowercase letters for realizations of their corresponding random variables. Let $X \in \mathcal{X}$ be the random variables of the observed data, e.g., $X$ is an image or a sentence for image and text generation, respectively.

Let $P$ denote the true distribution of the data, i.e., $X \sim P$, and $D = \{x_1, \ldots, x_N\}$ be our training sample, where $x_i, i = 1, \ldots, N$, are usually i.i.d. samples of $X$. Let $\mathcal{P} = \{P_\theta : \theta \in \Theta\}$ denote a parametric statistical model indexed by the parameter $\theta \in \Theta$, where $\Theta$ is the parameter space. $p$ denotes the density of the corresponding distribution $P$. In the deep generative model literature, deep neural networks are the most widely used parametric models. The goal of generative models is to learn the parameter $\theta$ such that $P_\theta$ can best approximate the true distribution $P$. In the context of maximum likelihood estimation, we minimize the negative log-likelihood of the parameters with:

$$\min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^{N} -\log p_\theta(x_i) = \min_{\theta \in \Theta} E_{\tilde{P}(X)}[-\log p_\theta(X)],$$

where $\tilde{P}(X)$ is the empirical distribution derived from training data $D$.

2.2 Flow-based Models

In the framework of flow-based generative models, a set of latent variables $Z \in \mathcal{Z}$ are introduced with a prior distribution $p_Z(z)$, which is typically a simple distribution like a multivariate Gaussian. For a bijection function $f : \mathcal{X} \rightarrow \mathcal{Z}$ (with $g = f^{-1}$), the change of the variable formula defines the model distribution on $X$ by

$$p_\theta(x) = p_Z(f_\theta(x)) \left| \det \left( \frac{\partial f_\theta(x)}{\partial x} \right) \right|,$$

where $\frac{\partial f_\theta(x)}{\partial x}$ is the Jacobian of $f_\theta$ at $x.$
The generative process is defined straightforwardly as the following:

\[ z \sim p_Z(z) \]
\[ x = g_\theta(z) \]  
(3)

Flow-based generative models focus on certain types of transformations \( f_\theta \) that allow the inverse functions \( g_\theta \) and Jacobian determinants to be tractable to compute. By stacking multiple such invertible transformations in a sequence, which is also called a (normalizing) flow (Rezende and Mohamed, 2015), the flow is then capable of warping a simple distribution \( p_Z(z) \) into a complex one \( (p(x)) \) through:

\[ X \xleftarrow{f_1} H_1 \xleftarrow{f_2} H_2 \xleftarrow{f_3} \ldots \xleftarrow{f_K} Z, \]

where \( f = f_1 \circ f_2 \circ \ldots \circ f_K \) is a flow of \( K \) transformations. For brevity, we omit the parameter \( \theta \) from \( f_\theta \) and \( g_\theta \).

2.3 Glow

Recently, several types of invertible transformations emerged to enhance the expressiveness of flows, among which Glow (Kingma and Dhariwal, 2018) has stood out for its simplicity and effectiveness on both density estimation and high-fidelity synthesis. The following briefly describes the three types of transformations that comprise Glow.

**Actnorm.** Kingma and Dhariwal (2018) proposed an activation normalization layer (Actnorm) as an alternative for batch normalization (Ioffe and Szegedy, 2015) to alleviate the challenges in model training. Similar to batch normalization, Actnorm performs an affine transformation of the activations using a scale and bias parameter per channel for 2D images, such that

\[ y_{i,j} = s \odot x_{i,j} + b, \]

where both \( x \) and \( y \) are tensors of shape \([h \times w \times c]\) with spatial dimensions \((h, w)\) and channel dimension \( c \).

**Invertible \( 1 \times 1 \) convolution.** To incorporate a permutation along the channel dimension, Glow includes a trainable invertible \( 1 \times 1 \) convolution layer to generalize the permutation operation as:

\[ y_{i,j} = W x_{i,j}, \]

where \( W \) is the weight matrix with shape \( c \times c \).

**Affine Coupling Layers.** Following Dinh et al. (2016), Glow includes affine coupling layers in its architecture of:

\[ x_a, x_b = \text{split}(x) \]
\[ y_a = x_a \]
\[ y_b = s(x_a) \odot x_b + b(x_a) \]
\[ y = \text{concat}(y_a, y_b), \]

where \( s(x_a) \) and \( b(x_a) \) are outputs of two neural networks with \( x_a \) as input. The \( \text{split()} \) and \( \text{concat()} \) functions perform operations along the channel dimension.

From this designed architecture of Glow, we see that interactions between spatial dimensions are incorporated only in the coupling layers. The coupling layer, however, is typically costly for memory resources, making it infeasible to stack a significant number of coupling layers into a single model, especially when processing high-resolution images. The main goal of this work is to design a new type of transformation that simultaneously models the dependencies in both the spatial and channel dimensions while maintaining a relatively small memory footprint to improve the capacity of the generative flow.

3 Masked Convolutional Generative Flows

In this section, we describe the architectural components of the masked convolutional generative flow (MACow). First, we introduce the proposed flow transformation using masked convolutions in §3.1. Then, we present a fine-grained version of the multi-scale architecture adopted by previous generative flows (Dinh et al., 2016; Kingma and Dhariwal, 2018) in §3.2.
3.1 Flow with Masked Convolutions

Applying autoregressive models to normalizing flows has been previously explored in studies (Kingma et al., 2016; Papamakarios et al., 2017), with idea of sequentially modeling the input random variables in an autoregressive order to ensure the model cannot read input variables behind the current one:

\[ y_t = s(x_{<t}) \odot x_t + b(x_{<t}), \]  

(4)

where \( x_{<t} \) denotes the input variables in \( x \) positioned ahead of \( x_t \) in the autoregressive order. \( s() \) and \( b() \) are two autoregressive neural networks typically implemented using spatial masks (Germain et al., 2015; Oord et al., 2016).

Despite effectiveness in high-dimensional space, autoregressive flows suffer from two crucial problems: (1) The training procedure is unstable when stacking multiple layers to increase the flow capacities for complex distributions. (2) Inference and synthesis are inefficient, due to the non-parallelizable inverse function.

We propose to use masked convolutions to restrict the local connectivity in a small “masked” kernel to address these two problems. The two autoregressive neural networks, \( s() \) and \( b() \), are implemented with one-layer masked convolutional networks with small kernels (e.g. \( 2 \times 5 \) in Figure 1) to ensure they only read contexts in a small neighborhood based on:

\[ s(x_{<t}) = s(x_{t*}), \quad b(x_{<t}) = b(x_{t*}), \]  

(5)

where \( x_{t*} \) denotes the input variables, restricted in a small kernel, on which \( x_t \) depends. By using masks in rotational ordering and stacking multiple layers of flows, the model captures a large receptive field (see Figure 1), and models dependencies in both the spatial and channel dimensions.

Efficient Synthesis. As discussed above, synthesis from autoregressive flows is inefficient since the inverse must be computed by sequentially traversing through the autoregressive order. In the context of 2D images with shape \([h \times w \times c]\), the time complexity of synthesis is quadratic, i.e. \( O(h \times w \times \text{NN}(h, w, c)) \), where \( \text{NN}(h, w, c) \) is the time of computing the outputs from the neural network \( s() \) and \( b() \) with input shape \([h \times w \times c]\). In our proposed flow with masked convolutions, computation of \( x_{i,j} \) begins as soon as all \( x_{t*} \) are available, contrary to the autoregressive requirement that all \( x_{<i,j} \) must have been already computed. Moreover, at each step we only need to feed a slice of the image (with shape \([h \times kw \times c]\) or \([kh \times w \times c]\) depending on the direction of the mask) into \( s() \) and \( b() \). Here \([kh \times kw \times c]\) is the shape of the kernel in the convolution. Thus, the time complexity reduces significantly from quadratic to linear, which is \( O(h \times \text{NN}(kh, w, c)) \) or \( O(w \times \text{NN}(kw, h, c)) \) for horizontal and vertical masks, respectively.

Discussion. The previous work closely related to MACOW is the Emerging Convolutions proposed in Hoogeboom et al. (2019). There are two main differences. i) the pattern of the mask is different. Emerging Convolutions use “causal masks” (Oord et al., 2016) whose inverse function falls into a complete autoregressive transformation. In contrast, MACOW achieves significantly more efficient inference and sampling (§4.3), due to the carefully designed masks (Figure 1). ii) The Emerging Convolutional Flow, proposed as an alternative to the invertible \( 1 \times 1 \) convolution in Glow, is basically a linear transformation with masked convolutions, which does not introduce “nonlinearity” to the random variables. MACOW, however, introduces such nonlinearity similar to the coupling layers.
3.2 Fine-grained Multi-Scale Architecture

Dinh et al. (2016) proposed a multi-scale architecture using a squeezing operation, which has been demonstrated to be helpful for training very deep flows. In the original multi-scale architecture, the model factors out half of the dimensions at each scale to reduce computational and memory costs. In this paper, inspired by the size upscaling in subscale ordering (Menick and Kalchbrenner, 2019) that generates an image as a sequence of sub-images with equal size, we propose a fine-grained multi-scale architecture to improve model performance further. In this fine-grained multi-scale architecture, each scale consists of $M/2$ blocks, and after each block, the model splits out $1/M$ dimensions of the input. Figure 2 illustrates the graphical specification of the two architecture versions. Note that the fine-grained architecture reduces the number of parameters compared with the original architecture with the same number of flow layers. Experimental improvements demonstrate the effectiveness of the fine-grained multi-scale architecture (§4).

4 Experiments

We evaluate our MACow model on both low-resolution and high-resolution datasets. For a step of MACow, we use $T = 2$ masked convolution units, and the Glow step is the same as that described in Kingma and Dhariwal (2018) where an ActNorm is followed by an Invertible $1 \times 1$ convolution, which is followed by a coupling layer. Each coupling layer includes three convolution layers where the first and last convolutions are $3 \times 3$, while the center convolution is $1 \times 1$. For low-resolution images, we use affine coupling layers with 512 hidden channels, while for high-resolution images we use additive layers with 256 hidden channels to reduce memory cost. ELU (Clevert et al., 2015) is used as the activation function throughout the flow architecture. For variational dequantization, the dequantization noise distribution $q_{\phi}(u|x)$ is modeled with a conditional MACow with shallow architecture. Additional details on architectures, results, and analysis of the conducted experiments are provided in Appendix B.

4.1 Low-Resolution Images

We begin our experiments with an evaluation of the density estimation performance of MACow on two low-resolution image datasets that are commonly used to evaluate the deep generative models: CIFAR-10 with images of size $32 \times 32$ (Krizhevsky and Hinton, 2009) and the $64 \times 64$ downscaled version of ImageNet (Oord et al., 2016).

We perform experiments to dissect the effectiveness of each component of our MACow model with ablation studies. The Org model utilizes the original multi-scale architecture, while the +fine-grained model augments the original one with the fine-grained multi-scale architecture proposed in §3.2. The

---

1 In our experiments, we set $M = 4$. Note that the original multi-scale architecture is a special case of the fine-grained version with $M = 2$. 
Table 1: Density estimation performance on CIFAR-10 $32 \times 32$ and ImageNet $64 \times 64$. Results are reported in bits/dim.

<table>
<thead>
<tr>
<th>Model</th>
<th>CIFAR-10</th>
<th>ImageNet-64</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Autoregressive</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAF VAE (Kingma et al., 2016)</td>
<td>3.11</td>
<td>–</td>
</tr>
<tr>
<td>Parallel Multiscale (Reed et al., 2017)</td>
<td>–</td>
<td>3.70</td>
</tr>
<tr>
<td>PixelRNN (Oord et al., 2016)</td>
<td>3.00</td>
<td>3.63</td>
</tr>
<tr>
<td>Gated PixelCNN (van den Oord et al., 2016)</td>
<td>3.03</td>
<td>3.57</td>
</tr>
<tr>
<td>MAE (Ma et al., 2019)</td>
<td>2.95</td>
<td>–</td>
</tr>
<tr>
<td>PixelCNN++ (Salimans et al., 2017)</td>
<td>2.92</td>
<td>–</td>
</tr>
<tr>
<td>PixelSNAIL (Chen et al., 2017)</td>
<td><strong>2.85</strong></td>
<td><strong>3.52</strong></td>
</tr>
<tr>
<td>SPN (Menick and Kalchbrenner, 2019)</td>
<td>–</td>
<td><strong>3.52</strong></td>
</tr>
<tr>
<td><strong>Flow-based</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Real NVP (Dinh et al., 2016)</td>
<td>3.49</td>
<td>3.98</td>
</tr>
<tr>
<td>Glow (Kingma and Dhariwal, 2018)</td>
<td>3.35</td>
<td>3.81</td>
</tr>
<tr>
<td>Flow++: Unif (Ho et al., 2019)</td>
<td>3.29</td>
<td>–</td>
</tr>
<tr>
<td>Flow++: Var (Ho et al., 2019)</td>
<td><strong>3.09</strong></td>
<td>3.69</td>
</tr>
<tr>
<td>MACOW: Org</td>
<td>3.31</td>
<td>3.78</td>
</tr>
<tr>
<td>MACOW: +fine-grained</td>
<td>3.28</td>
<td>3.75</td>
</tr>
<tr>
<td>MACOW: +var</td>
<td>3.16</td>
<td><strong>3.69</strong></td>
</tr>
</tbody>
</table>

Table 2: Negative log-likelihood scores for 5-bit LSUN and CelebA-HQ datasets in bits/dim.

<table>
<thead>
<tr>
<th>Model</th>
<th>CelebA-HQ</th>
<th>LSUN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>bedroom</td>
<td>tower</td>
</tr>
<tr>
<td>Glow (Kingma and Dhariwal, 2018)</td>
<td>1.03</td>
<td>1.20</td>
</tr>
<tr>
<td>SPN (Menick and Kalchbrenner, 2019)</td>
<td><strong>0.61</strong></td>
<td>–</td>
</tr>
<tr>
<td>MACOW: Unif</td>
<td>0.95</td>
<td>1.16</td>
</tr>
<tr>
<td>MACOW: Var</td>
<td>0.67</td>
<td><strong>0.98</strong></td>
</tr>
</tbody>
</table>

The +var model further implements the variational dequantization on the top of +fine-grained to replace the uniform dequantization (see Appendix A for details). For each ablation, we slightly adjust the number of steps in each level so that all the models have a similar number of parameters.

Table 1 provides the density estimation performance for different variations of our MACOW model along with the top-performing autoregressive models (first section) and flow-based generative models (second section). First, on both datasets, fine-grained models outperform Org ones, demonstrating the effectiveness of the fine-grained multi-scale architecture. Second, with the uniform dequantization, MACOW combined with the fine-grained multi-scale architecture significantly improves the performance over Glow on both datasets, and obtains slightly better results than Flow++ on CIFAR-10. In addition, with variational dequantization, MACOW achieves comparable result in bits/dim with Flow++ on ImageNet $64 \times 64$. On CIFAR-10, however, the performance of MaCow is around 0.07 bits/dim behind Flow++.

Compared with the state-of-the-art autoregressive generative models PixelSNAIL (Chen et al., 2017) and SPN (Menick and Kalchbrenner, 2019), the performance of MACOW is approximately 0.31 bits/dim worse on CIFAR-10 and 0.14 worse on ImageNet $64 \times 64$. Further improving the density estimation performance of MACOW on natural images is left to future work.

### 4.2 High-Resolution Images

We next demonstrate experimentally that our MACOW model is capable of high fidelity samples at high-resolution. Following Kingma and Dhariwal (2018), we choose the CelebA-HQ dataset (Karras et al., 2018), which consists of 30,000 high-resolution images from the CelebA dataset (Liu et al., 2015), and the LSUN (Yu et al., 2015) datasets including categories bedroom, tower and church. We train our models on 5-bit images with the fine-grained multi-scale architecture and both the uniform and variational dequantization. For each model, we adjust the number of steps in each level so that all the models have similar numbers of parameters with Glow for a fair comparison.
4.2.1 Density Estimation

Table 2 illustrates the negative log-likelihood scores in bits/dim of two versions of MACow on the 5-bit 128 × 128 LSUN and 256 × 256 CelebA-HQ datasets. With uniform dequantization, MACow improves the log-likelihood over Glow from 1.03 bits/dim to 0.95 bits/dim on CelebA-HQ dataset. Equipped with variational dequantization, MACow obtains 0.67 bits/dim, which is 0.06 bits/dim behind the state-of-the-art autoregressive generative model SPN (Menick and Kalchbrenner, 2019) and significantly narrows the gap. On the LSUN datasets, MACow with uniform dequantization outperforms Glow with 0.4 bits/dim on the bedroom category. With variational dequantization, the model achieves further improvements on all the three categories of LSUN datasets.

4.2.2 Image Generation

Consistent with previous work on likelihood-based generative models (Parmar et al., 2018; Kingma and Dhariwal, 2018), we found that sampling from a reduced-temperature model often results in higher-quality samples. Figure 3 showcases some random samples for 5-bit CelebA-HQ 256 × 256 with temperature 0.7 and LSUN 128 × 128 with temperature 0.9. The images are extremely high
quality for non-autoregressive likelihood models, despite that maximum likelihood is a principle that values diversity over sample quality in a limited capacity setting (Theis et al., 2016). More samples of images, including samples of low-resolution ones, are provided in Appendix C.

4.3 Comparison on Synthesis Speed

In this section, we compare the synthesis speed of MACow at test time with that of Glow (Kingma and Dhariwal, 2018), Masked Autoregressive Flows (MAF) (Papamakarios et al., 2017) and Emerging Convolutions (Hoogeboom et al., 2019). Following Hoogeboom et al. (2019), we measure the time to sample a datapoint when computed in mini-batches with size 100. For fair comparison, we re-implemented Glow using PyTorch (Paszke et al., 2017), and all experiments are conducted on a single NVIDIA TITAN X GPU.

Table 3a shows the sampling speed of MACow on CIFAR-10, together with that of the baselines. MACow is 7.3 times slower than Glow, much faster than Emerging Convolution and MAF, whose factors are 360 and 600 respectively. The sampling speed of MACow on datasets with different image sizes is shown in Table 3b. We see that the time of synthesis increases approximately linearly with the increase of image resolution.

5 Conclusion

In this paper, we propose a new type of generative flow, coined MACow, which exploits masked convolutional neural networks. By restricting the local dependencies in a small masked kernel, MACow boasts fast and stable training as well as efficient sampling. Experiments on both low- and high-resolution benchmark datasets of images show the capability of MACow on both density estimation and high-fidelity generation, achieving state-of-the-art or comparable likelihood as well as its superior quality of samples compared to previous top-performing models.

A potential direction for future work is to extend MACow to other forms of data, in particular text, on which no attempt (to the best of our knowledge) has been made to apply flow-based generative models. Another exciting direction is to combine MACow with variational inference to automatically learn meaningful (low-dimensional) representations from raw data.

References


2The reduced-temperature sampling is only applied to LSUN and CelebA-HQ 5-bits images, where MACow adopts additive coupling layers. For CIFAR-10 and ImageNet 8-bits images, we sample with temperature 1.0.

3https://github.com/XuezheMax/macow


