PixelSNAIL: An Improved Autoregressive Generative Model

Xi Chen[1 2] **Nikhil Mishra**[1 2] **Mostafa Rohaninejad**[1 2] **Pieter Abbeel**[1 2]

Abstract

Autoregressive generative models achieve the best results in density estimation tasks involving high dimensional data, such as images or audio. They pose density estimation as a sequence modeling task, where a recurrent neural network (RNN) models the conditional distribution over the next element conditioned on all previous elements. In this paradigm, the bottleneck is the extent to which the RNN can model long-range dependencies, and the most successful approaches rely on causal convolutions. Taking inspiration from recent work in meta reinforcement learning, where dealing with long-range dependencies is also essential, we introduce a new generative model architecture that combines causal convolutions with self attention. In this paper, we describe the resulting model and present state-of-the-art log-likelihood results on heavily benchmarked datasets: CIFAR-10 (2.85 bits per dim), 32 × 32 ImageNet (3.80 bits per dim) and 64 × 64 ImageNet (3.52 bits per dim). Our implementation will be made available at `anonymized`.

1. Introduction

Autoregressive generative models over high-dimensional data $\mathbf{x} = (x_1, \ldots, x_n)$ factor the joint distribution as a product of conditionals:

$$p(\mathbf{x}) = p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i | x_1, \ldots, x_{i-1})$$

A recurrent neural network (RNN) is then trained to model $p(x_i|x_{1:i-1})$. Optionally, the model can be conditioned on additional global information $\mathbf{h}$ (such as a class label, when applied to images), in which case it in models $p(x_i|x_{1:i-1}, \mathbf{h})$. Such methods are highly expressive and allow modeling complex dependencies. Compared to GANs (Goodfellow et al., 2014), neural autoregressive models offer tractable likelihood computation and ease of training, and have been shown to outperform latent variable models (van den Oord et al., 2016c;b; Salimans et al., 2017).

The main design consideration is the neural network architecture used to implement the RNN, as it must be able to easily refer to earlier parts of the sequence. A number of possibilities exist:

- Traditional RNNs, such as GRUs or LSTMs: these propagate information by keeping it in their hidden state from one timestep to the next. This temporally-linear dependency significantly inhibits the extent to which they can model long-range relationships in the data.
- Causal convolutions (van den Oord et al., 2016b; Salimans et al., 2017): these apply convolutions over the sequence (masked or shifted so that the current prediction is only influenced by previous element). They offer high-bandwidth access to the earlier parts of the sequence. However, their receptive field has a finite size, and still experience noticeable attenuation with regards to elements far away in the sequence.
- Self-attention (Vaswani et al., 2017): these models turn the sequence into an unordered key-value store that can be queried based on content. They feature an unbounded receptive field and allow undeteriorated access to information far away in the sequence. However, they only offer pinpoint access to small amounts of information, and require additional mechanism to incorporate positional information.

Causal convolutions and self-attention demonstrate complementary strengths and weaknesses: the former allow high bandwidth access over a finite context size, and the latter allow access over an infinitely large context. Interleaving the two thus offers the best of both worlds, where the model can have high-bandwidth access without constraints on the amount of information it can effectively use. The convolutions can be seen as aggregating information to build the context over which to perform an attentive lookup. Using this approach (dubbed SNAIL), Mishra et al. (2017) demonstrated significant performance improvements on a

[1] covariant.ai [2] UC Berkeley, EECS Dept.. Correspondence to: Xi Chen <peter@covariant.ai>.

Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

number of tasks in meta-learning setup, where the challenge of long-term temporal dependencies is also prevalent, as an agent should be able to adapt its behavior based on past experience.

In this paper, we consider the task of autoregressive generative modeling by taking inspirations from SNAIL, as the fundamental bottleneck of access to past information is the same. Building off the current state-of-the-art in generative models, a class of convolution-based architectures known as PixelCNNs (van den Oord et al. (2016b) and Salimans et al. (2017)), we present a new architecture, PixelSNAIL, that incorporates ideas from (Mishra et al., 2017) to obtain state-of-the-art results on the heavily benchmarked CIFAR-10, Imagenet 32×32 and Imagenet 64×64 datasets.

2. Methodology

For self-containedness, we first review the formulation of modeling high-dimensional natural images by neural autoregressive models and describe prior works' strengths and weaknesses. Next, we elaborate on the design principles behind PixelSNAIL and introduce a family of architectures that achieves good performance.

2.1. Neural Autoregressive Image Modeling

Natural images are usually represented as 3-dimensional random variables $Height \times Width \times 3$, where 3 color channels (RGB) are recorded at each location. To model such a random variable autoregressively, one can first impose an ordering and then factor the joint distribution as a product of conditionals over that ordering:

$$p(\mathbf{x}) = p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i | x_1, \ldots, x_{i-1})$$

For natural images, most prior works have chosen to use the "raster scan" ordering (Oord et al., 2016b; van den Oord et al., 2016b; Salimans et al., 2017), where along each row left pixels come before right pixels and top rows come before bottom rows.

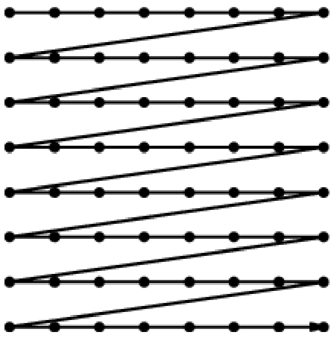

Figure 1. Raster Scan Ordering

State-of-the-art neural autoregressive models employ causal convolution models to represent the conditional distributions (van den Oord et al., 2016b; Salimans et al., 2017). In this type of architecture, the initial image $\mathbf{x}$ is processed through a series of causal convolutions and the outcome is a 3D tensor that has shape $Height \times Width \times Channels$, where at each spatial location (x, y) a vector of length $Channels$ describes the sufficient statistics for the conditional $p(x_i | x_{\leq i-1})|_{i=x*Width+y}$.

In order for the probability model to be valid (and causal), the conditional distribution for x_i should only depend on pixel values before i. Such constraints are enforced via either masked convolution (Oord et al., 2016b) or shift-based convolution (van den Oord et al., 2016b). In masked convolution (illustrated in Figure 2), a normal convolution is applied but the filter is masked in such a way that it cannot depends on values at current or later pixel locations:

1	1	1	1	1
1	1	1	1	1
1	1	0	0	0
0	0	0	0	0
0	0	0	0	0

Figure 2. An example masked 5×5 filter

van den Oord et al. (2016b) pointed out that masked convolutions, though causal, are limited in terms of expressiveness since they create blind spots in the receptive field. To address this problem, they introduced shift-based convolutions: at each layer, ordinary convolutions are applied, and then the whole feature map is shifted to maintain causality.

One of the benefits of using causal convolution architectures is that, given a single image $\mathbf{x}$, all the conditional distributions can be calculated in just one forward pass. Since all conditionals are calculated in parallel through highly optimized convolution operations, causal convolution architectures are efficient and scalable to high-dimensional density modeling problems.

However, convolution operations, by nature, only aggregate information locally. In order to model long-range dependencies,the receptive field must grow by repeatedly applying convolutions. Noticing this problem, van den Oord et al. (2016a) and Salimans et al. (2017) respectively proposed to use dilated convolutions and strided convolutions (followed by corresponding upsampling) to achieve faster receptive field growth. The resulting improvements in density-estimation performance suggest that improving the model's ability to capture long-range dependencies is essential.

One should note that, even with dilated convolutions or strided convolutions, information access to remote pixel locations is still limited: the information needs to be relayed through a series of intermediate locations since each convolution operation only operates in a limited context. We will explore architectural decisions that offer better information access to pixels far away from any conditional distribution and show that the improved ability to model long-range statistics leads to better density modelling performance.

Even though, all prior works use raster scan ordering (to the best of our knowledge), it's worth noting that any ordering is equivalent in expressiveness: for any arbitrary ordering, the joint distribution over $\mathbf{x}$ can be expressed as a product of conditionals. However, for particular ordering choices, the conditional $p(x_i|x_1, \ldots, x_{i-1})$ might be a complex distribution that our current modeling tools, like convolutional networks, are incapable of expressing. As such, it could be beneficial to explore other orderings that can give rise to conditional distributions that are easier to learn.

We know that the conditional distribution of a pixel location is mostly influenced by the values of its neighboring pixels (Salimans et al., 2017) but the widely used raster scan ordering only has a small number neighboring pixels available in the conditioning context $x_1, \ldots, x_{i-1}$: only to the left and above and most of the context is wasted on regions that might have little correlation with the current pixel like the far top-right corner. One possible alternative is zigzag ordering, which allows each conditional distribution to depend on pixels to the left and above:

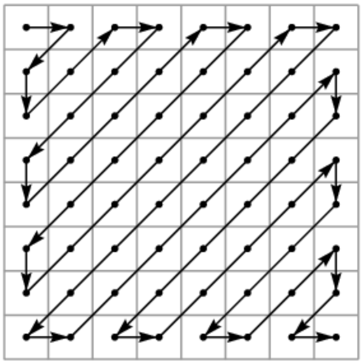

Figure 3. Zigzag Ordering

However, one will notice that when such an ordering will introduce blind spots when combined with combined masked or shift-based convolutional architectures.

Motivated by these issues, we introduce the PixelSNAIL model family, which generalizes the causal convolution architectures discussed thus far by allowing a much larger and more flexible receptive field. As a result, PixelSNAIL models achieve superior modeling performance.

2.2. PixelSNAIL

The key idea behind PixelSNAIL is to introduce attention blocks, in a style similar to Self Attention in (Vaswani et al., 2017; Mishra et al., 2017), into neural autoregressive modelling. As explained previously, the ability to model long-range dependencies is crucial to performance, so it's natural to use attention blocks to equip all conditionals with the ability to refer to all of their available context.

An attention block applies one key-value lookup for the feature vector at every spatial location and the lookups are done for all spatial locations in parallel to exploit GPU parallelism.

Concretely, an attention block that has type $H \times W \times C_1 \rightarrow H \times W \times C_2$ defines 3 functions that operate on feature vectors:

- $f_{\text{key}}(x) :: C_1 \rightarrow \text{Dim}_{\text{key}}$
- $f_{\text{query}}(x) :: C_1 \rightarrow \text{Dim}_{\text{key}}$
- $f_{\text{value}}(x) :: C_1 \rightarrow C_2$

According to some autoregressive ordering, we can name the feature vectors of a 2D feature map, $\mathbf{y}$, as $y_1, y_2, \cdots, y_N$. Then for $\mathbf{z} = \text{attention}(\mathbf{y})$, the mapping is defined as:

$$z_i = \sum_{j<i} p_{ij} f_{\text{value}}(y_j)$$

where

$$p_i = \text{softmax}([f_{\text{key}}(y_1)^T f_{\text{query}}(y_i), \cdots, f_{\text{key}}(y_{i-1})^T f_{\text{query}}(y_i)])$$

Each conditional can access any pixels in its context through the attention operator (notice the summation over all available context: $\sum_{j<i}$), easy information access of remote pixels improves modeling of long-range statistics.

Note also that the autoregressive ordering is enforced only in the summation step and hence, in implementation, one can simply mask out entries that shouldn't be summed over to make the operator causal. This kind of masking scheme is also very flexible and can permit, for instance, the Zigzag ordering discussed above.

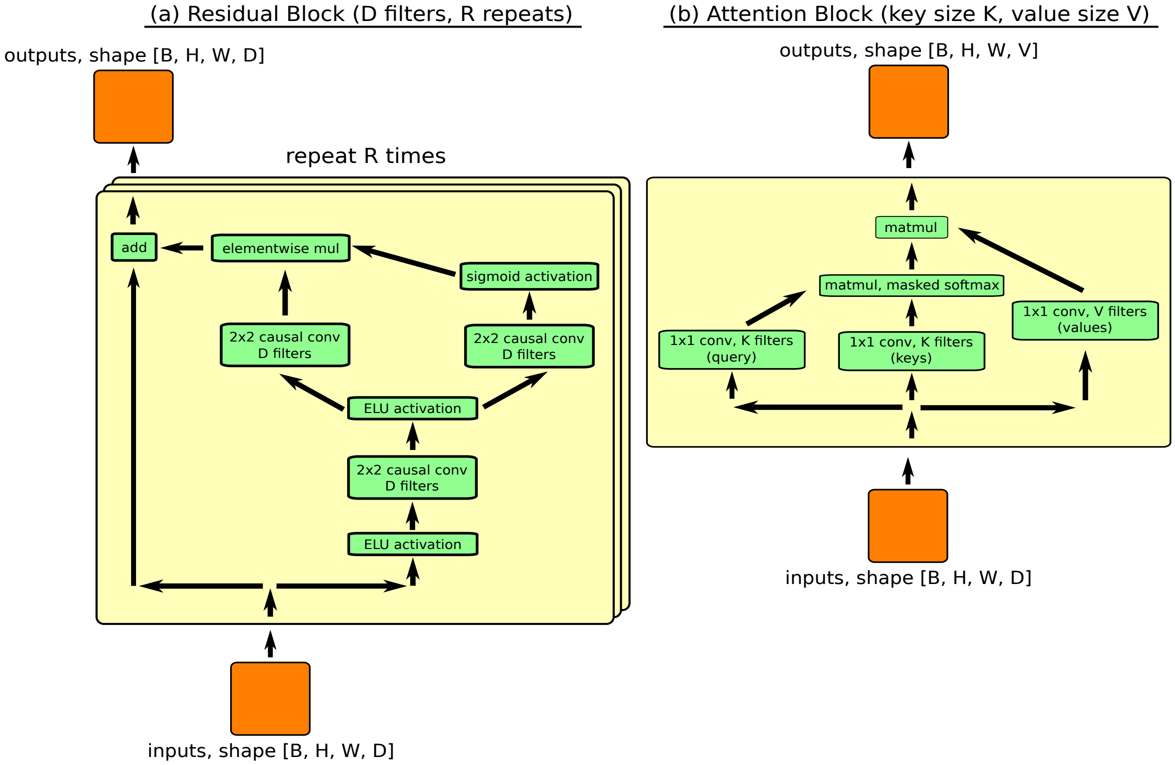
(a) Residual Block (D filters, R repeats)
outputs, shape [B, H, W, D]
repeat R times
add
elementwise mul
sigmoid activation
2x2 causal conv D filters
2x2 causal conv D filters
ELU activation
2x2 causal conv D filters
ELU activation
inputs, shape [B, H, W, D]
(b) Attention Block (key size K, value size V)
outputs, shape [B, H, W, V]
matmul
matmul, masked softmax
1x1 conv, V filters (values)
1x1 conv, K filters (query)
1x1 conv, K filters (keys)
inputs, shape [B, H, W, D]

Figure 4. The modular components that make up PixelSNAIL: (a) a residual block, and (b) an attention block. For both datasets, we used residual blocks with 256 filters and 4 repeats, and attention blocks with key size 16 and value size 128.

The PixelSNAIL model family is primarily composed of two building blocks, which are illustrated in Figure 4 and described below:

- A *residual block* applies several 2D-convolutions to its input, each with residual connections. To make them causal, the convolutions are masked or shifted so that the current pixel can only access pixels to the left and above from it. We use a gated activation function similar to (van den Oord et al., 2016b; Oord et al., 2016a). Throughout the model, we used 4 convolutions per block and 256 filters in each convolution.

- An *attention block* performs a single key-value lookup. It projects the input to a lower dimensionality to produce the keys and values and then uses softmax-attention like in (Vaswani et al., 2017; Mishra et al., 2017) (masked so that the current pixel can only attend over previously generated pixels). We used keys of size 16 and values of size 128.

Figure 5 illustrates the full PixelSNAIL architecture, which interleaves the residual blocks and attention blocks depicted in Figure 4. In the CIFAR-10 model only, we applied dropout of 0.5 after the first convolution in every residual block, to prevent overfitting. We did not use any dropout for ImageNet, as the dataset is much larger. On both datasets, we use Polyak averaging (Polyak & Juditsky, 1992) (following (Salimans et al., 2017)) over the training parameters. We used an exponential moving average weight of 0.9995 for CIFAR-10 and 0.9997 for ImageNet. As the output distribution, we use the discretized mixture of logistics introduced by Salimans et al. (2017), with 10 mixture components for CIFAR-10 and 32 for ImageNet. To predict the subpixel (red,green,blue) values, we used the same linear-autoregressive parametrization as Salimans et al. (2017).

To mitigate the problems of bad initialization, we employ weight Normalization with data-dependent initialization (Salimans & Kingma, 2016) in all experiments.

Our code will be made available, and can be found at: `https://github.com/neocxi/pixelsnail-public`.

3. Experiments

3.1. Long-range Dependency

In the previous section, we hypothesize that attention blocks make it easier to access information from a large context than causal convolutions. Here we conduct a simple diagnostic experiment to investigate this hypothesis. We choose

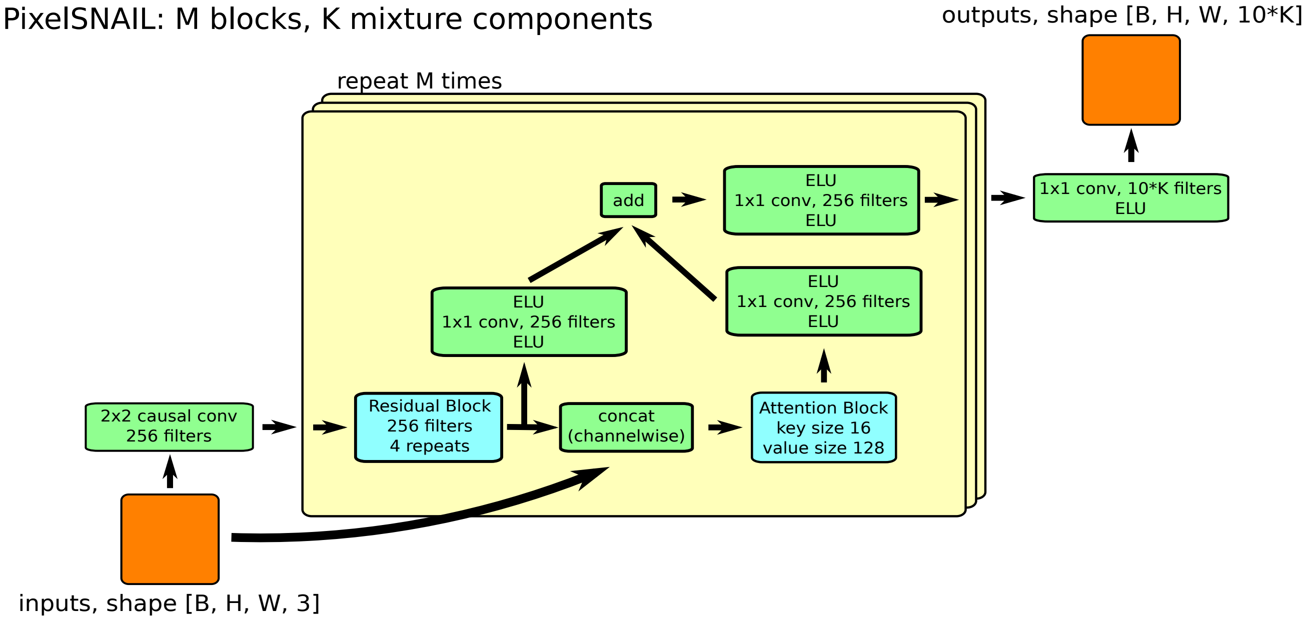

Figure 5. The entire PixelSNAIL model architecture, using the building blocks from Figure 4. We used 12 blocks for both datasets, with 10 mixture components for CIFAR-10 and 32 for ImageNet.

a conditional distribution $p(x_{15,15}|\cdots)$ at the center of the image, and we calculate the log probability's sensitivity to input image for a randomly initialized model:

$$\nabla_{\mathbf{x}} \log p(x_{(15,15)}|\cdots)$$

This test captures the first-order dependency between the inspected conditional distribution and all pixels in its conditioning context. We expect the gradient to be nonzero for pixel values that have an influence on the conditional distribution.

In Figure 6, we inspect a shift-based causal convolution model (Gated PixelCNN (van den Oord et al., 2016b)). We used a medium size model with 7 gated resnet blocks and 6M parameters:

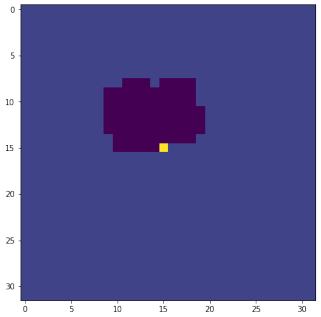

Figure 6. Gated PixelCNN. The yellow dot indicates the pixel under inspection, and dark purple dots indicate locations with derivative magnitude larger than 0.001.

We can observe that the receptive field is limited. On top of that, the "holes" within the theoretical receptive field limit attest to the the difficulty with which information propagates through long distances in this type of architecture.

Then we run the same experiment on PixelCNN++ (Salimans et al., 2017) with identical number of resnet blocks.

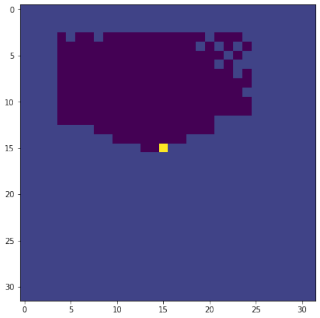

Figure 7. PixelCNN++

Here one can see that the strided convolutions introduced in PixelCNN++ effectively expand the receptive field. However, there are still "holes" within the theoretical limit, suggesting similar difficulty of information propogation.

Next we run the same test on a PixelSNAIL with 7 blocks (4 shift-based convolutions and 3 attention blocks).

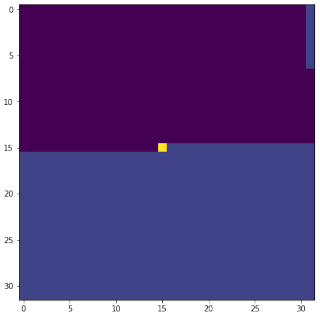

Figure 8. PixelSNAIL

PixelSNAIL achieves a much larger effective receptive field size for the same number of layers and fewer holes within theoretical receptive field limit. While this doesn't mean that a PixelSNAIL model relies on all of the available context *at convergence*, the existence of gradient signal means gradient descent can in principle guide a PixelSNAIL model to capture learn long-range dependencies.

We would like to stress that these tests are not conclusive. It's possible that other models have higher order dependency that's not visualized by gradient magnitude and it's also conceivable that during training, the effective receptive fields would change. We nevertheless believe that they provide valuable insights into the inductive biases encoded by different model architectures.

Lastly, we provide a visualization of the receptive field of a PixelSNAIL model that uses the Zigzag ordering instead of the raster scan ordering. This modification only required us to change 2 lines of code, but yet we see that PixelSNAIL is able to approach theoretical receptive field limit, despite the drastically different context shape.

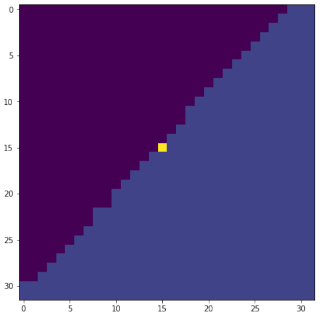

Figure 9. PixelSNAIL with Zigzag Ordering

3.2. Density Modelling Performance

In Table 1, we provide negative log-likelihood results (in bits per dim) for PixelSNAIL on both CIFAR-10, Imagenet 32×32 and Imagenet 64×64. We compare PixelSNAIL's performance to a number of autoregressive models. These include: (i) PixelRNN (Oord et al., 2016b), which uses LSTMs, (ii) PixelCNN (van den Oord et al., 2016b) and PixelCNN++ (Salimans et al., 2017), which only use causal convolutions, and (iii) Image Transformer (Anonymous, 2018), an attention-only architecture inspired by Vaswani et al. (2017). PixelSNAIL outperforms all of these approaches, which suggests that both causal convolutions and attention are essential components of the architecture. To maintain consistency with prior work, all of the PixelSNAIL results reported below use the raster scan ordering.

We would like to point out that, as of submission, the performance of ImageNet 32×32 and ImageNet 64×64 is still improving. Due to computational limits, we can only train these models on 4 GPUs but are able to outperform the previous state-of-the-art model that was trained on 32 GPUs (van den Oord et al., 2016b).

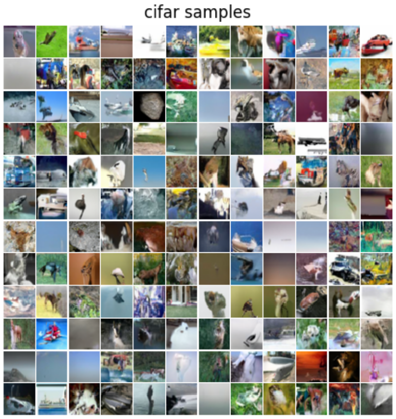

Figure 10. Samples from our CIFAR-10 model.

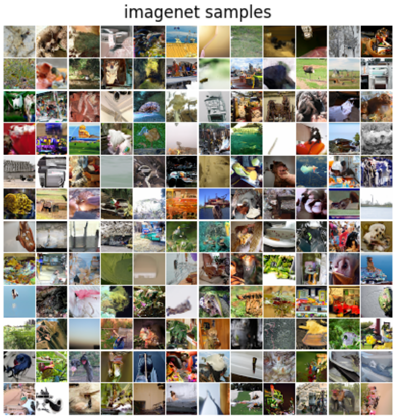

Figure 11. Samples from our 32×32 ImageNet model.

Table 1. Average negative log-likelihoods on CIFAR-10 and ImageNet 32 × 32, in bits per dim. PixelSNAIL outperforms other autoregressive models which only rely on causal convolutions xor self-attention.

Method	CIFAR-10	ImageNet 32 × 32	ImageNet 64 × 64
Conv DRAW (Gregor et al., 2016)	3.5	4.40	4.10
Real NVP (Dinh et al., 2016)	3.49	4.28	3.98
VAE with IAF (Kingma et al., 2016)	3.11	–	–
PixelRNN (Oord et al., 2016b)	3.00	3.86	3.63
Gated PixelCNN (van den Oord et al., 2016b)	3.03	3.83	3.57
Image Transformer (Anonymous, 2018)	2.98	3.81	–
PixelCNN++ (Salimans et al., 2017)	2.92	–	–
Block Sparse PixelCNN++ (OpenAI, 2017)	2.90	–	–
PixelSNAIL (ours)	**2.85**	**3.80**	**3.52**

4. Related Work

There is a large body of work on neural autoregressive models (Larochelle & Murray, 2011; Uria et al., 2013; Germain et al., 2015; Theis & Bethge, 2015; Oord et al., 2016b; van den Oord et al., 2016d). This type of autoregressive model was further explored for audio data (Oord et al., 2016a), video data (Kalchbrenner et al., 2016b) and language (Kalchbrenner et al., 2016a).

Within the domain of modelling natural images, different extensions to neural autoregressive models were also explored: (Reed et al., 2017) explored mixing parallel generation into autoregressive ordering to speed up generation time, but still mostly rely on raster scan ordering; (Kolesnikov & Lampert, 2017) proposed to decompose modeling of colorful images into two stages of first modeling grayscale images and then colorful images.

Other than autoregressive models, there are other natural generative models that provide tractable and exact likelihood computation. These models (Dinh et al., 2014; Rezende & Mohamed, 2015; Dinh et al., 2016) typically apply an invertible transformation that admits tractable determinant calculation to some continuous entropy source. It's worth noting that the architectural improvements proposed in this paper apply equally well to these invertible transformation and we believe it's an exciting area of future research.

Other than exact likelihood models, there are also a lot of works that model natural images with a Helmholtz Machine (Dayan et al., 1995) or variants thereof (Kingma & Welling, 2013; Rezende et al., 2014; de Freitas et al., 2001; Gregor et al., 2015b;a; Tran et al., 2015; Kingma et al., 2016) trained with approximate inference. And there is also a line of work that combined Helmholtz Machines with autoregressive models for images (Chen et al., 2016; Gulrajani et al., 2016) and for text data (Chung et al., 2015; Bowman et al., 2015; Fraccaro et al., 2016; Xu & Sun, 2016).

Among the implicit generative models that are trained without likelihood, GANs (Goodfellow et al., 2014) are the most popular models and generate the most realistic images. We refer readers to (Goodfellow, 2016) for a recent survey on this topic. GANs, based predominantly on CNNs, typically generate images that have realistic local texture but lack global coherence. It's possible that a PixelSNAIL style architecture will be able to improve global coherence.

Our work directly builds on top of (Mishra et al., 2017), which employs causal convolutions styled after Oord et al. (2016a) with self-attention (like in (Vaswani et al., 2017)) in the context of meta reinforcement learning (where the challenging of capturing long-range dependencies is also prevalent). Although Vaswani et al. (2017) utilized attention in the context of machine translation, a number of concurrent works have used the same technique in other domains. Wang et al. (2017) apply attention to video classification and activity recognition, and (Anonymous, 2018) use it for generative modeling of images. PixelSNAIL significantly outperforms the latter, which corroborates the findings in Mishra et al. (2017) that convolutions and attention complement each other well for modelling long-term dependencies.

5. Conclusion

We introduced PixelSNAIL, a class of autoregressive generative models that combine causal convolutions with self-attention. We demonstrate state-of-the-art density estimation performance on CIFAR-10, ImageNet 32 × 32 and ImageNet 64 × 64, with a publicly-available implementation at `https://github.com/neocxi/pixelsnail-public`.

Despite their tractable likelihood and strong empirical performance, one notable drawback of autoregressive generative models is that sampling is slow.PixelSNAIL's sampling speed is comparable to that of existing autoregressive models; the design of models that allow faster sampling without losing performance remains an open problem.

References

Anonymous. Image transformer. *Under review at the International Conference on Learning Representations (ICLR)*, 2018. URL https://openreview.net/forum?id=r16Vyf-0-.

Bowman, Samuel R, Vilnis, Luke, Vinyals, Oriol, Dai, Andrew M, Jozefowicz, Rafal, and Bengio, Samy. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

Chen, Xi, Kingma, Diederik P, Salimans, Tim, Duan, Yan, Dhariwal, Prafulla, Schulman, John, Sutskever, Ilya, and Abbeel, Pieter. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016.

Chung, Junyoung, Kastner, Kyle, Dinh, Laurent, Goel, Kratarth, Courville, Aaron C, and Bengio, Yoshua. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pp. 2980–2988, 2015.

Dayan, Peter, Hinton, Geoffrey E, Neal, Radford M, and Zemel, Richard S. The helmholtz machine. *Neural computation*, 7(5):889–904, 1995.

de Freitas, Nando, Højen-Sørensen, Pedro, Jordan, Michael I, and Russell, Stuart. Variational mcmc. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI'01, pp. 120–127, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-800-1.

Dinh, Laurent, Krueger, David, and Bengio, Yoshua. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Dinh, Laurent, Sohl-Dickstein, Jascha, and Bengio, Samy. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Fraccaro, Marco, Sønderby, Søren Kaae, Paquet, Ulrich, and Winther, Ole. Sequential neural models with stochastic layers. *arXiv preprint arXiv:1605.07571*, 2016.

Germain, Mathieu, Gregor, Karol, Murray, Iain, and Larochelle, Hugo. Made: Masked autoencoder for distribution estimation. *arXiv preprint arXiv:1502.03509*, 2015.

Goodfellow, Ian. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Gregor, Karol, Danihelka, Ivo, Graves, Alex, and Wierstra, Daan. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015a.

Gregor, Karol, Danihelka, Ivo, Graves, Alex, and Wierstra, Daan. Draw: A recurrent neural network for image generation. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015b.

Gregor, Karol, Besse, Frederic, Rezende, Danilo Jimenez, Danihelka, Ivo, and Wierstra, Daan. Towards conceptual compression. *arXiv preprint arXiv:1604.08772*, 2016.

Gulrajani, Ishaan, Kumar, Kundan, Ahmed, Faruk, Taiga, Adrien Ali, Visin, Francesco, Vazquez, David, and Courville, Aaron. Pixelvae: A latent variable model for natural images. *arXiv preprint arXiv:1611.05013*, 2016.

Kalchbrenner, Nal, Espheholt, Lasse, Simonyan, Karen, Oord, Aaron van den, Graves, Alex, and Kavukcuoglu, Koray. eural machine translation in linear time. *arXiv preprint arXiv:1610.00527*, 2016a.

Kalchbrenner, Nal, Oord, Aaron van den, Simonyan, Karen, Danihelka, Ivo, Vinyals, Oriol, Graves, Alex, and Kavukcuoglu, Koray. Video pixel networks. *arXiv preprint arXiv:1610.00527*, 2016b.

Kingma, Diederik P and Welling, Max. Auto-Encoding Variational Bayes. *Proceedings of the 2nd International Conference on Learning Representations*, 2013.

Kingma, Diederik P., Salimans, Tim, Jozefowicz, Rafal, Chen, Xi, Sutskever, Ilya, and Welling, Max. Improving variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, 2016.

Kolesnikov, Alexander and Lampert, Christoph H. Pixelcnn models with auxiliary variables for natural image modeling. In *International Conference on Machine Learning*, pp. 1905–1914, 2017.

Larochelle, Hugo and Murray, Iain. *The Neural Autoregressive Distribution Estimator*. AISTATS, 2011.

Mishra, Nikhil, Rohaninejad, Mostafa, Chen, Xi, and Abbeel, Pieter. A simple neural attentive meta-learner. In *NIPS 2017 Workshop on Meta-Learning*, 2017.

Oord, Aaron van den, Dieleman, Sander, Zen, Heiga, Simonyan, Karen, Vinyals, Oriol, Graves, Alex, Kalchbrenner, Nal, Senior, Andrew, and Kavukcuoglu, Koray. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016a.

Oord, Aaron van den, Kalchbrenner, Nal, and Kavukcuoglu, Koray. Pixel recurrent neural networks. *International Conference on Machine Learning (ICML)*, 2016b.

OpenAI. Block-sparse gpu kernels, Dec 2017. URL https://blog.openai.com/block-sparse-gpu-kernels/.

Polyak, Boris T and Juditsky, Anatoli B. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

Reed, Scott E., van den Oord, Aäron, Kalchbrenner, Nal, Gómez, Sergio, Wang, Ziyu, Belov, Dan, and de Freitas, Nando. Parallel multiscale autoregressive density estimation. In *Proceedings of The 34th International Conference on Machine Learning*, 2017.

Rezende, Danilo and Mohamed, Shakir. Variational inference with normalizing flows. In *Proceedings of The 32nd International Conference on Machine Learning*, pp. 1530–1538, 2015.

Rezende, Danilo J, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, pp. 1278–1286, 2014.

Salimans, Tim and Kingma, Diederik P. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *arXiv preprint arXiv:1602.07868*, 2016.

Salimans, Tim, Karpathy, Andrej, Chen, Xi, and Kingma, Diederik P. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

Theis, Lucas and Bethge, Matthias. Generative image modeling using spatial lstms. In *Advances in Neural Information Processing Systems*, pp. 1927–1935, 2015.

Tran, Dustin, Ranganath, Rajesh, and Blei, David M. Variational gaussian process. *arXiv preprint arXiv:1511.06499*, 2015.

Uria, Benigno, Murray, Iain, and Larochelle, Hugo. Rnade: The real-valued neural autoregressive density-estimator. In *Advances in Neural Information Processing Systems*, pp. 2175–2183, 2013.

van den Oord, Aaron, Dieleman, Sander, Zen, Heiga, Simonyan, Karen, Vinyals, Oriol, Graves, Alex, Kalchbrenner, Nal, Senior, Andrew, and Kavukcuoglu, Koray. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016a.

van den Oord, Aaron, Kalchbrenner, Nal, Espeholt, Lasse, Vinyals, Oriol, Graves, Alex, et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems (NIPS)*, 2016b.

van den Oord, Aaron, Kalchbrenner, Nal, and Kavukcuoglu, Koray. Pixel recurrent neural networks. In *International Conference on Machine Learning (ICML)*, 2016c.

van den Oord, Aaron, Kalchbrenner, Nal, Vinyals, Oriol, Espeholt, Lasse, Graves, Alex, and Kavukcuoglu, Koray. Conditional image generation with pixelcnn decoders. *arXiv preprint arXiv:1606.05328*, 2016d.

Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Lukasz, and Polosukhin, Illia. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

Wang, Xiaolong, Girshick, Ross, Gupta, Abhinav, and He, Kaiming. Non-local neural networks. *arXiv preprint arXiv:1711.07971*, 2017.

Xu, Weidi and Sun, Haoze. Semi-supervised variational autoencoders for sequence classification. *arXiv preprint arXiv:1603.02514*, 2016.