CInC Flow: Characterizable Invertible 3×3 Convolution

Sandeep Nagar[1], Marius Dufraisse[2], Girish Varma[1]

sandeep.nagar@research.iiit.ac.in https://github.com/Naagar/Normalizing_Flow_3x3_inv

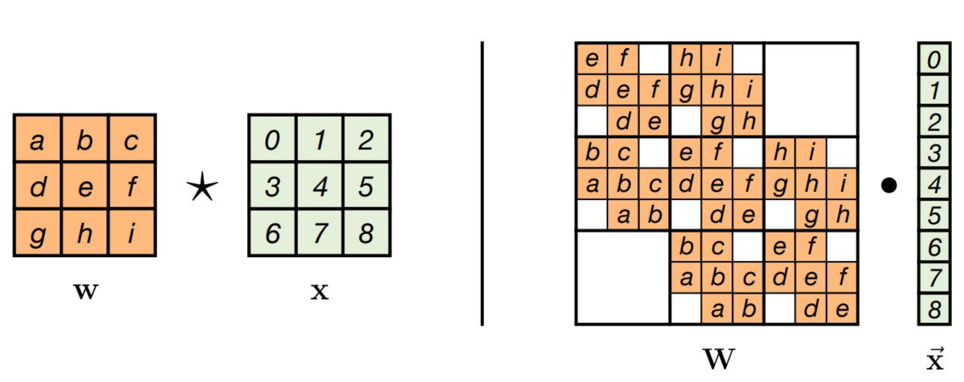
[1]Machine Learning Lab, IIIT Hyderabad, India. [2]CS Dept., École Normale Supérieure (ENS), Paris-Saclay, France.

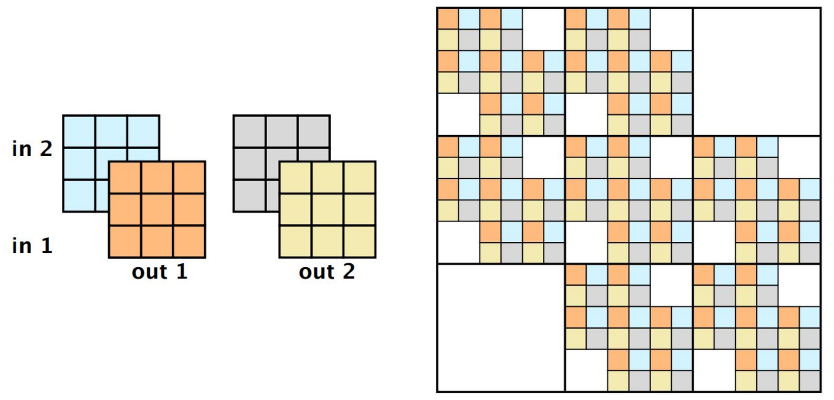
INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

école normale supérieure paris-saclay

the 4th tractable probabilistic modeling workshop tpm uai2021

Invertible Convolutions for Normalizing Flows





Figures adapted from Hoogeboom et al. Emerging Convolutions for Generative Normalizing Flows. ICML, 2019.

A 3x3 convolution with 1 in and out channels and its matrix form.

A 3x3 convolution with 2 in and out channels and its matrix form.

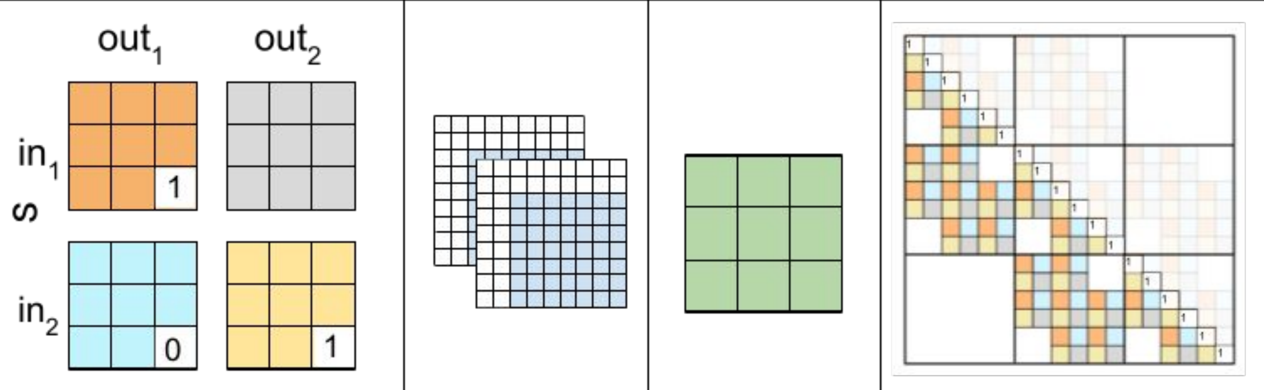
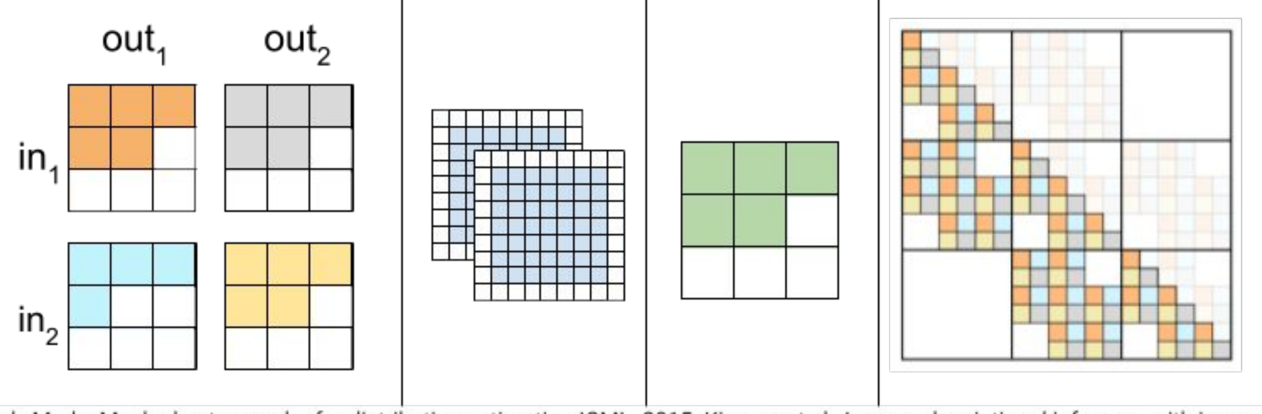
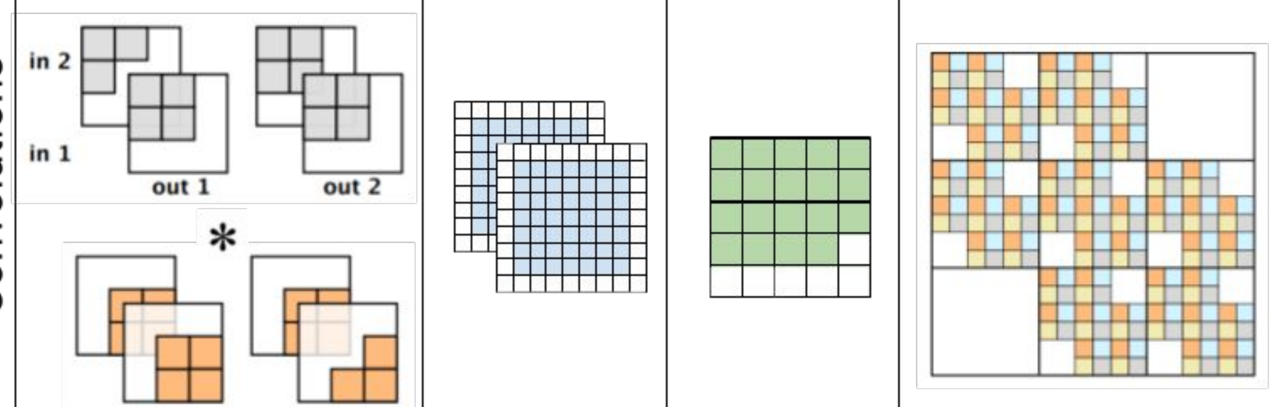
A general convolution need not be invertible and hence cannot be used for designing normalizing flows, which can be trained using Maximum Likelihood.

Goal: Design CNNs that are invertible, which can be used to build efficient and expressive normalizing flows..

Charecterizable Invertible Convolution (CInC)

We design a convolution which

1. is guaranteed to be invertible during training,
2. has more learnable parameters leading to better expressivity,
3. and is easy to implement efficiently.

	Filters / Padding / Receptive Field / Matrix	Observations n=#in,out channels.
CInC (Ours) Convolution		#learnable parameters $9n^2 - n(n-1)/2$ #convs = 1 Invertibility is guaranteed in training since diagonal entries of matrix are 1s.
Autoregressive Convolutions		#learnable parameters $5n^2$ #convs = 1 Number of learnable parameters are reduced by almost 50% resulting in lesser expressive power.
Emerging Convolutions		#learnable parameters $10n^2$ #convs = 2 Having more convolutions will increase runtime during generation as well as latent vector computation passes.

Germain et al., Made: Masked autoencoder for distribution estimation.ICML 2015. Kingma et al., Improved variational inference with inverse autoregressive flow, NIPS 2016.

Hoogeboom et al. Emerging Convolutions for Generative Normalizing Flows. ICML, 2019.

Theoretical Guarantees

Characterization: for N=1, diagonal entries of convolution matrix (M) are $K_{n,n}$ of kernel (K) with size n and input is padded(top and left) with n-1.

M is invertible iff $K_{n,n} \neq 0$.

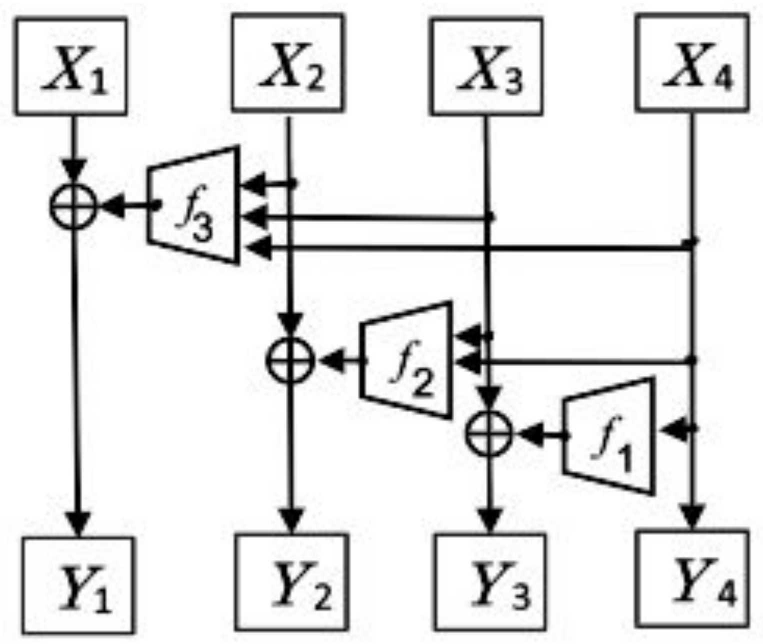
Assumption: N, # input channels = # output channels

Improved Coupling Schemes

Coupling Layers: we propose to use a modified version of the coupling layer designed to have a bigger receptive field. Inspired from generalized Feistal (Hong et al., 2010.)

Quad-coupling (proposed):

We divide the input into four blocks $x_1, x_2, x_3, x_4 = y_4$

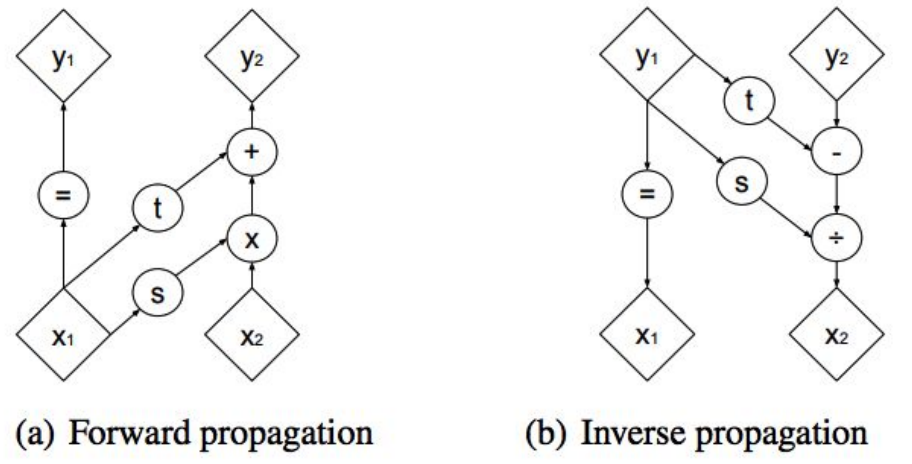


Why Quad-coupling ?

1. expressive coupling mechanism
2. Flexibility

Affine coupling: (Dinh et al., 2017)

Divide the input into two blocks x_1, x_2



(a) Forward propagation (b) Inverse propagation

-- output (y): concatenation of y_i

-- f_i and g_i are learned

-- component wise addition $\oplus$

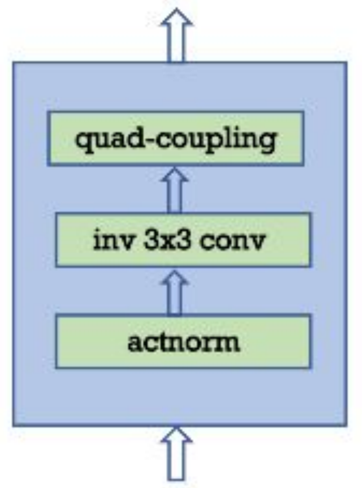
Model Architecture

L levels, and D flow modules per level.

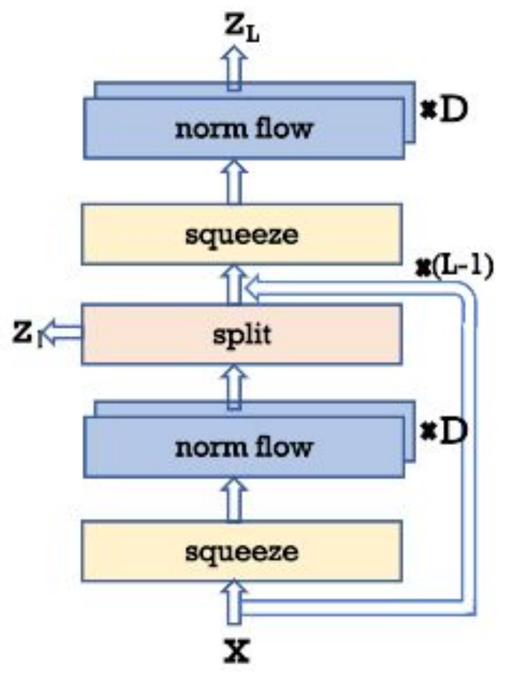
Flow module: Actnorm, Inv. 3x3 conv, Quad-coupling

Squeeze module: reorders pixels by reducing the spatial dimensions by a half, and increasing the channel depth by four.

x : input, z : output



Flow module



Benchmarks

Sampling time:(on one core CPU)

Time (sec.) needed to sample 100 images is almost two time faster.

Dataset	Sampling time (in sec) Emerging	CInC Flow
Cifar10	2.45	**1.31**
ImageNet32	4.96	**2.76**

Ablation for Quad Coupling

Bits per dimension for cifar10

Coupling	Emerging 3x3 Inv. conv.	Our 3x3 Inv. conv.
Affine	3.3851	3.4209
Quad	3.3612	3.3879

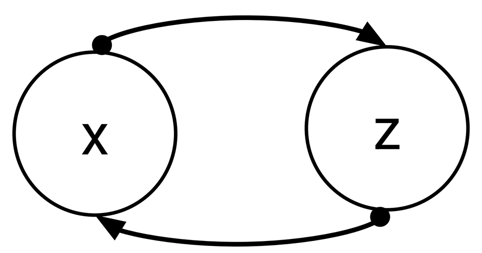
Bits per dimension comparison

Dataset	Glow	Emerging	CInC 3x3	+Quad
Cifar10	3.36	3.34	**3.3498**	3.347
ImageNet32	4.09	4.09	**4.0140**	4.0377
ImageNet64	3.81	3.81	3.8946	3.8514
Galaxy	----	2.2722	2.2739	**2.2591**

Qualitative Results

Demonstrated image manipulation capabilities on the CelebA dataset which has various attributes.

Latent space (Z): by changing the z we add or remove features.



Original Fake

Gradually modifying the age parameter:

Younger Original Older