



INTERNATIONAL INSTITUTE OF
INFORMATION TECHNOLOGY

HYDERABAD

the 4th
tractable probabilistic
modeling workshop

uai2021

école —
normale —
supérieure —
paris-saclay —

CInC Flow: Characterizable Invertible 3×3 Convolution



Sandeep Nagar¹, Marius Dufraisse², Girish Varma¹

¹Machine Learning Lab, IIIT Hyderabad, India.

²CS Dept., École Normale Supérieure (ENS), Paris-Saclay, France.

Characterizable Invertible Convolution (CInC)

Goal:

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

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,
4. a new coupling method.

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 $K_{n,n} \neq 0$.

	Filters	Padding	Receptive Field	Matrix	Observations $n = \# \text{in, out channels}$.
CInC (Ours)					#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.

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

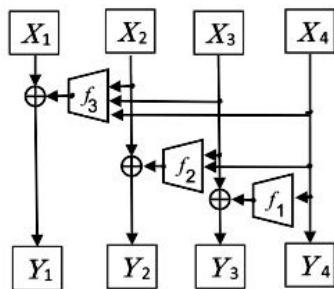
Assumption: N ,
 $\# \text{ input channels} = \# \text{ output channels}$

Characterizable Invertible Convolution (CInC)

Quad-coupling (proposed):

we use a modified version of the coupling layer designed to have a bigger receptive field. Inspired from generalized Feistel

(Hong et al., 2010.)



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

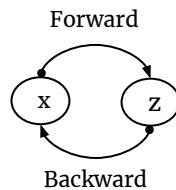
Why Quad-coupling ?

- Expressive coupling mechanism
- Flexibility

- output (y): concatenation of y_i
- f_i and g_i are learned
- component wise addition \oplus

Latent space (Z): by changing the z we add or remove features. Demonstrated image manipulation capabilities on the CelebA dataset which has various attributes.

Benchmark and Quantitative results:



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

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

