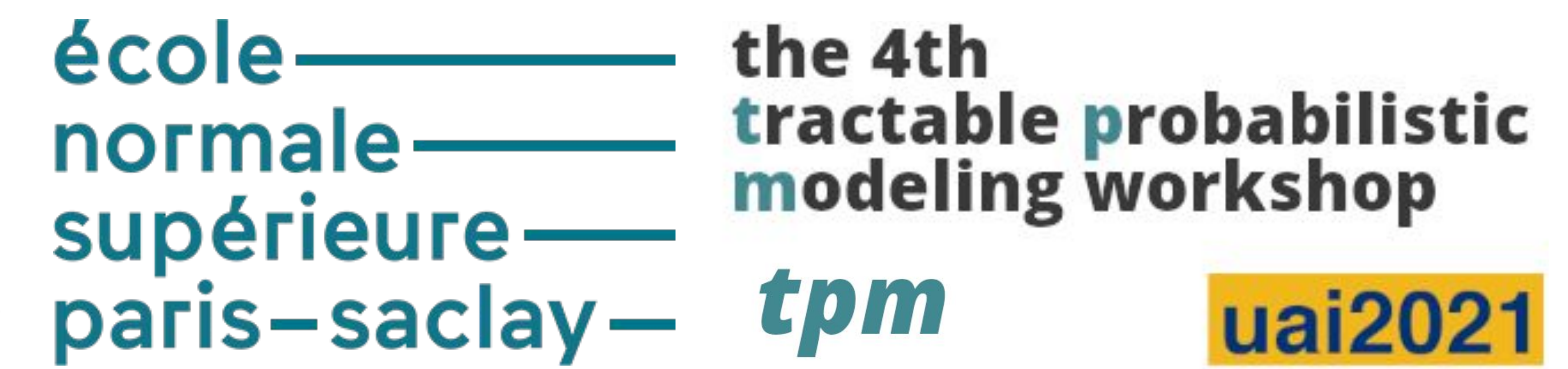


# CInC Flow: Characterizable Invertible 3x3 Convolution

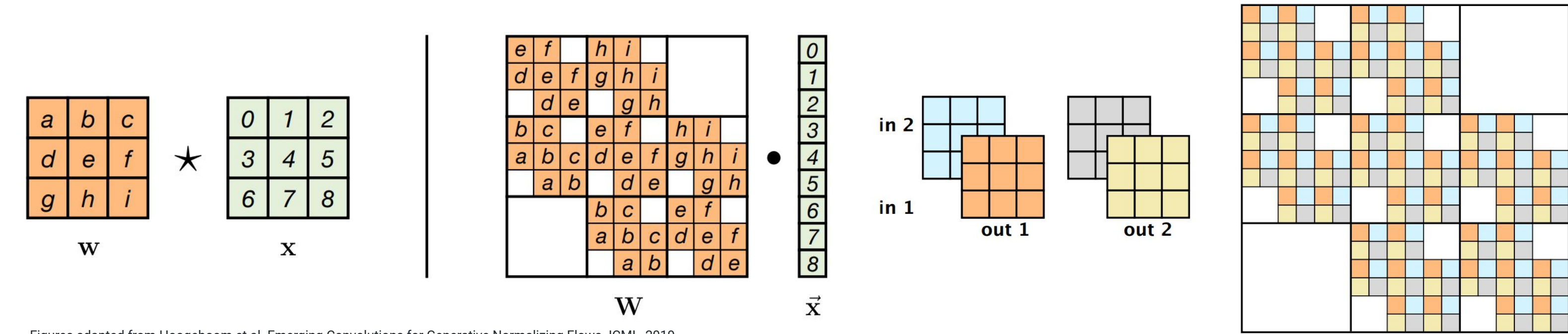
Sandeep Nagar<sup>1</sup>, Marius Dufraisie<sup>2</sup>, Girish Varma<sup>1</sup>

✉ sandeep.nagar@research.iiit.ac.in    [https://github.com/Naagar/Normalizing\\_Flow\\_3x3\\_inv](https://github.com/Naagar/Normalizing_Flow_3x3_inv)

<sup>1</sup>Machine Learning Lab, IIIT Hyderabad, India. <sup>2</sup>CS Dept., École Normale Supérieure (ENS), Paris-Saclay, France.



## Invertible Convolutions for Normalizing Flows



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..

## Characterizable 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.

## 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

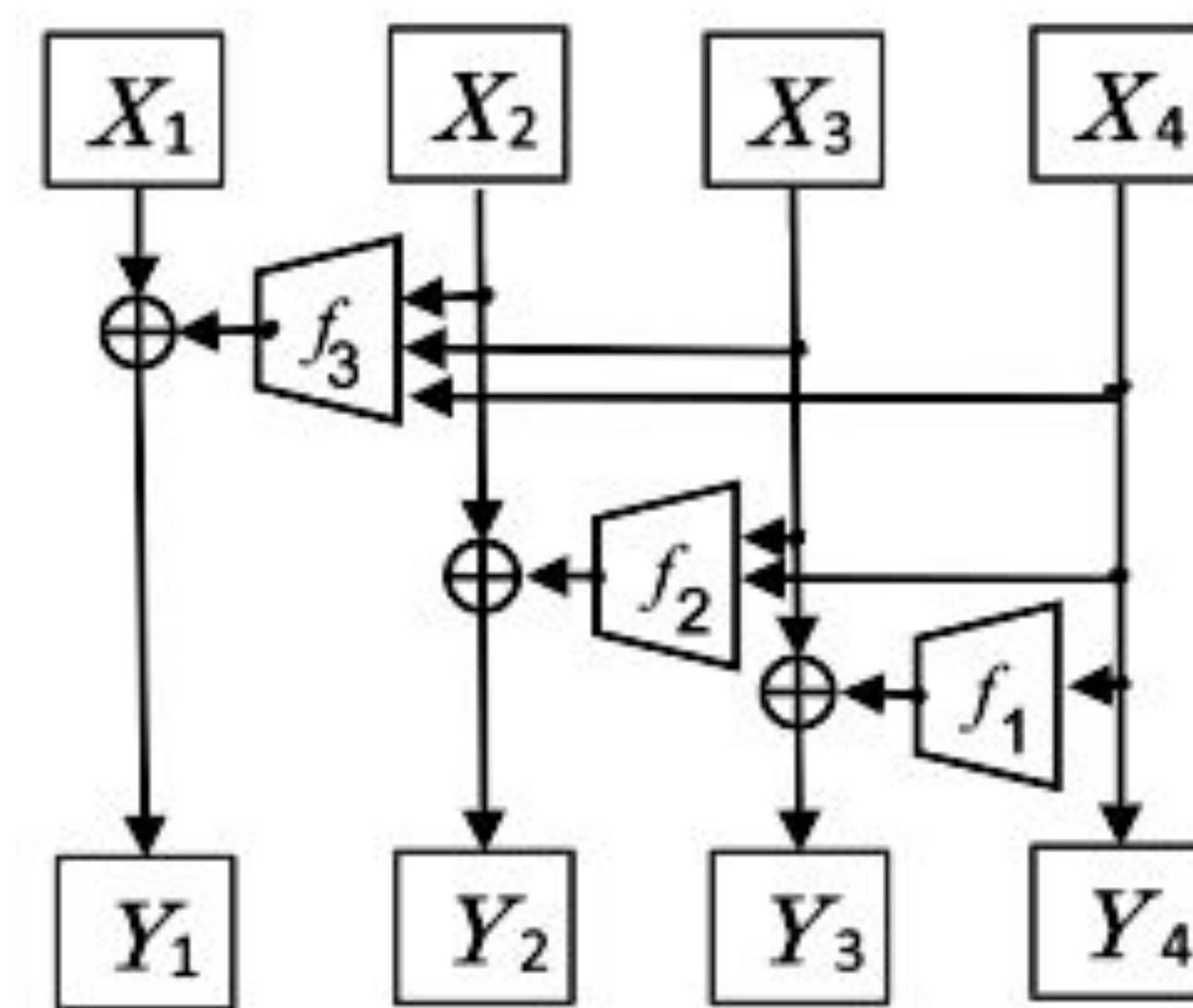
## Improved Coupling Schemes

**Coupling Layers:** we propose to use a modified version of the coupling layer designed to have a bigger receptive field. Inspired from generalized Feistel (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

## Model Architecture

$L$  levels, and  $D$  flow modules per level.

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

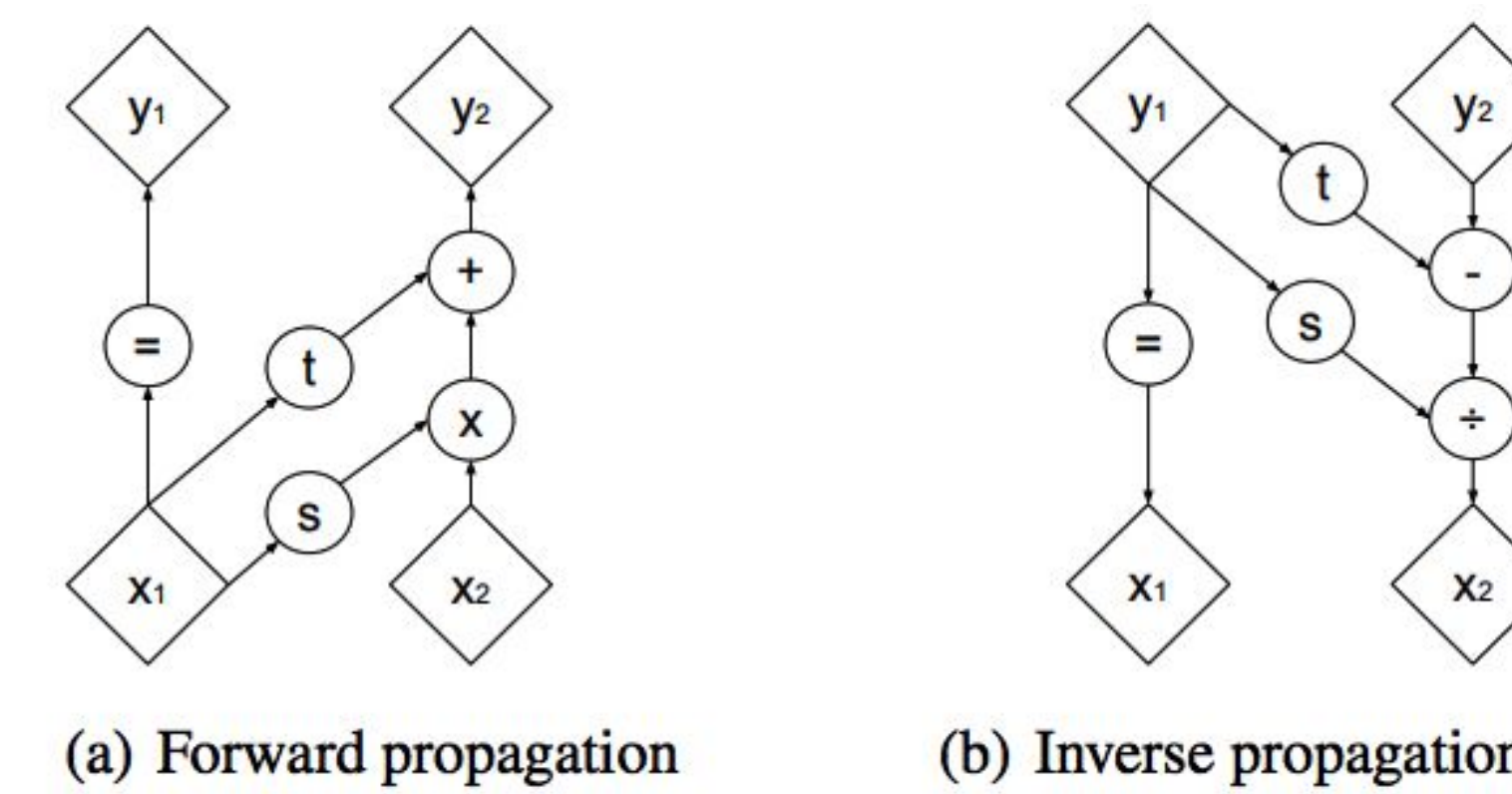
**Squeeze module:** reorders pixels by reducing the spatial dimensions by a half, and increasing the channel depth by four.

$$x : \text{input}, z : \text{output}$$

### Affine coupling:

(Dinh et al., 2017)

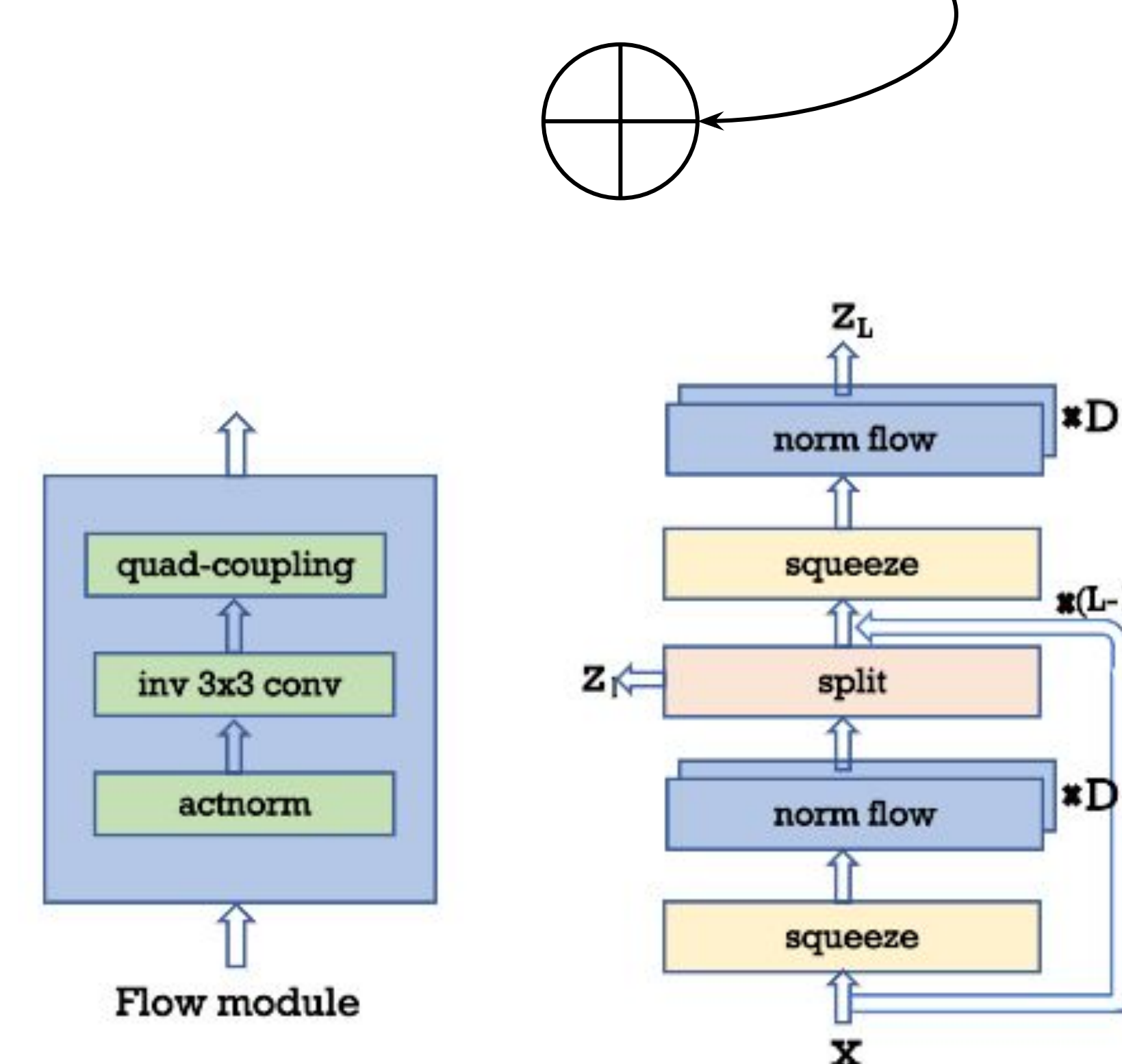
Divide the input into two blocks  $x_1, x_2$



-- output ( $y$ ): concatenation of  $y_1, y_2$

--  $f_1$  and  $g_1$  are learned

-- component wise addition



## 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	<b>1.31</b>
ImageNet32	4.96	<b>2.76</b>

### Ablation for Quad Coupling

Bits per dimension for cifar10

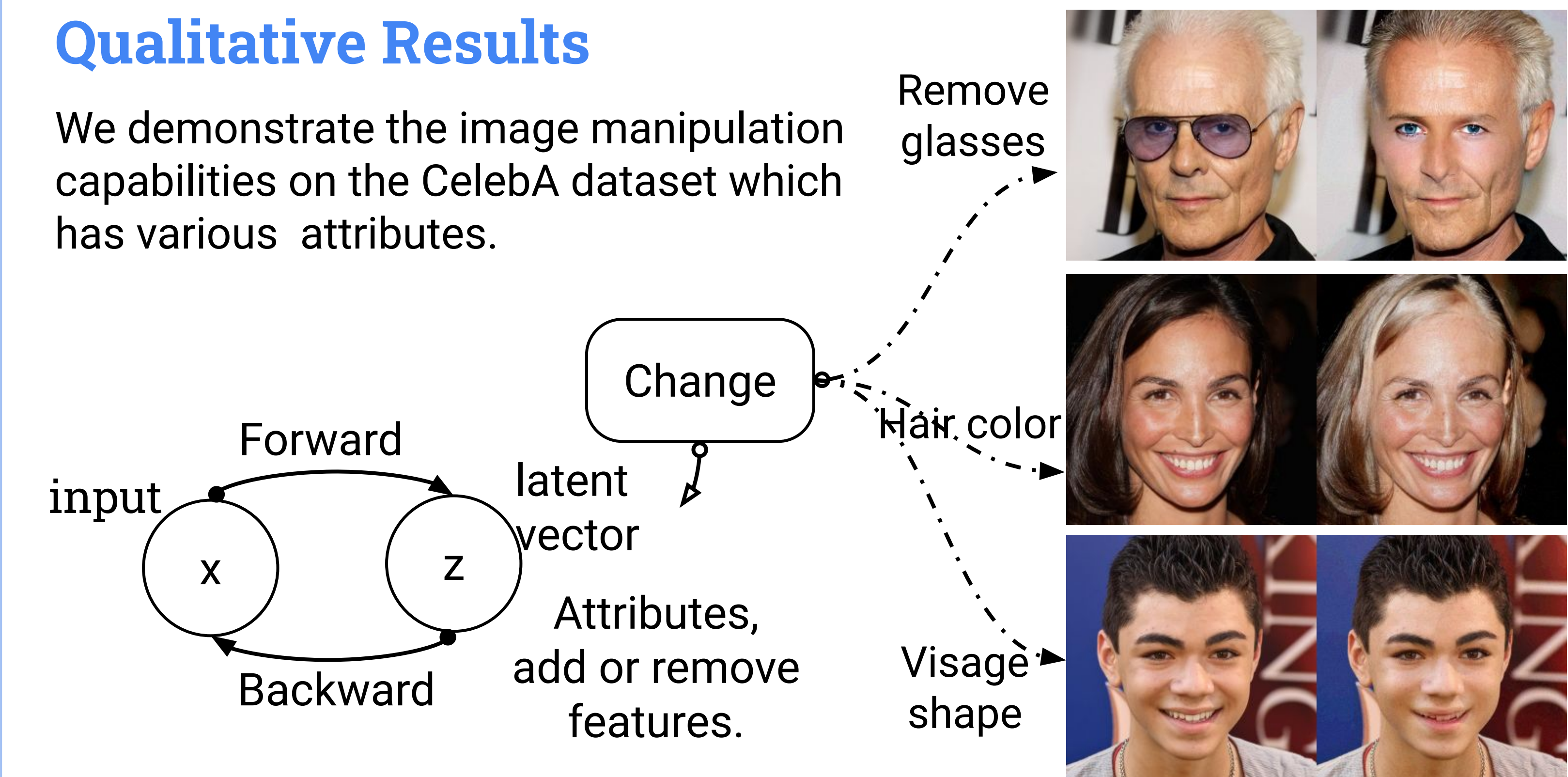
Coupling	Emerging 3x3	Our 3x3
	Inv. conv.	Inv. conv.
Affine	3.3851	3.4209
Quad	3.3612	3.3879

### Bits per dimension comparison

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

## Qualitative Results

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



### Gradually modifying the age parameter:

