Normalizing Flows are Capable Generative Models (TarFlow) [ICML 2025]

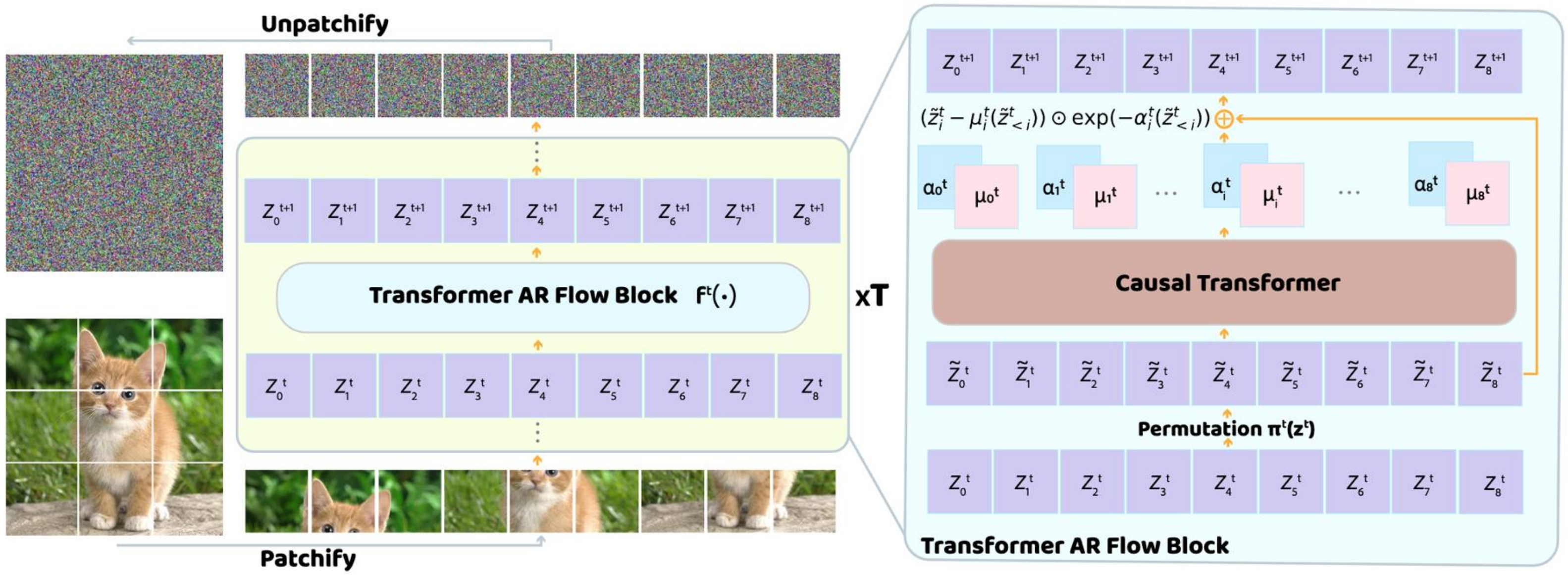
Presented by Tony Qin

Shuangfei Zhai[1], Ruixiang Zhang[1], Preetum Nakkiran[1], David Berthelot[1], Jiatao Gu[1], Huangjie Zheng[1] , Tianrong Chen[1], Miguel Angel Bautista[1], Navdeep Jaitly[1], Josh Susskind[1]

[1]Apple

Table of Contents

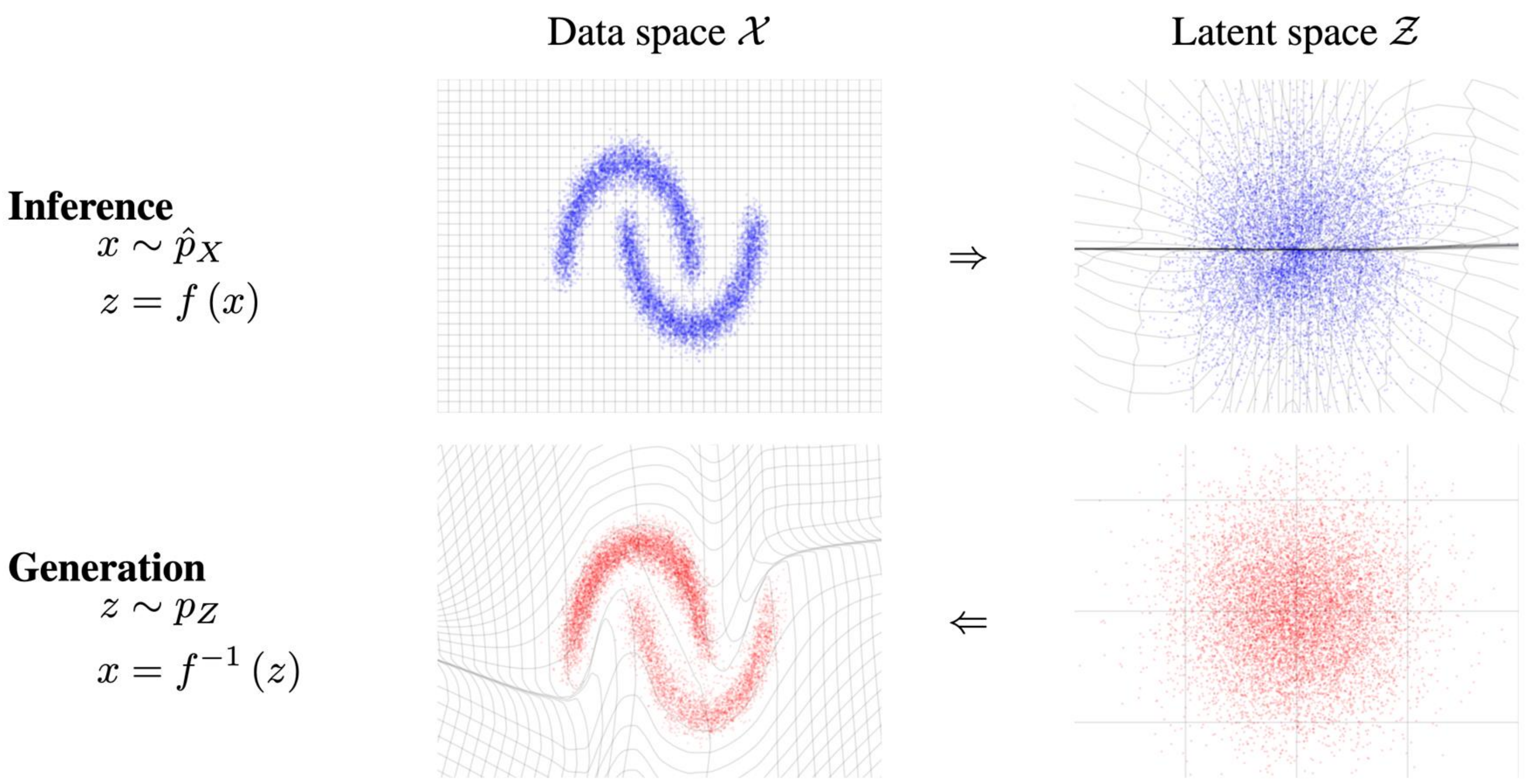
Table of Contents



"**Architecture design** is arguably the most challenging aspect of NF models. We suspect that a large part of the reason that NFs have not been as performant as other families of models is the lack of an architecture that allows for stable and scalable training."

Table of Contents

Normalizing Flows



L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using Real NVP," Feb. 27, 2017, *arXiv*: arXiv:1605.08803. doi: [10.48550/arXiv.1605.08803](https://doi.org/10.48550/arXiv.1605.08803).

Normalizing Flows

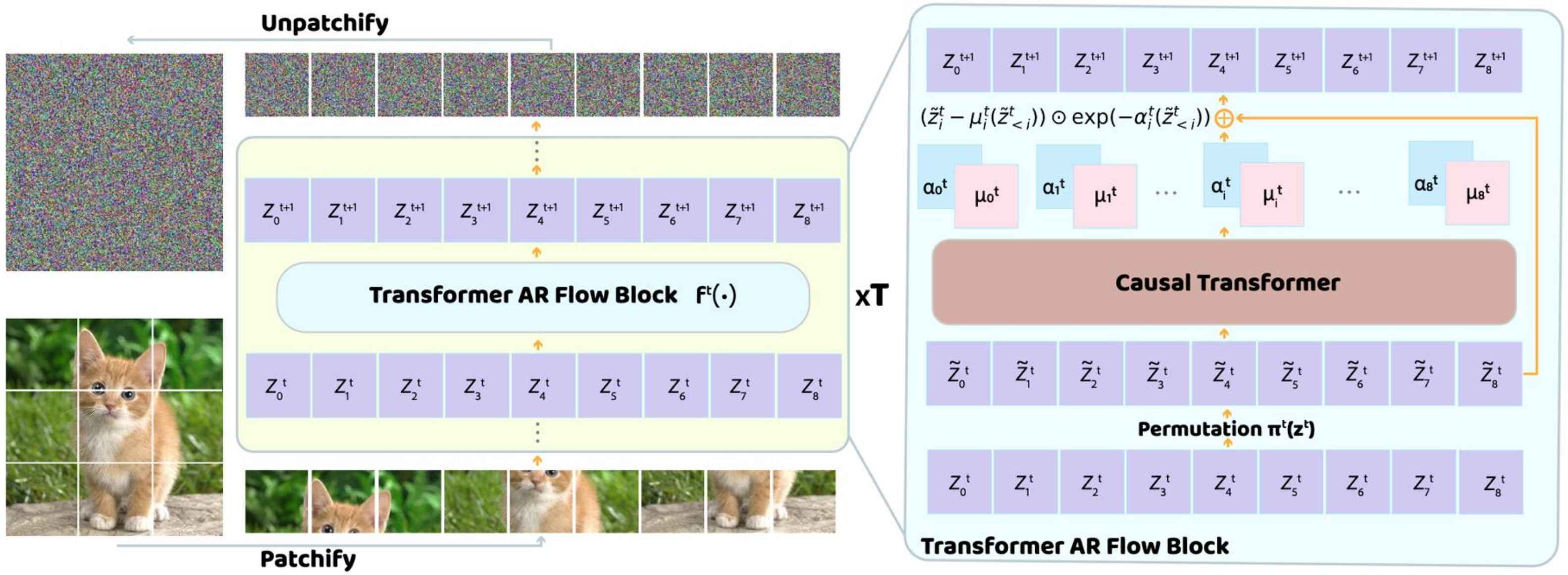
We wish to learn a transformation $f(x) = z$ that

1) Is invertible

2) Has a tractable determinant of the Jacobian (triangular Jacobian)

$$p_{model}(x) = p_0\big(f(x)\big) \left| \det \frac{df(x)}{dx} \right|$$

Forward Pass



$$z = f(x)$$

Forward Pass

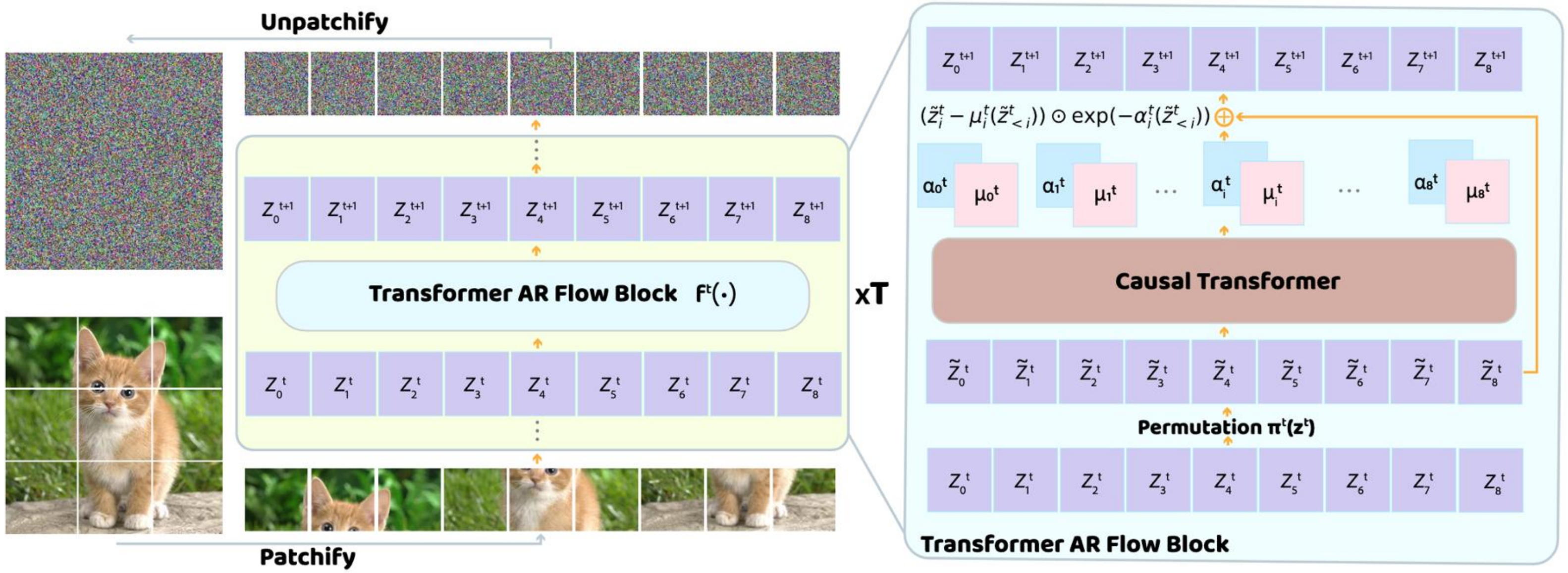
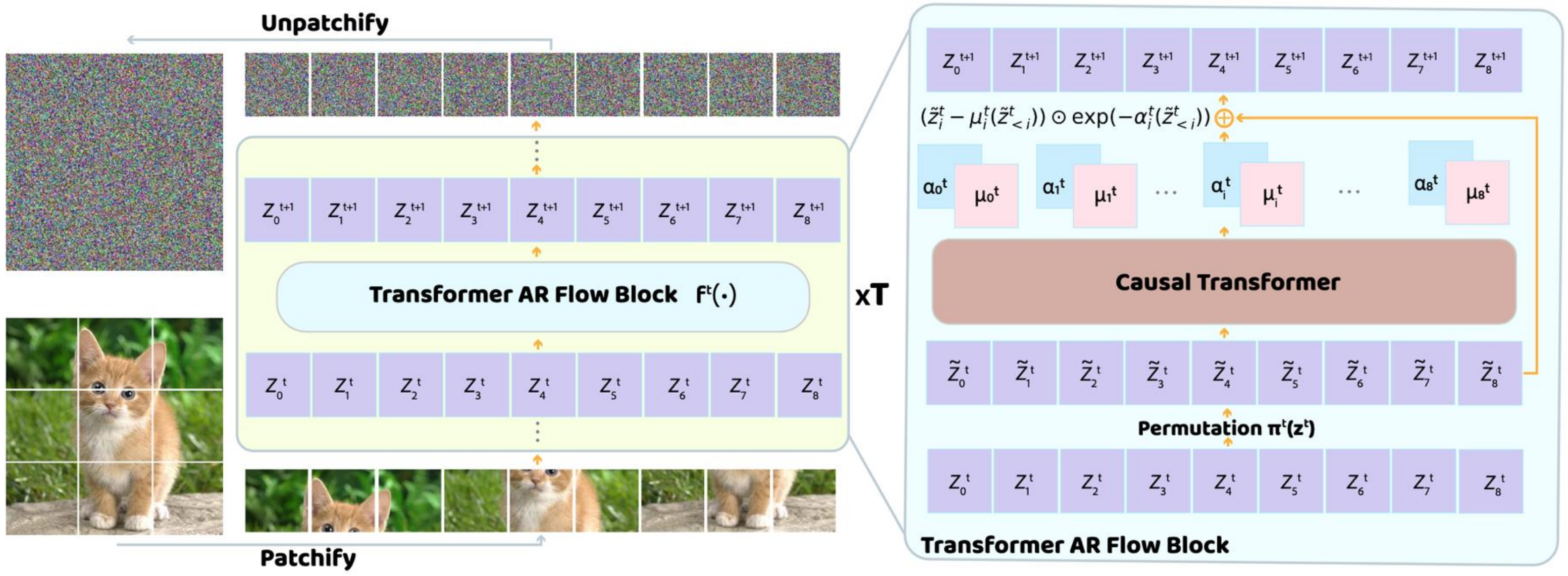
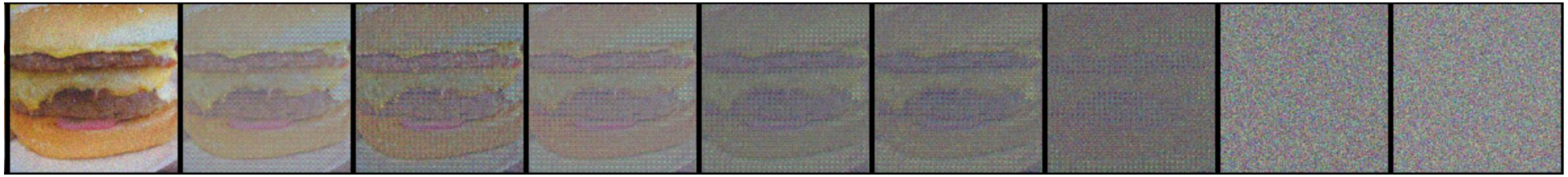


Image: $Z^0 \leftarrow x$

Latent: $Z^t = f^{t-1}(f^{t-2}(f^{\cdots}(f^1(f^0(Z^0)) \ldots)$

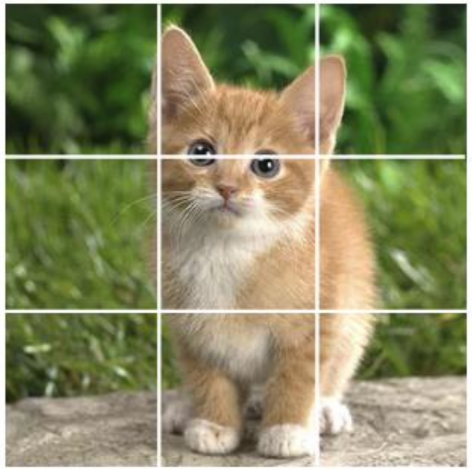
Forward Pass





Z^0 Z^1 Z^2 Z^3 Z^4 Z^5 Z^6 Z^7 Z^8

Forward Pass: Patchify



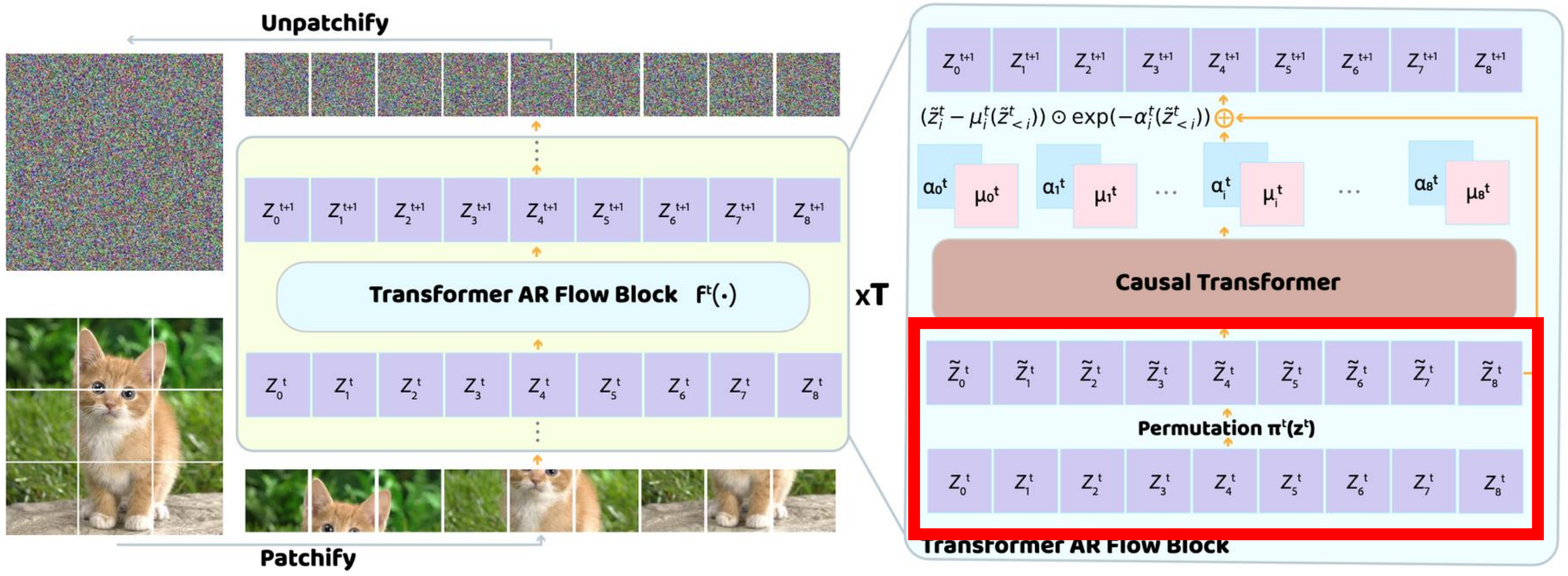
Z_0^0 Z_1^0 Z_2^0 Z_3^0 Z_4^0 Z_5^0 Z_6^0 Z_7^0 Z_8^0

$1 \times D$ $1 \times D$ $1 \times D$ $1 \times D$ $1 \times D$ $1 \times D$ $1 \times D$ $1 \times D$ $1 \times D$

$$\underset{\mathrm{N} \times D}{Z^0} = \boxed{Z_0^0 \,|\, Z_1^0 \,|\, Z_2^0 \,|\, Z_3^0 \,|\, Z_4^0 \,|\, Z_5^0 \,|\, Z_6^0 \,|\, Z_7^0 \,|\, Z_8^0}$$

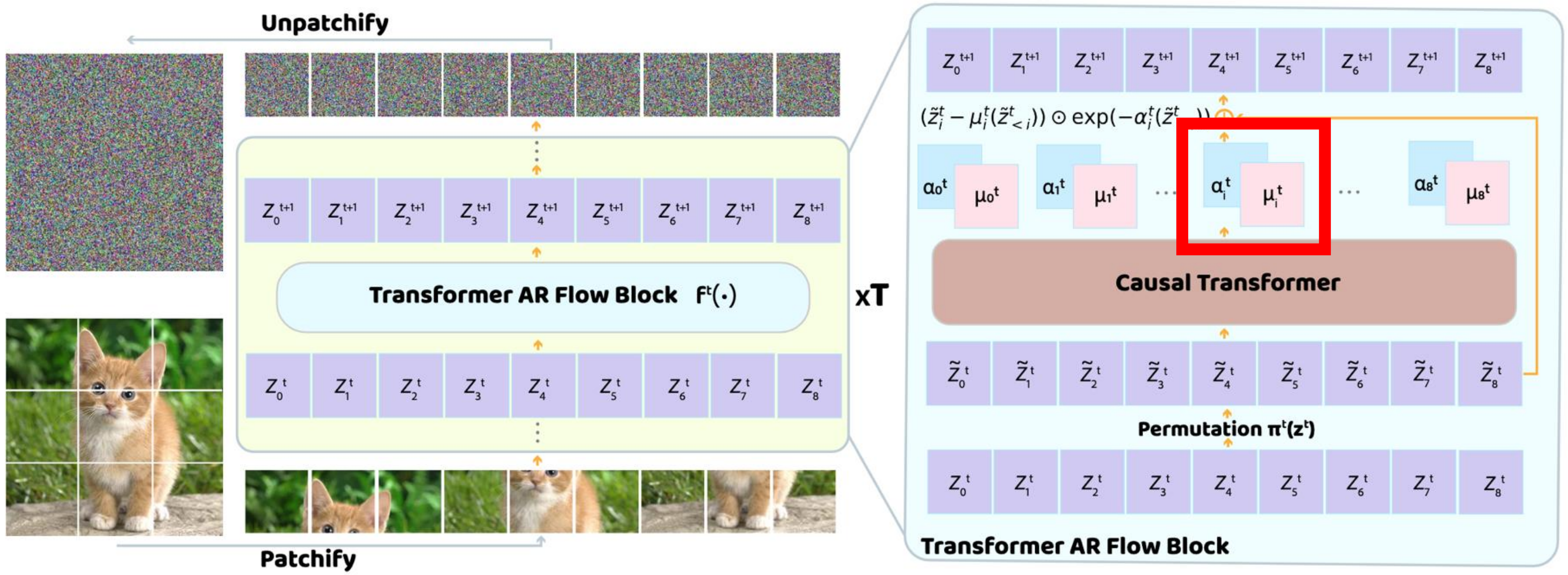
We first patchify the image, just like for a Vision Transformer.

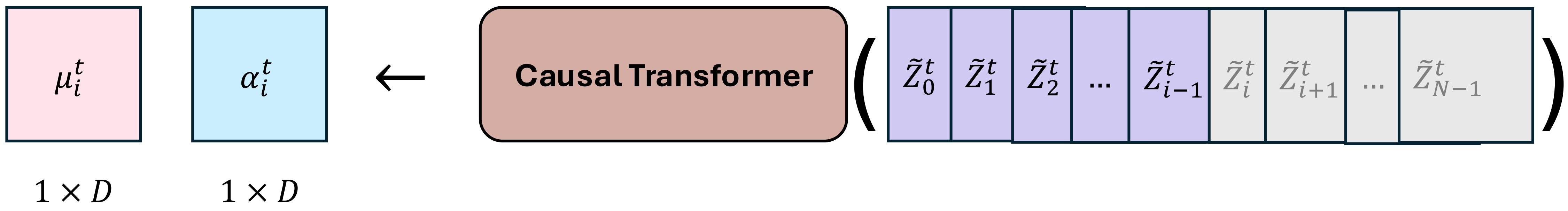
Forward Pass



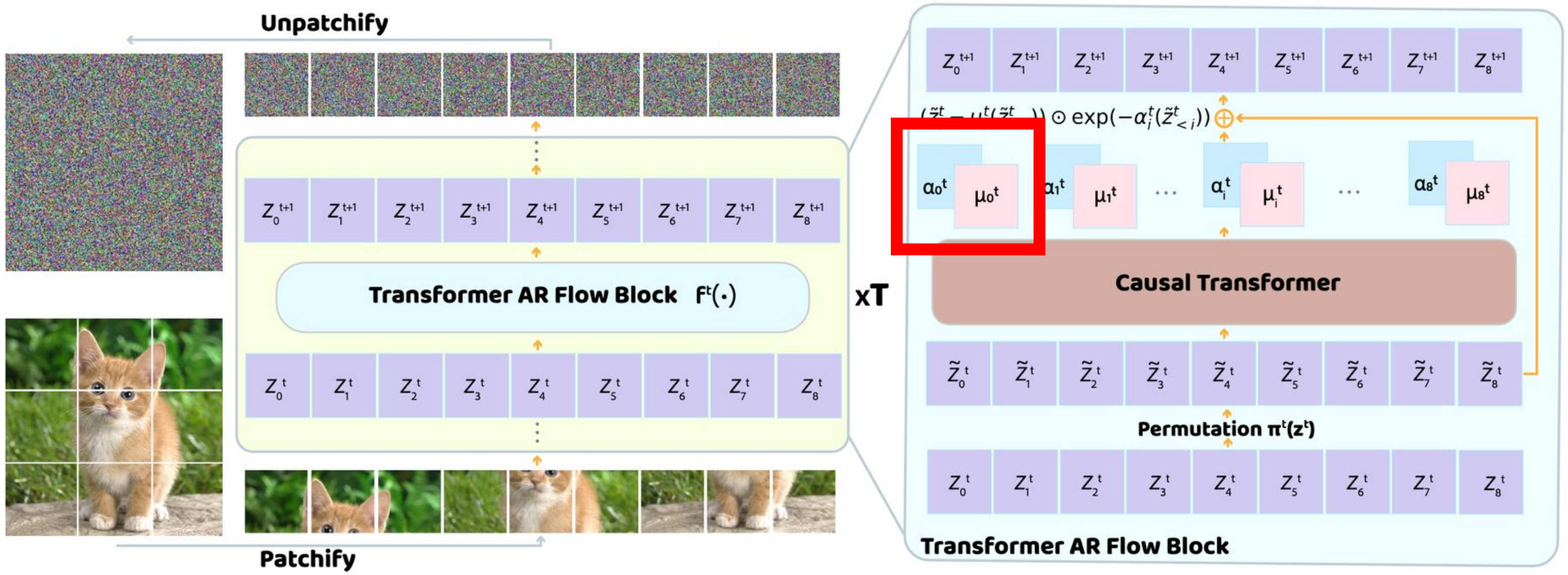
π^0: identity
$\pi^{t>0}$: reverse

Forward Pass



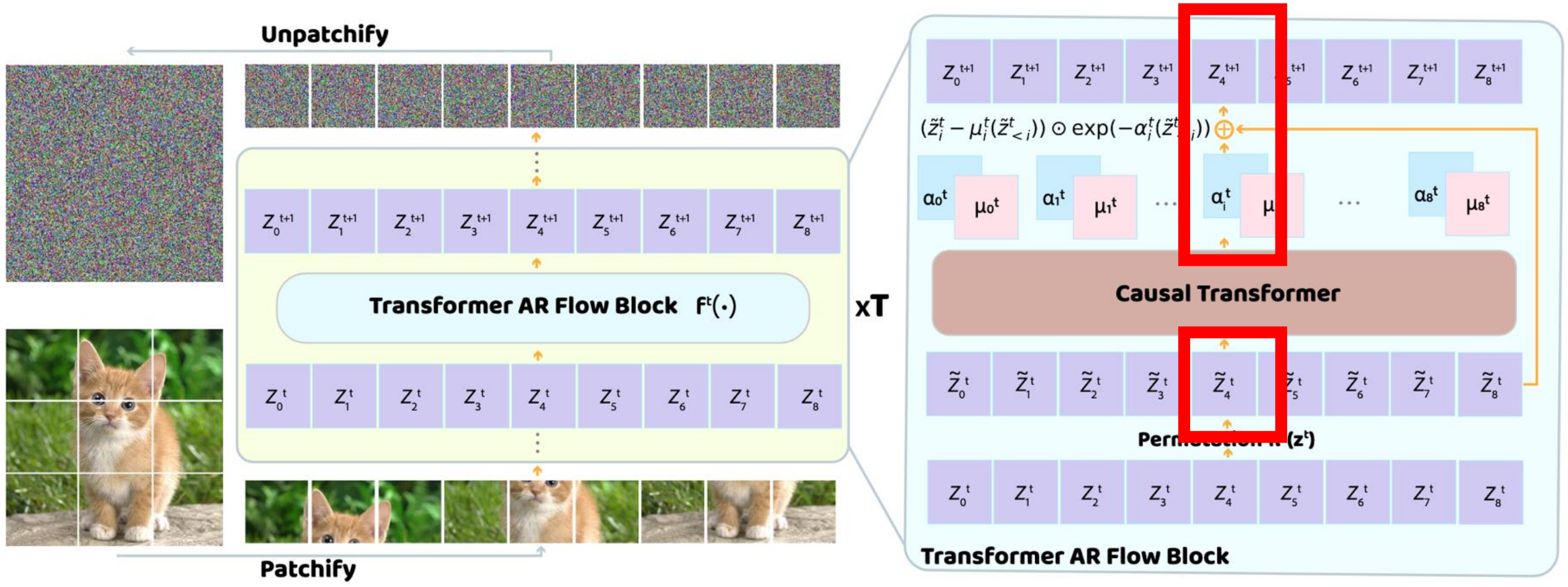


Forward Pass



$$\begin{array}{cccc} \mu_0^t & \alpha_0^t & \leftarrow & \mathbf{0} & \mathbf{0} \\ 1 \times D & 1 \times D & & 1 \times D & 1 \times D \end{array}$$

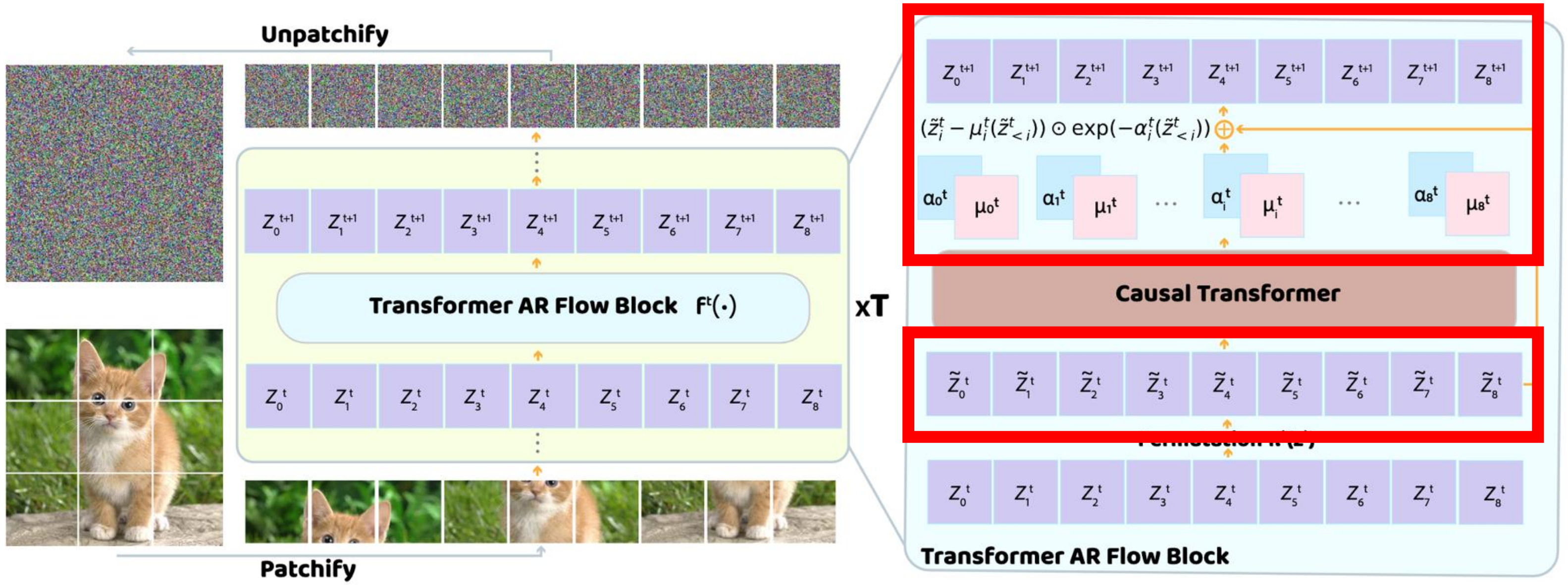
Forward Pass



$$Z_i^{t+1} \leftarrow \left(\tilde{Z}_i^t - \mu_i^t \right) \odot \exp\left(-\alpha_i^t \right)$$

$1 \times D$ $1 \times D$ $1 \times D$ $1 \times D$

Forward Pass



$$Z^{t+1} \leftarrow \left(\tilde{Z}^t - \mu^t \right) \odot \exp\left(-\alpha^t \right)$$

$N \times D$ $N \times D$ $N \times D$ $N \times D$

Forward Pass: Training

$$p_{model}(x) = p_0\big(f(x)\big)\left|\det\frac{df(x)}{dx}\right|$$

NLL: $\min_f \; -\log p_0\big(f(x)\big) - \log\left(\left|\det\left(\frac{df(x)}{dx}\right)\right|\right)$

$$= \min_f \; -0.5\|f(x)\|_2^2 - \log\left(\left|\det\left(\frac{df(x)}{dx}\right)\right|\right)$$

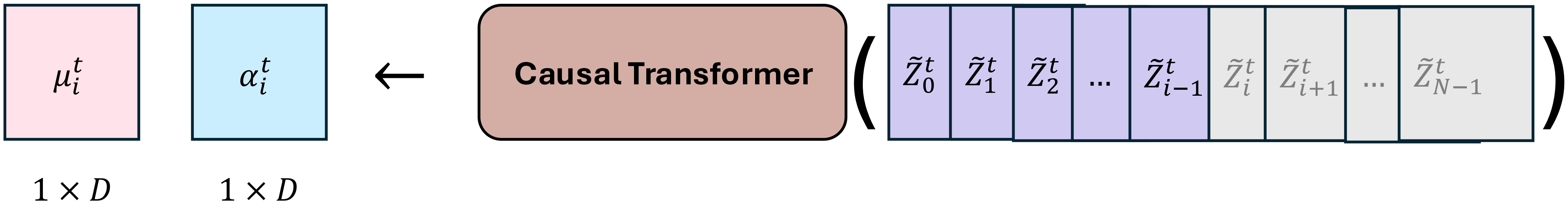
Where p_0 is the standard Gaussian

Forward Pass: Jacobian

$$z_i^{t+1} = (\tilde{z}_i^t - \mu_i^t(\tilde{z}_{<i}^t)) \odot \exp\left(-\alpha_i^t(\tilde{z}_{<i}^t)\right)$$

$$\Rightarrow \frac{dz_i^{t+1}}{d\tilde{z}_i^t} = \exp\left(-\alpha_i^t(\tilde{z}_{<i}^t)\right) \quad ; \quad \frac{dz_i^{t+1}}{d\tilde{z}_{>i}^t} = 0$$

The Jacobian is triangular!



Forward Pass: Jacobian

$$z_i^{t+1} = (\tilde{z}_i^t - \mu_i^t(\tilde{z}_{<i}^t)) \odot \exp\left(-\alpha_i^t\left(\tilde{z}_{<i}^t\right)\right)$$

$$\Rightarrow \frac{dz_i^{t+1}}{d\tilde{z}_i^t} = \exp\left(-\alpha_i^t\left(\tilde{z}_{<i}^t\right)\right) \quad ; \quad \frac{dz_i^{t+1}}{d\tilde{z}_{>i}^t} = 0$$

The Jacobian is triangular!

$$\Rightarrow \log\left(\left|\det\left(\frac{df^t(z^t)}{dz^t}\right)\right|\right) = -\sum_{i=1}^{N-1}\sum_{j=0}^{D-1} \alpha_i^t\left(\tilde{z}_{<i}^t\right)_j$$

$$-\sum\left(\boxed{\alpha^t}\right)$$

$$\Rightarrow \log\left(\left|\det\left(\frac{df(x)}{dx}\right)\right|\right) = -\sum_{t=0}^{T-1}\sum_{i=1}^{N-1}\sum_{j=0}^{D-1} \alpha_i^t\left(\tilde{z}_{<i}^t\right)_j$$

$$-\sum\left(\begin{array}{|c|}\hline \alpha^{T-1} \\ \hline \alpha^{\cdots} \\ \hline \alpha^2 \\ \hline \alpha^1 \\ \hline \end{array}\right)$$

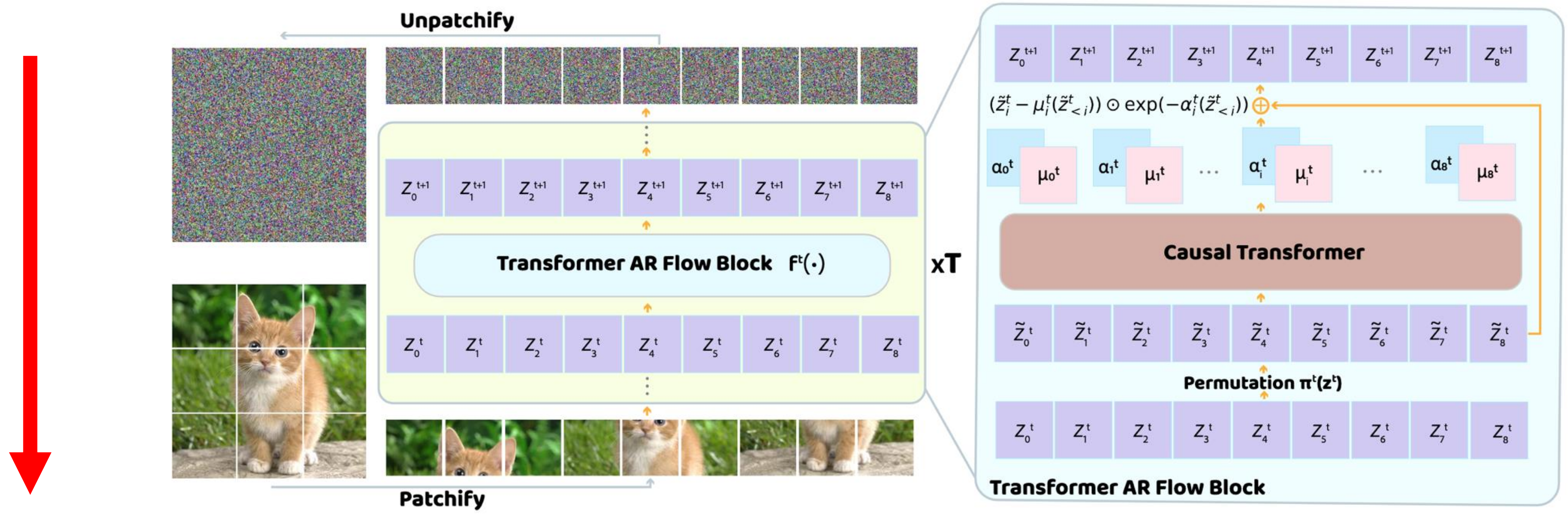
Forward Pass: Training

$$p_{model}(x) = p_0\big(f(x)\big)\left| \det \frac{df(x)}{dx} \right|$$

NLL: $\min_f \; -0.5\|f(x)\|_2^2 - \log\left(\left|\det\left(\frac{df(x)}{dx}\right)\right|\right)$

$$= \min_f -0.5\|z^T\|_2^2 + \sum_{t=0}^{T-1}\sum_{i=1}^{N-1}\sum_{j=0}^{D-1} \alpha_i^t\left(\tilde{z}_{<i}^t\right)_j$$

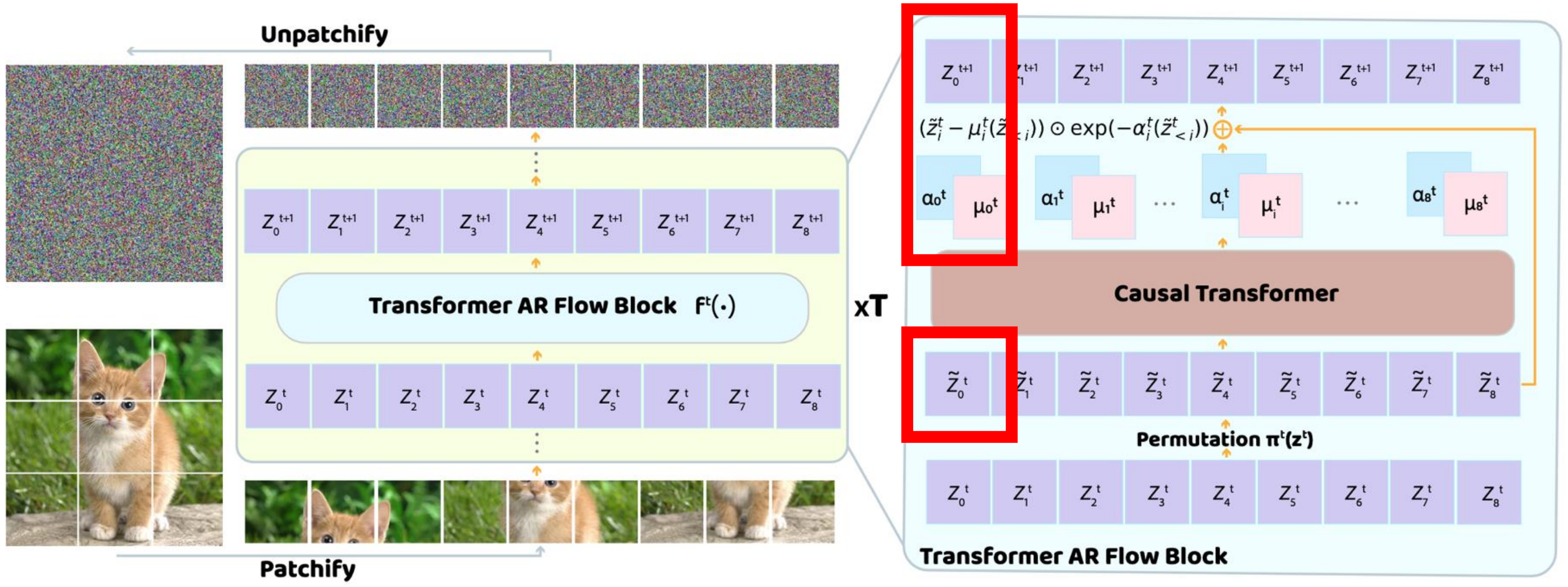
Backward Pass (Generation)



$$z = f(x)$$

$$\Rightarrow x = f^{-1}(z)$$

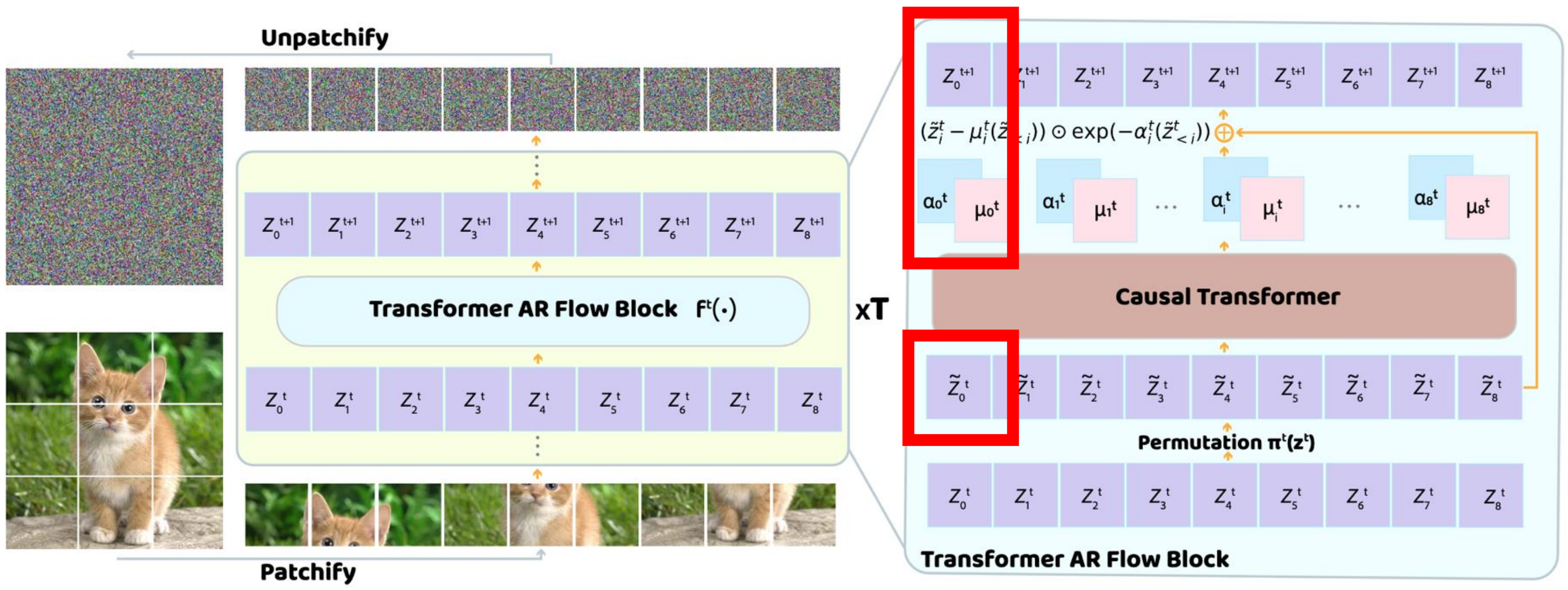
Backward Pass (Generation)



$$\boxed{Z_0^{t+1}} = \left(\boxed{\tilde{Z}_0^t} - \boxed{\mathbf{0}} \right) \odot \exp\left(\boxed{\mathbf{0}} \right)$$

$1 \times D$ $\qquad$ $1 \times D$ $\qquad$ $1 \times D$ $\qquad$ $1 \times D$

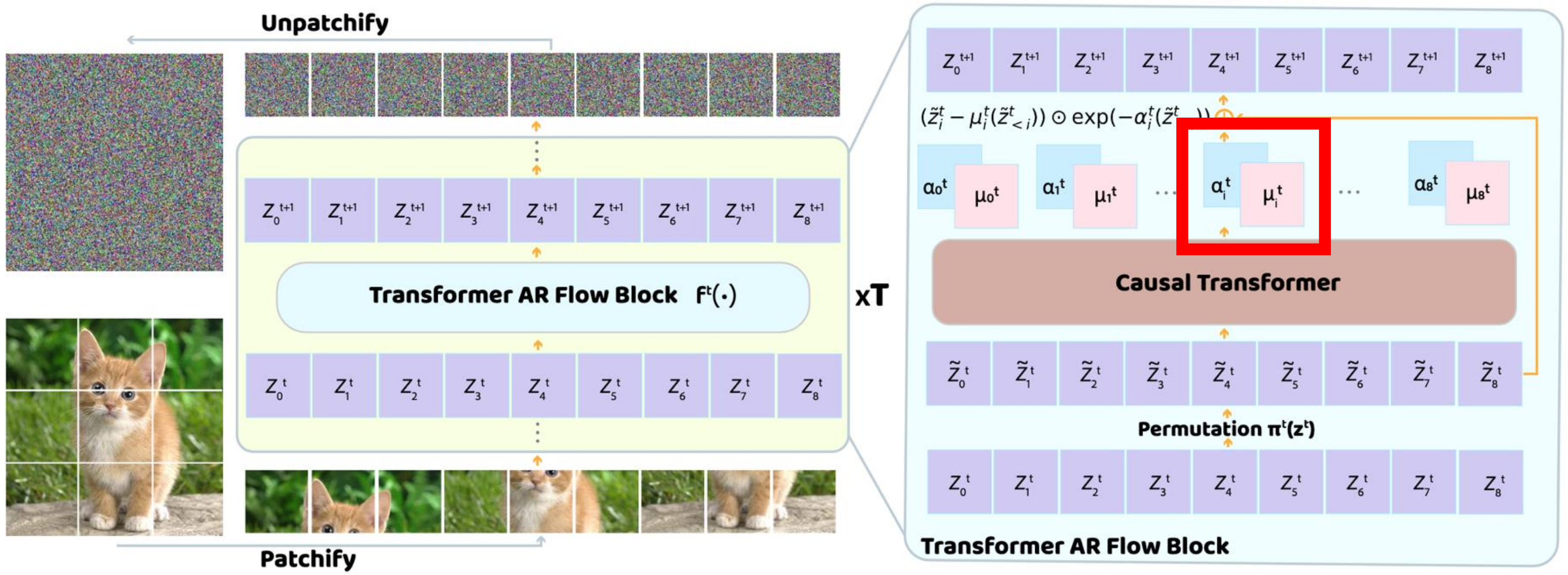
Backward Pass (Generation)



$$\tilde{Z}_0^t \leftarrow Z_0^{t+1}$$

$1 \times D$ $\qquad$ $1 \times D$

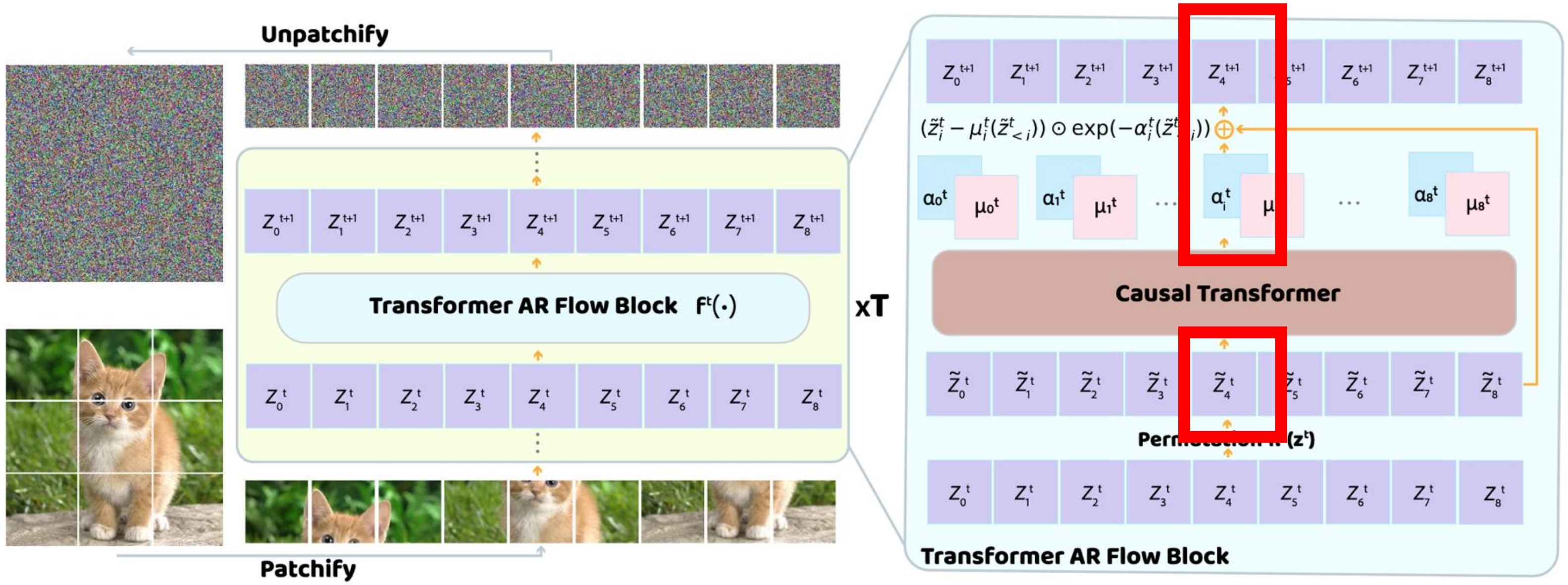
Backward Pass (Generation)



μ_i^t α_i^t $\leftarrow$ **Causal Transformer** $\left(\tilde{Z}_0^t \; \tilde{Z}_1^t \; \tilde{Z}_2^t \; \ldots \; \tilde{Z}_{i-1}^t \; \tilde{Z}_i^t \; \tilde{Z}_{i+1}^t \; \ldots \; \tilde{Z}_{N-1}^t \right)$

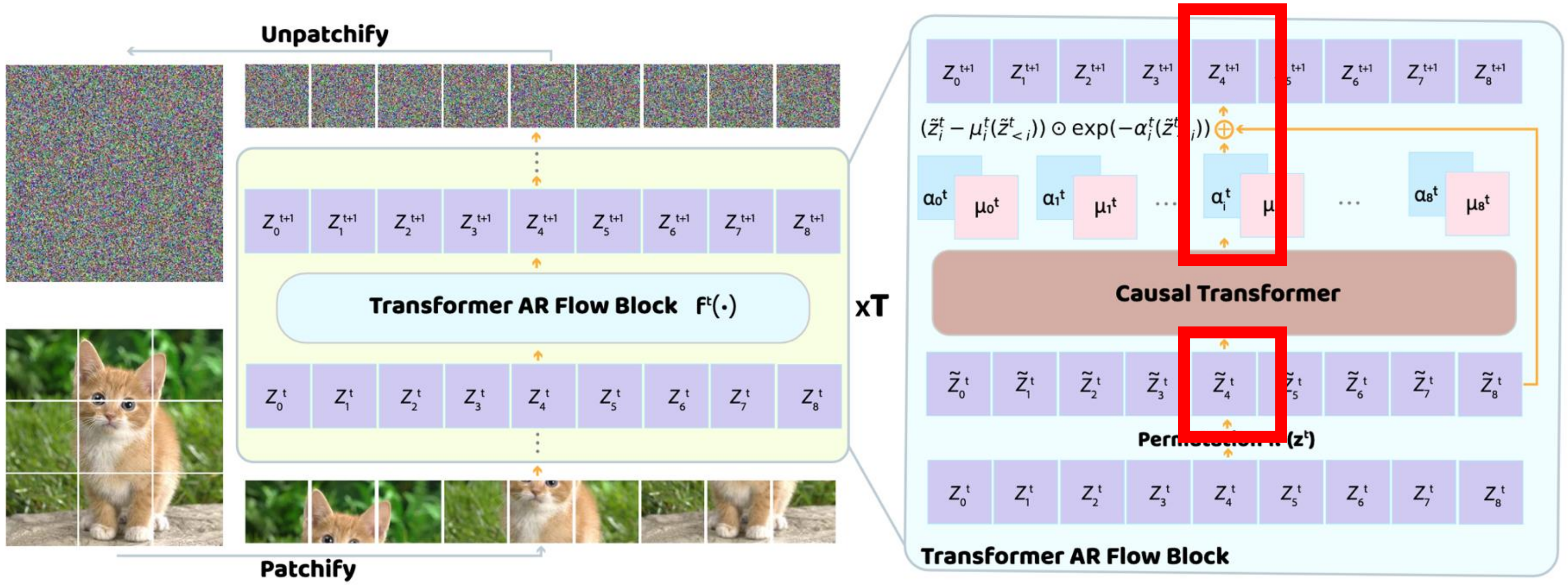
$1 \times D$ $1 \times D$

Backward Pass (Generation)



$$\underset{1 \times D}{z_i^{t+1}} = \Big(\underset{1 \times D}{\tilde{z}_i^t} - \underset{1 \times D}{\mu_i^t} \Big) \odot \exp\Big(\underset{1 \times D}{-\alpha_i^t} \Big)$$

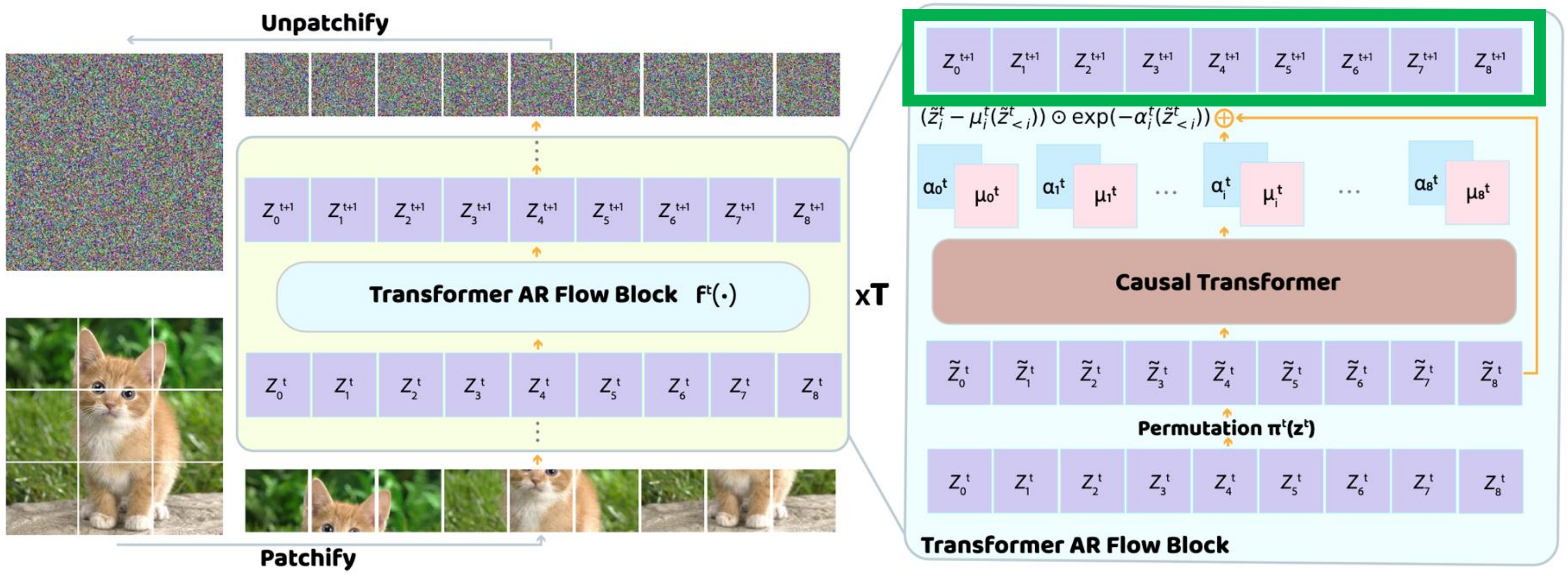
Backward Pass (Generation)



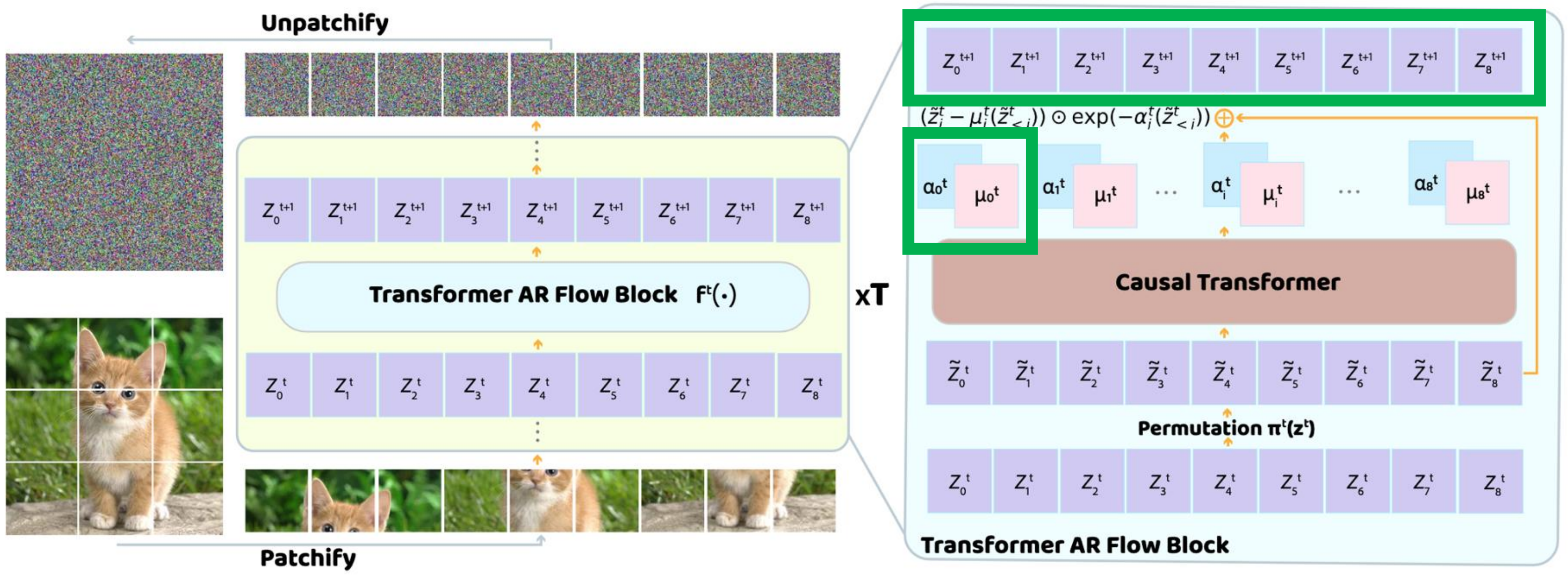
$$\tilde{Z}_i^t \leftarrow Z_i^{t+1} \odot \exp\left(\alpha_i^t\right) + \mu_i^t$$

$1 \times D$ $\quad 1 \times D$ $\quad 1 \times D$ $\quad 1 \times D$

Backward Pass (Generation)

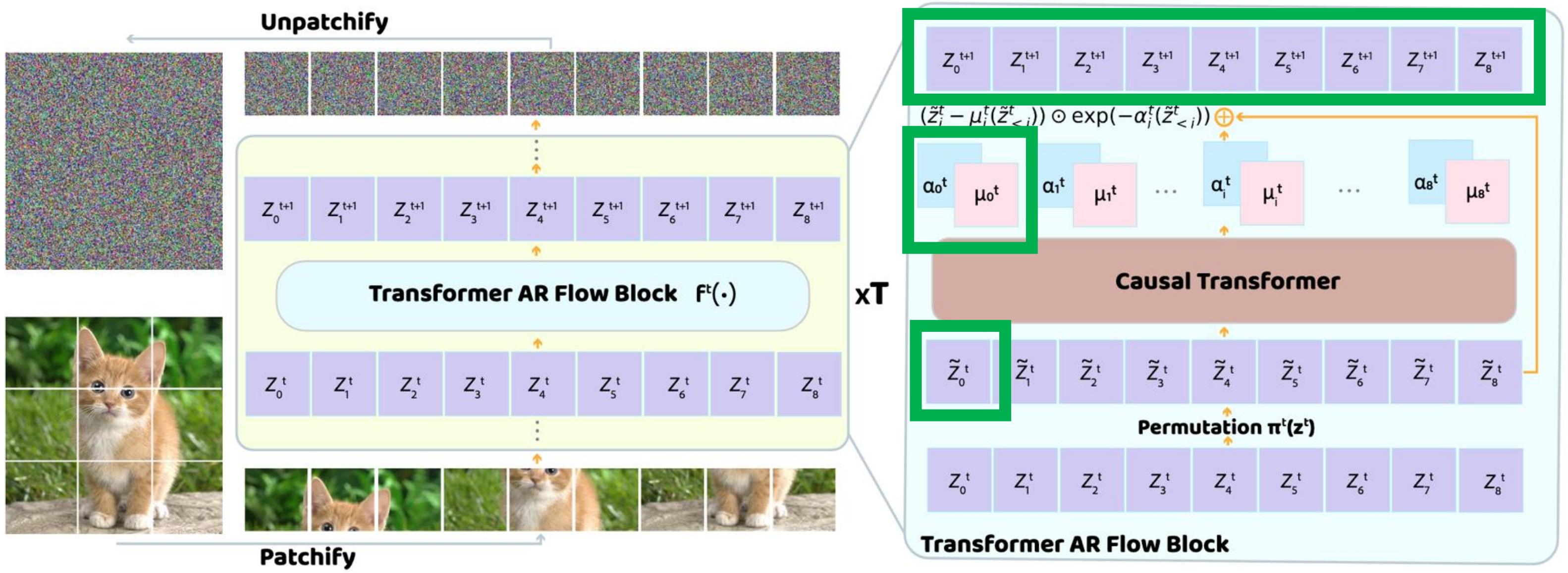


Backward Pass (Generation)



$$\underset{1 \times D}{\boxed{\mu_0^t}} \quad \underset{1 \times D}{\boxed{\alpha_0^t}} \quad \leftarrow \quad \underset{1 \times D}{\boxed{\mathbf{0}}} \quad \underset{1 \times D}{\boxed{\mathbf{0}}}$$

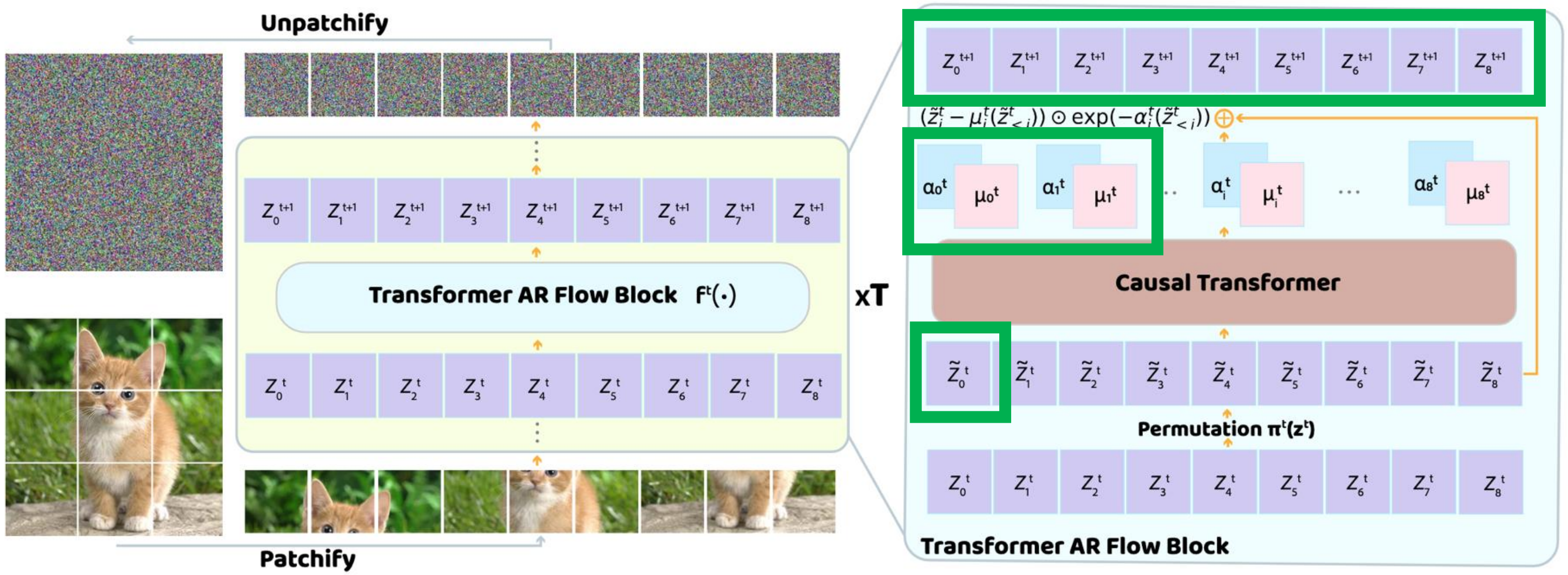
Backward Pass (Generation)

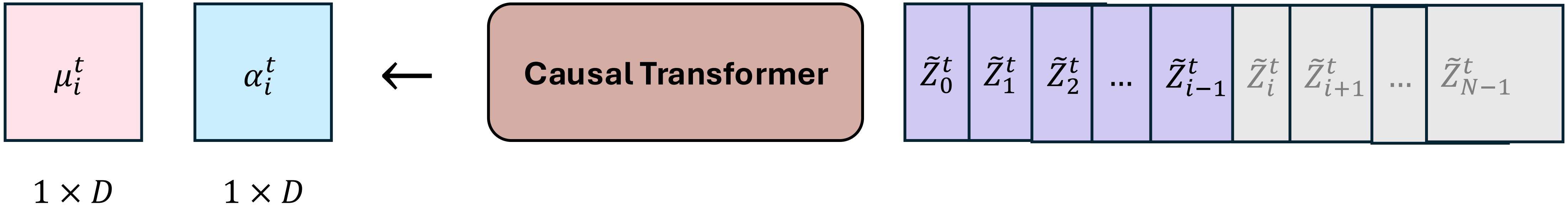


$$\tilde{Z}_0^t \leftarrow Z_0^{t+1}$$

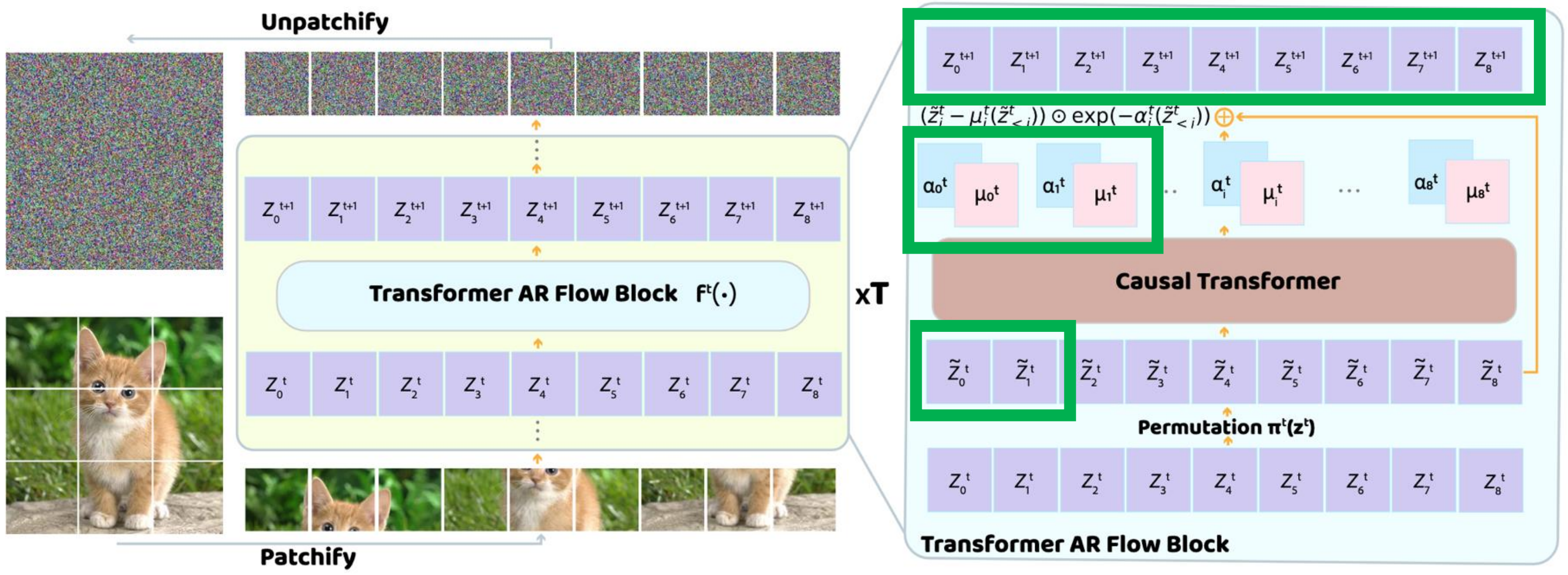
$1 \times D$ $\quad$ $1 \times D$

Backward Pass (Generation)





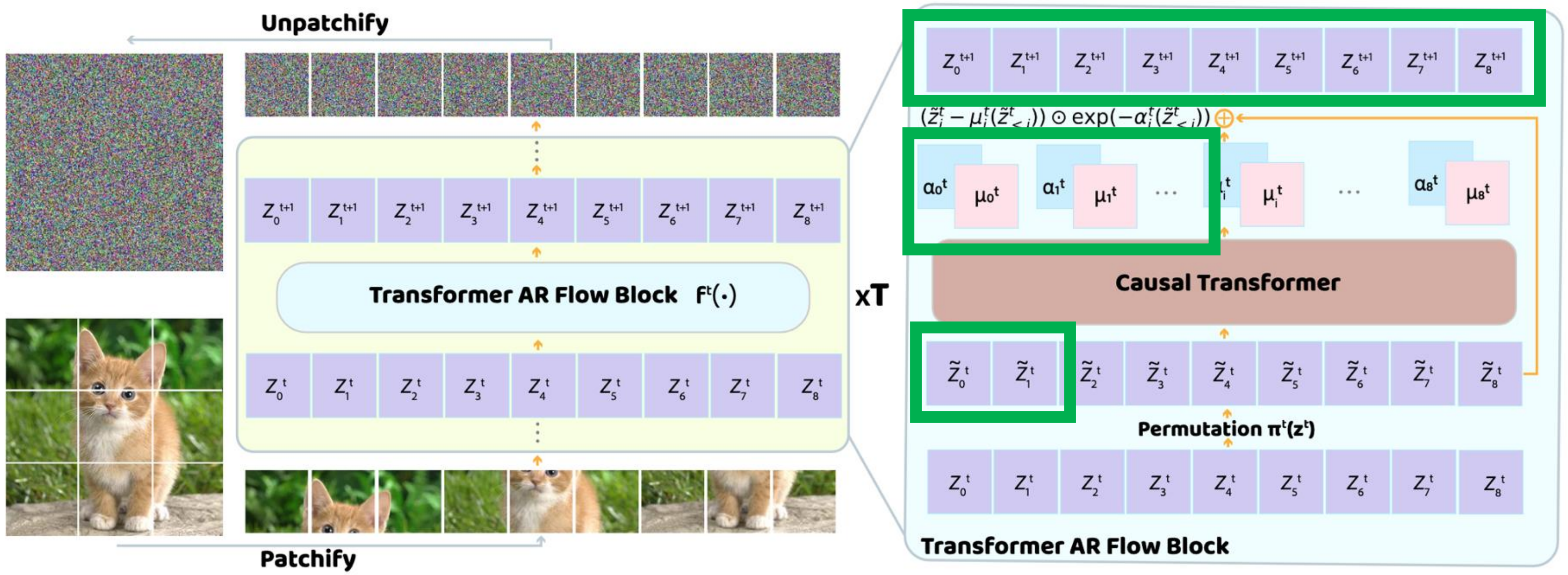
Backward Pass (Generation)

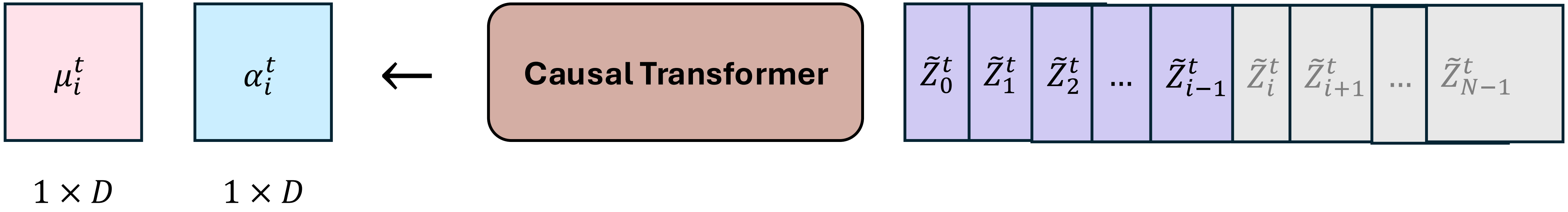


$$\tilde{z}_i^t \leftarrow z_i^{t+1} \odot \exp\left(\alpha_i^t\right) + \mu_i^t$$

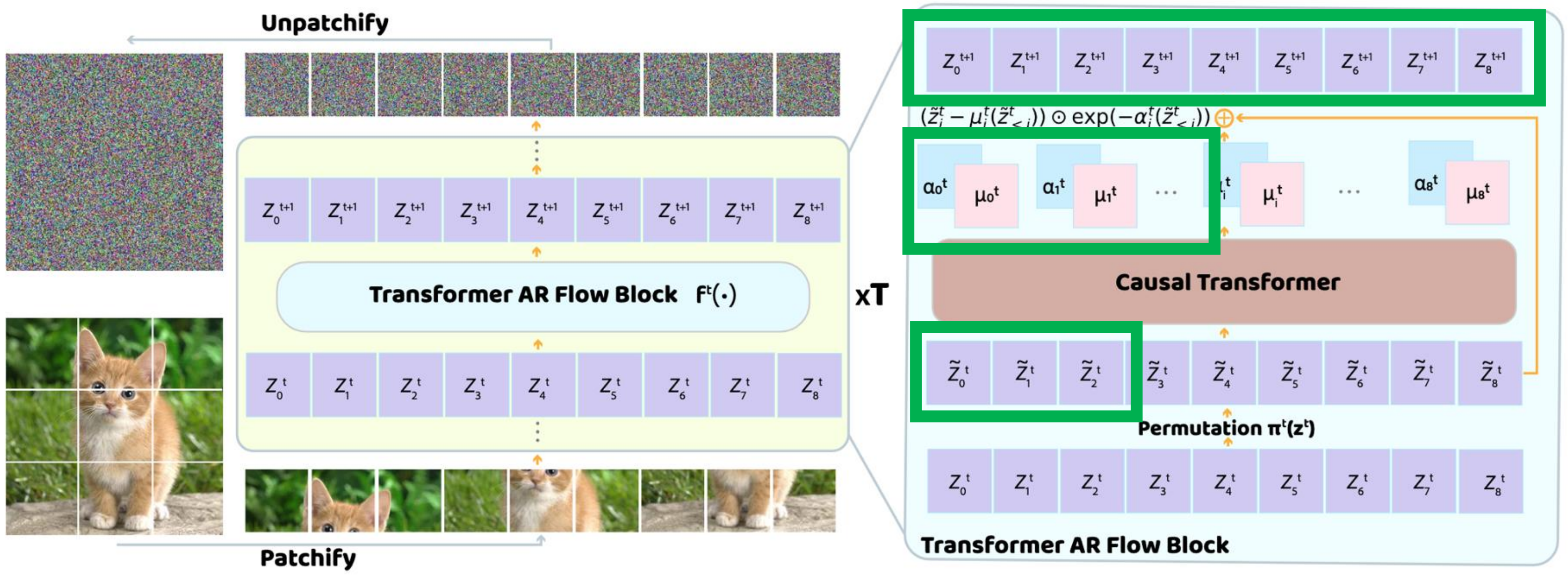
$1 \times D$ $\quad$ $1 \times D$ $\quad$ $1 \times D$ $\quad$ $1 \times D$

Backward Pass (Generation)





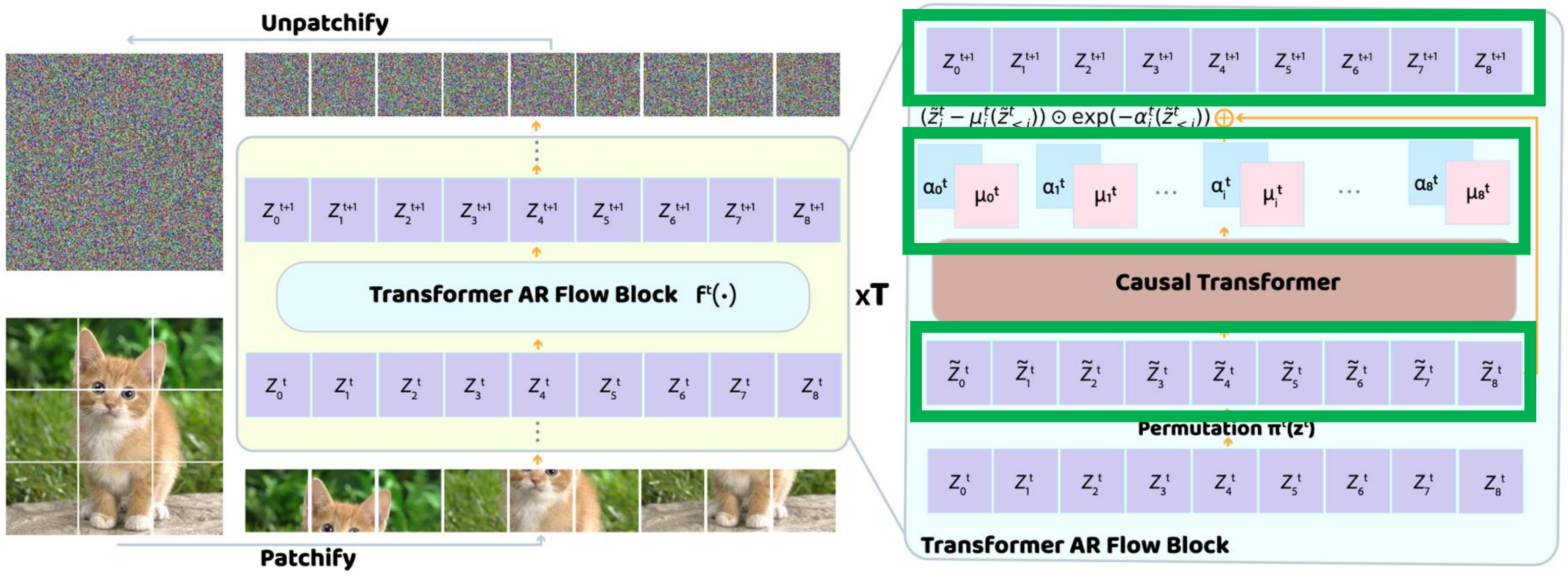
Backward Pass (Generation)



$$\tilde{Z}_i^t \leftarrow Z_i^{t+1} \odot \exp\left(\alpha_i^t\right) + \mu_i^t$$

$1 \times D$ $\quad$ $1 \times D$ $\quad$ $1 \times D$ $\quad$ $1 \times D$

Backward Pass (Generation)



$$\tilde{z}_i^t \leftarrow z_i^{t+1} \odot \exp\left(\alpha_i^t\right) + \mu_i^t$$

$1 \times D \qquad 1 \times D \qquad 1 \times D \qquad 1 \times D$

Normalizing Flows

We wish to learn a transformation $f(x) = z$ that

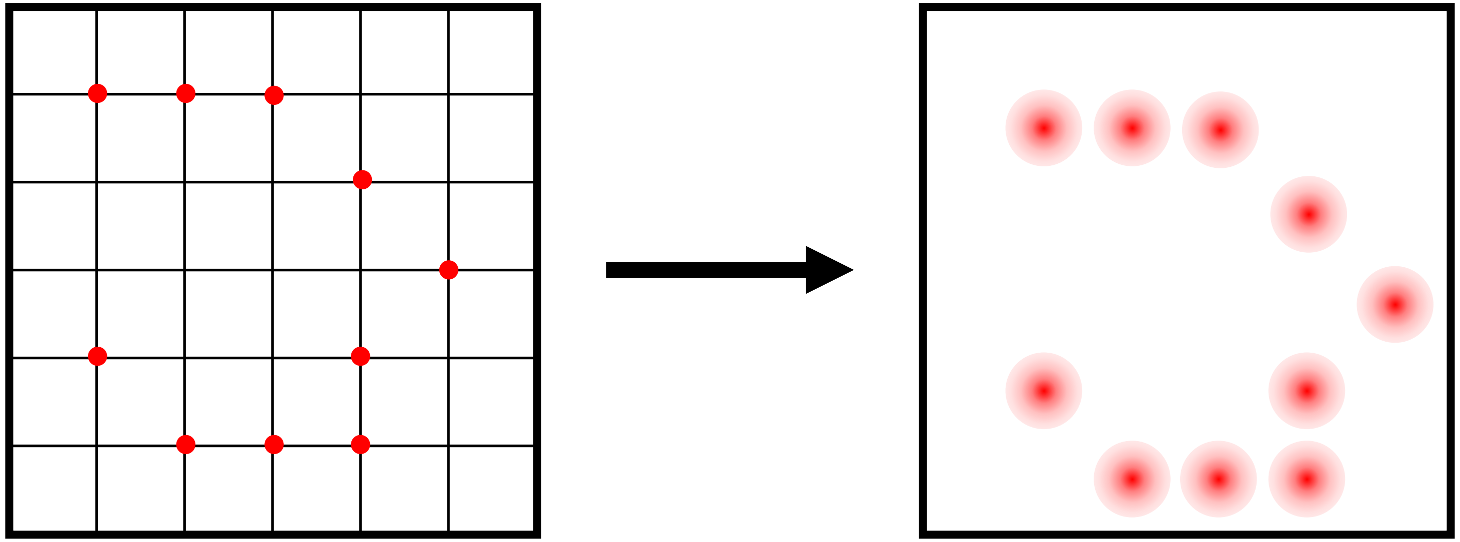
1) Is invertible

2) Has a tractable determinant of the Jacobian (triangular Jacobian)

$$p_{model}(x) = p_0\big(f(x)\big)\left|\det\frac{df(x)}{dx}\right|$$

Table of Contents

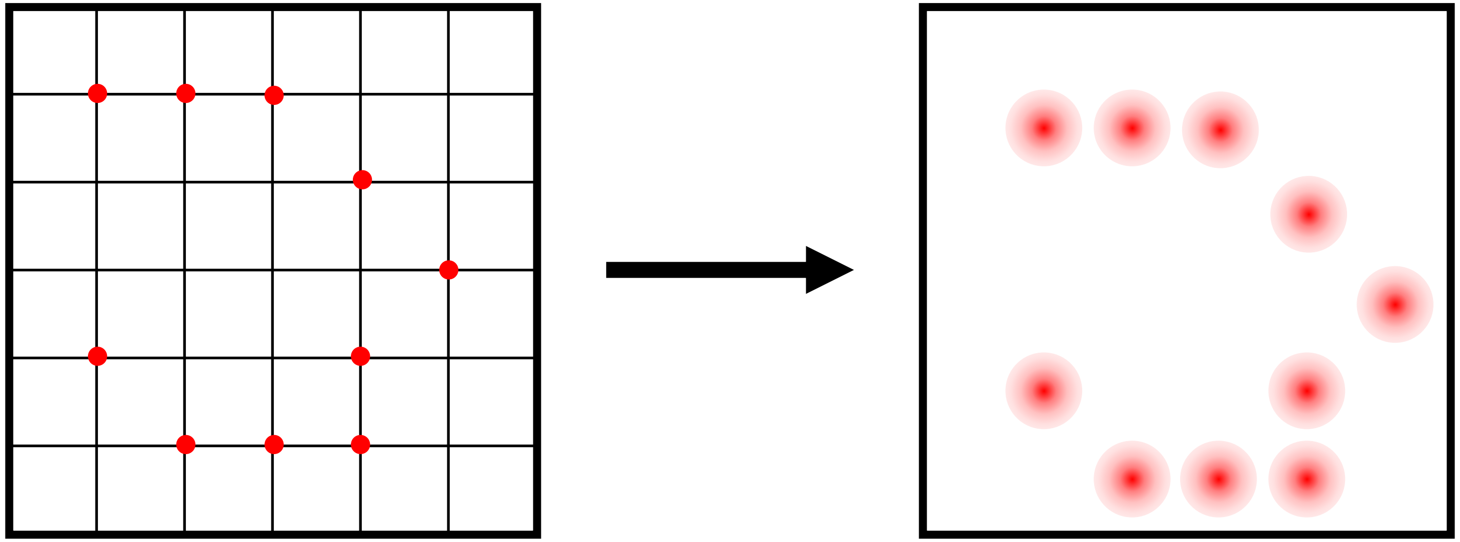
Noise Augmented Training



Problem: Input data is pixel-quantized images

Solution: Applying Gaussian noise to inputs provides more support in the input space.

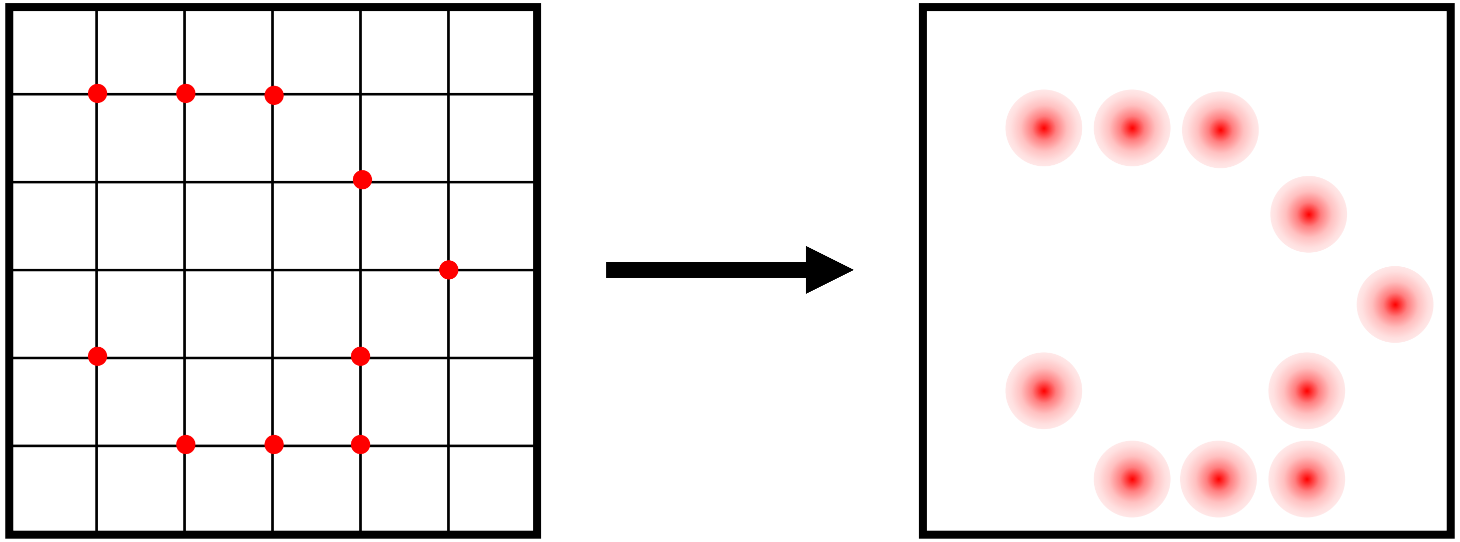
Noise Augmented Training



$x \sim p_{data}$

$$\epsilon \sim N(0, \sigma^2 I)$$
$$y = x + \epsilon$$
$$y \sim q$$

Noise Augmented Training



$x \sim p_{data}$

$$\epsilon \sim N(0, \sigma^2 I)$$
$$y = x + \epsilon$$
$$y \sim q$$

In practice, σ is 25x larger than the pixel quantization distance.

Score Based Denoising

$$x \sim p_{data}$$

$$\epsilon \sim N(0, \sigma^2 I)$$

$$y = x + \epsilon$$

$$y \sim q$$

Problem: TarFlow is trained on noisy distribution q(y). So it will generate noisy images.
Solution: Denoise output

Score Based Denoising

$$x \sim p_{data}$$

$$\epsilon \sim N(0, \sigma^2 I)$$

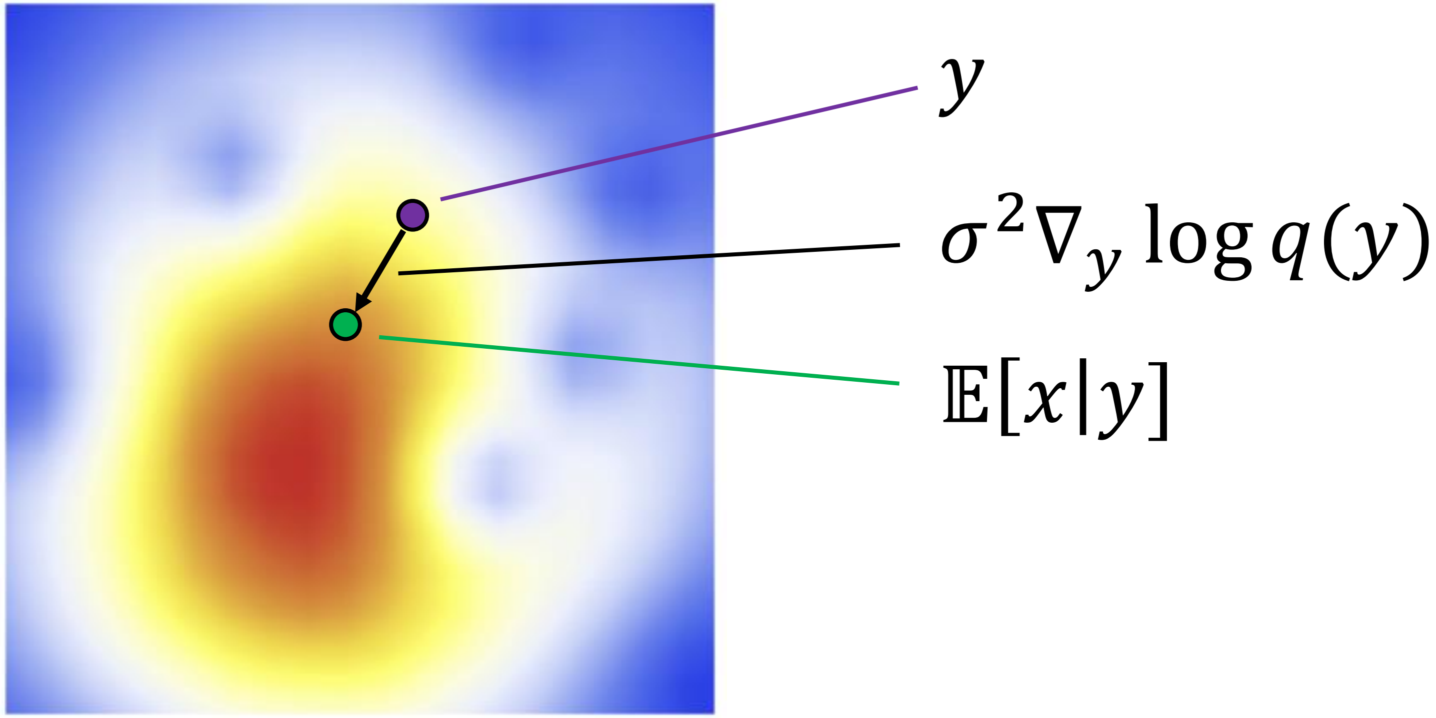
$$y = x + \epsilon$$

$$y \sim q$$

$$\mathbb{E}[x|y] = y + \underbrace{\sigma^2 \nabla_y \log q(y)}_{\text{Score}}$$

Problem: Training with noise augmentation leads to noisy generated outputs.
Solution: Denoise output

Score Based Denoising

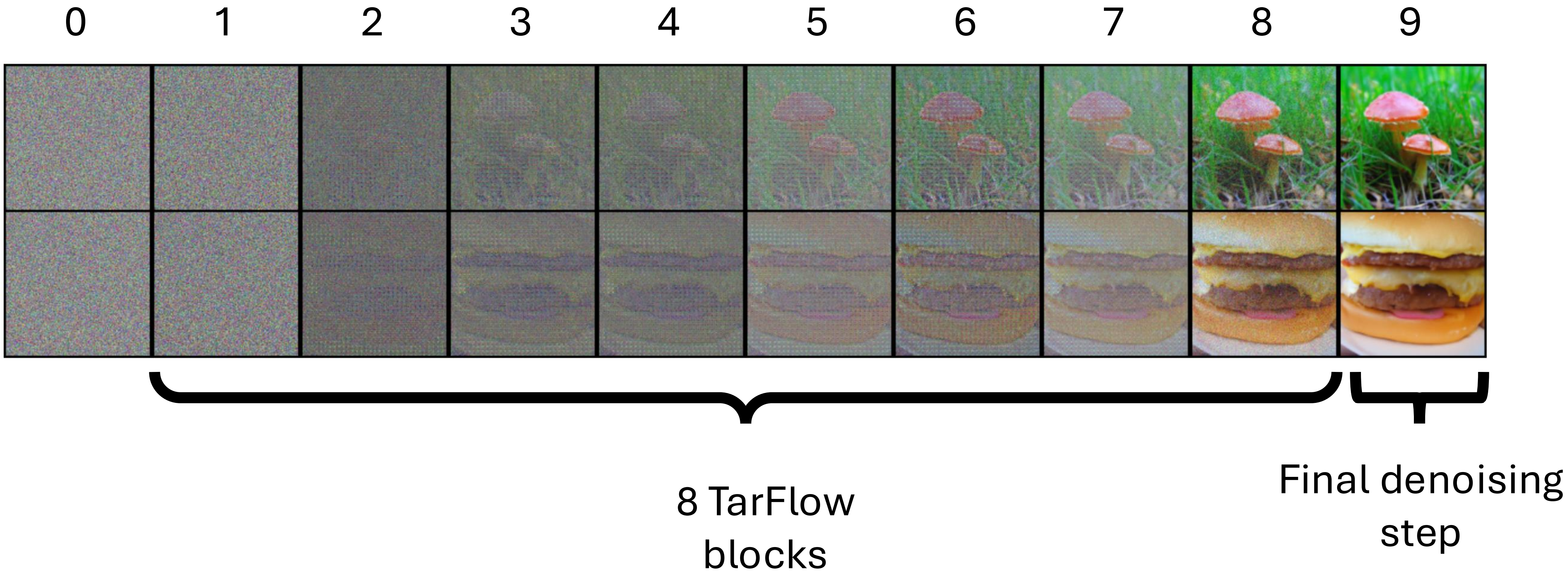


Learned TarFlow distribution $q(y)$

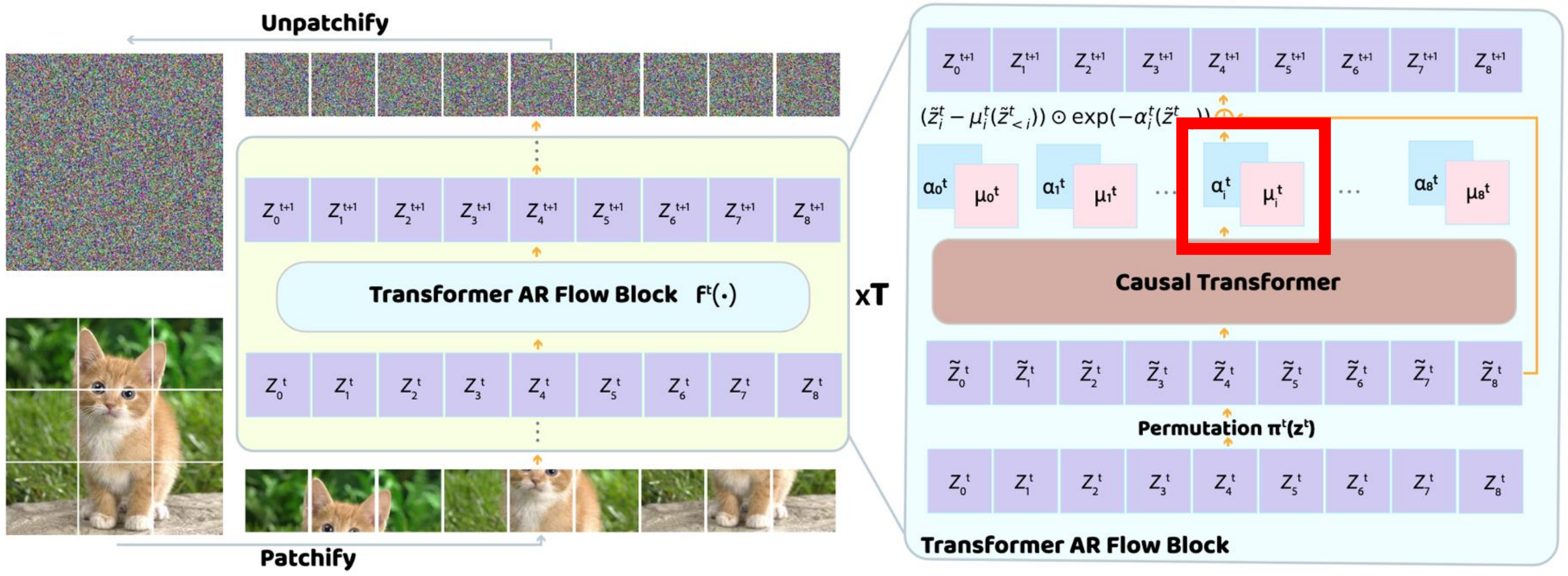
$$\mathbb{E}[x|y] = y + \sigma^2\nabla_y \log q(y)$$

Density distribution figure from D. J. Rezende and S. Mohamed, "Variational Inference with Normalizing Flows," Jun. 14, 2016, *arXiv*: arXiv:1505.05770. doi: 10.48550/arXiv.1505.05770.

Score Based Denoising



Conditional Generation



$\mu_i^t(\tilde{z}_{<i}^t; c)$ $\quad$ $\alpha_i^t(\tilde{z}_{<i}^t; c)$ $\quad \leftarrow \quad$ **Causal Transformer** $(\tilde{z}_0^t \; \tilde{z}_1^t \; \tilde{z}_2^t \; \ldots \; \tilde{z}_{i-1}^t \; \tilde{z}_i^t \; \tilde{z}_{i+1}^t \; \ldots \; \tilde{z}_N^t \; ; c)$

$1 \times D$ $\quad$ $1 \times D$

Guidance

(a) guidance $w = 0$ (no guidance)

Problem: Conditioning on class c works, but the resulting images are distorted.
Solution: Use guidance, an idea used with diffusion models.

Guidance

$$\tilde{\mu}_i^t\left(\tilde{z}_{<i}^t; c, w\right)$$
$$= (1 + w)\mu_i^t\left(\tilde{z}_{<i}^t; c\right) \; - w\mu_i^t\left(\tilde{z}_{<i}^t; \emptyset\right)$$

Guidance

Class ∅ is trained with a mix of images from all classes

$$\tilde{\mu}_i^t\left(\tilde{z}_{<i}^t; c, w\right) \\ = (1+w)\mu_i^t\left(\tilde{z}_{<i}^t; c\right) - w\mu_i^t\left(\tilde{z}_{<i}^t; \emptyset\right)$$

Guidance

Class ∅ is trained with a mix of images from all classes

$$\tilde{\mu}_i^t\left(\tilde{z}_{<i}^t; c, w\right) = \underbrace{(1+w)\mu_i^t\left(\tilde{z}_{<i}^t; c\right)}_{\text{Class c conditioned}} - \underbrace{w\mu_i^t\left(\tilde{z}_{<i}^t; \emptyset\right)}_{\text{Average conditioned}}$$

Guidance

Class ∅ is trained with a mix of images from all classes

$$\tilde{\mu}_i^t\left(\tilde{z}_{<i}^t; c, w\right) = \underbrace{(1+w)\mu_i^t\left(\tilde{z}_{<i}^t; c\right)}_{\text{Class c conditioned}} - \underbrace{w\mu_i^t\left(\tilde{z}_{<i}^t; \emptyset\right)}_{\text{Average conditioned}}$$

Idea: move away from the “average” to get a more distinctly class c-like image.

Guidance

(a) guidance $w = 0$ (no guidance)

(b) guidance $w = 2$

(c) guidance $w = 6$

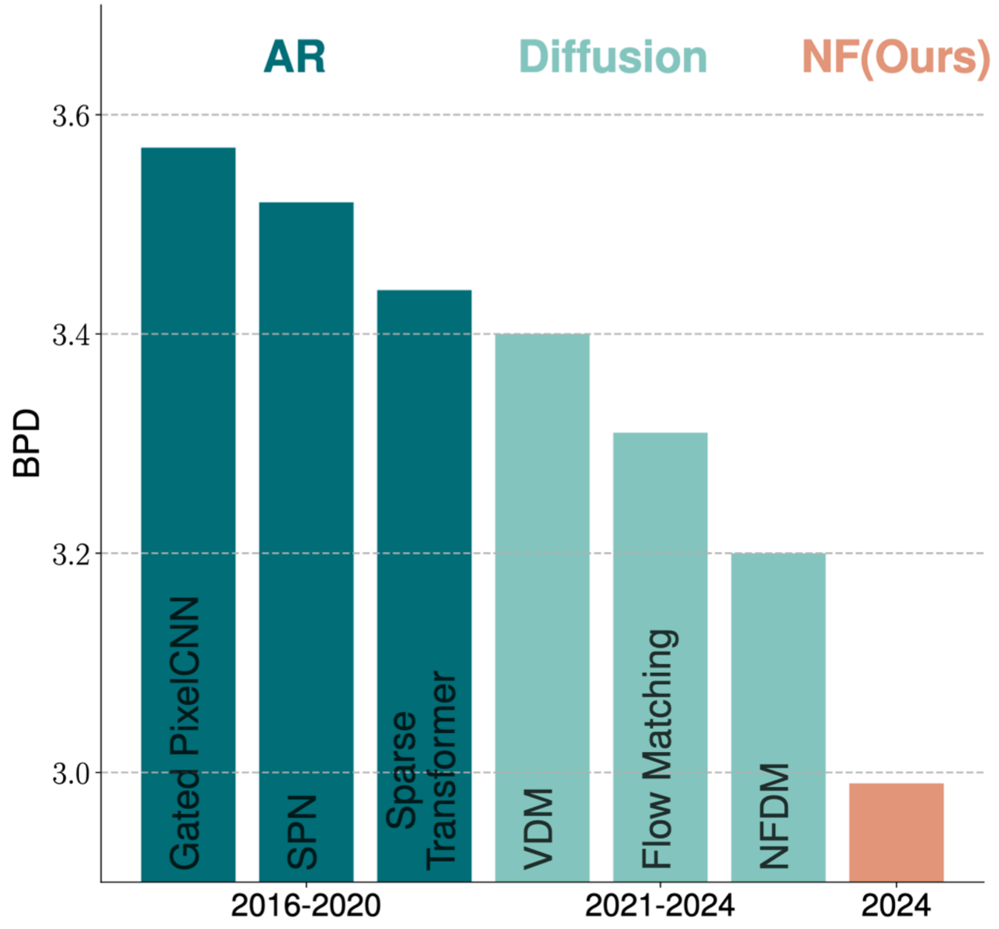
$$\tilde{\mu}_i^t\left(\tilde{z}_{<i}^t; c, w\right) = (1 + w)\mu_i^t\left(\tilde{z}_{<i}^t; c\right) - w\mu_i^t\left(\tilde{z}_{<i}^t; \emptyset\right)$$
$$\tilde{\alpha}_i^t\left(\tilde{z}_{<i}^t; c, w\right) = (1 + w)\alpha_i^t\left(\tilde{z}_{<i}^t; c\right) - w\alpha_i^t\left(\tilde{z}_{<i}^t; \emptyset\right)$$

Table of Contents

Experiments

Table 2. Bits per dim evaluation on unconditional ImageNet 64x64 test set. We denote the TarFlow configuration in the format [P-Ch-T-K-p_ϵ].

Model	Type	BPD ↓
Very Deep VAE (Child, 2021)	VAE	3.52
Glow (Kingma & Dhariwal, 2018)	Flow	3.81
Flow++ (Ho et al., 2019)	Flow	3.69
PixelCNN (van den Oord et al., 2016a)	AR	3.83
SPN (Menick & Kalchbrenner, 2019)	AR	3.52
Sparse Transformer (Child et al., 2019)	AR	3.44
Routing Transformer (Roy et al., 2021)	AR	3.43
Improved DDPM (Nichol & Dhariwal, 2021)	Diff/FM	3.54
VDM (Kingma et al., 2021)	Diff/FM	3.40
Flow Matching (Lipman et al., 2023a)	Diff/FM	3.31
NFDM (Bartosh et al., 2024)	Diff/FM	3.20
TarFlow [2-768-8-8-$\mathcal{U}(0, \frac{1}{128})$] (Ours)	NF	**2.99**



Experiments

Table 3. Fréchet Inception Distance (FID) evaluation on Conditional ImageNet 64×64. We denote the TARFLOW configuration in the format [P-Ch-T-K-p_ϵ].

Model	Type	FID ↓
EDM (Karras et al., 2022)	Diff/FM	$\boxed{1.55}$
iDDPM (Nichol & Dhariwal, 2021)	Diff/FM	$\boxed{2.92}$
ADM(dropout) (Dhariwal & Nichol, 2021)	Diff/FM	$\boxed{2.09}$
IC-GAN (Casanova et al., 2021)	GAN	6.70
BigGAN (Brock et al., 2019)	GAN	4.06
CD(LPIPS)(Song et al., 2023)	CM	4.70
iCT-deep(Song & Dhariwal, 2023)	CM	3.25
TARFLOW [4-1024-8-8-$\mathcal{N}(0, 0.05^2)$] (Ours)	NF	3.99
TARFLOW [2-768-8-8-$\mathcal{N}(0, 0.05^2)$] (Ours)	NF	2.90
TARFLOW [2-1024-8-8-$\mathcal{N}(0, 0.05^2)$] (Ours)	NF	2.66

Table 4. Fréchet Inception Distance (FID) evaluation on Conditional ImageNet 128×128. We denote the TARFLOW configuration in the format [P-Ch-T-K-p_ϵ].

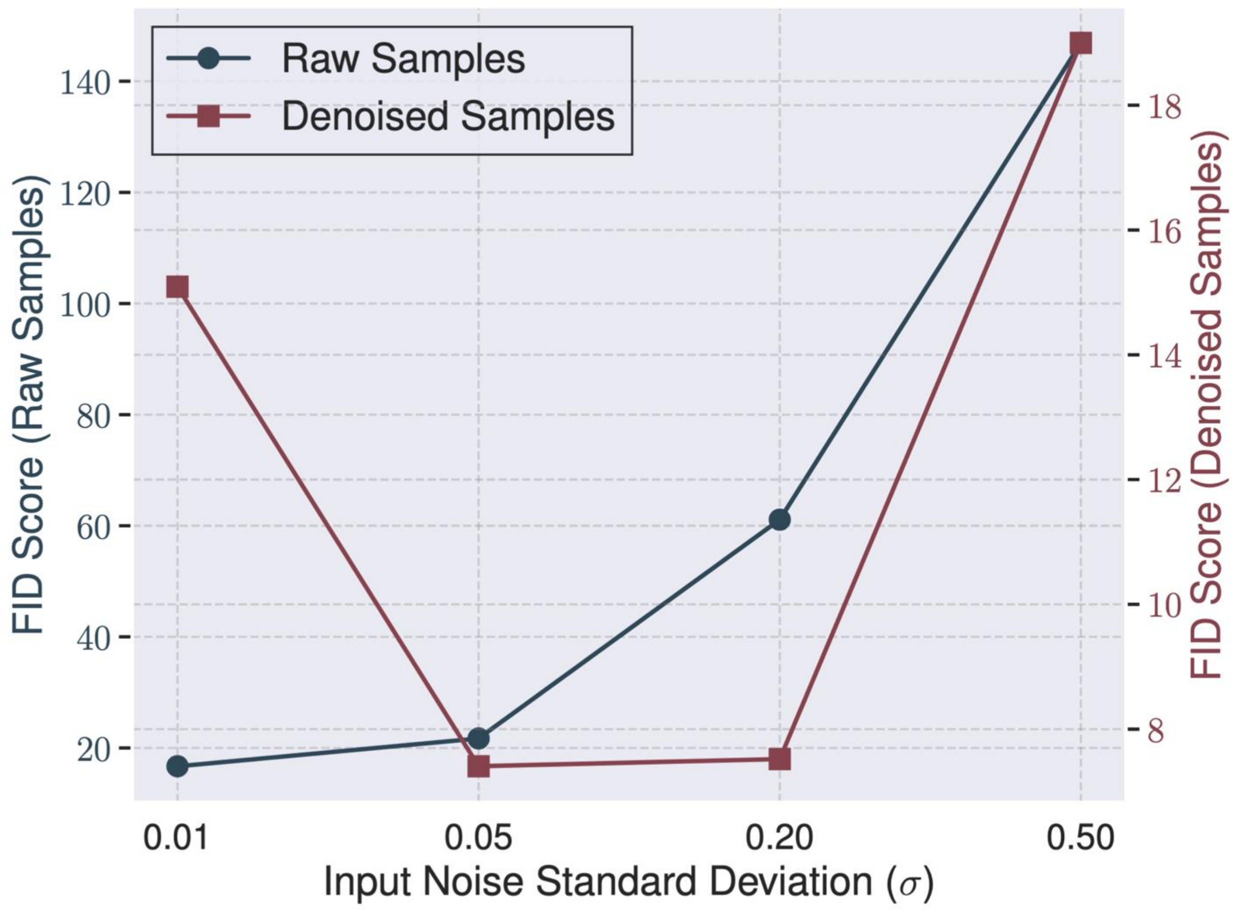
Model	Type	FID ↓
ADM-G (Dhariwal & Nichol, 2021)	Diff/FM	$\boxed{2.97}$
CDM (Ho et al., 2022)	Diff/FM	$\boxed{3.52}$
Simple Diff (Hoogeboom et al., 2023)	Diff/FM	$\boxed{1.94}$
RIN (Jabri et al., 2023)	Diff/FM	$\boxed{2.75}$
BigGAN (Brock et al., 2019)	GAN	8.70
BigGAN-deep (Brock et al., 2019)	GAN	5.70
TARFLOW [4-1024-8-8-$\mathcal{N}(0, 0.05^2)$] (Ours)	NF	5.29
TARFLOW [4-1024-8-8-$\mathcal{N}(0, 0.15^2)$] (Ours)	NF	5.03

TarFlow does not consistently outperform diffusion models

Qualitative Results

(d) guidance $w = 2$, denoised

Experiments: Denoising



Experiments: Guidance

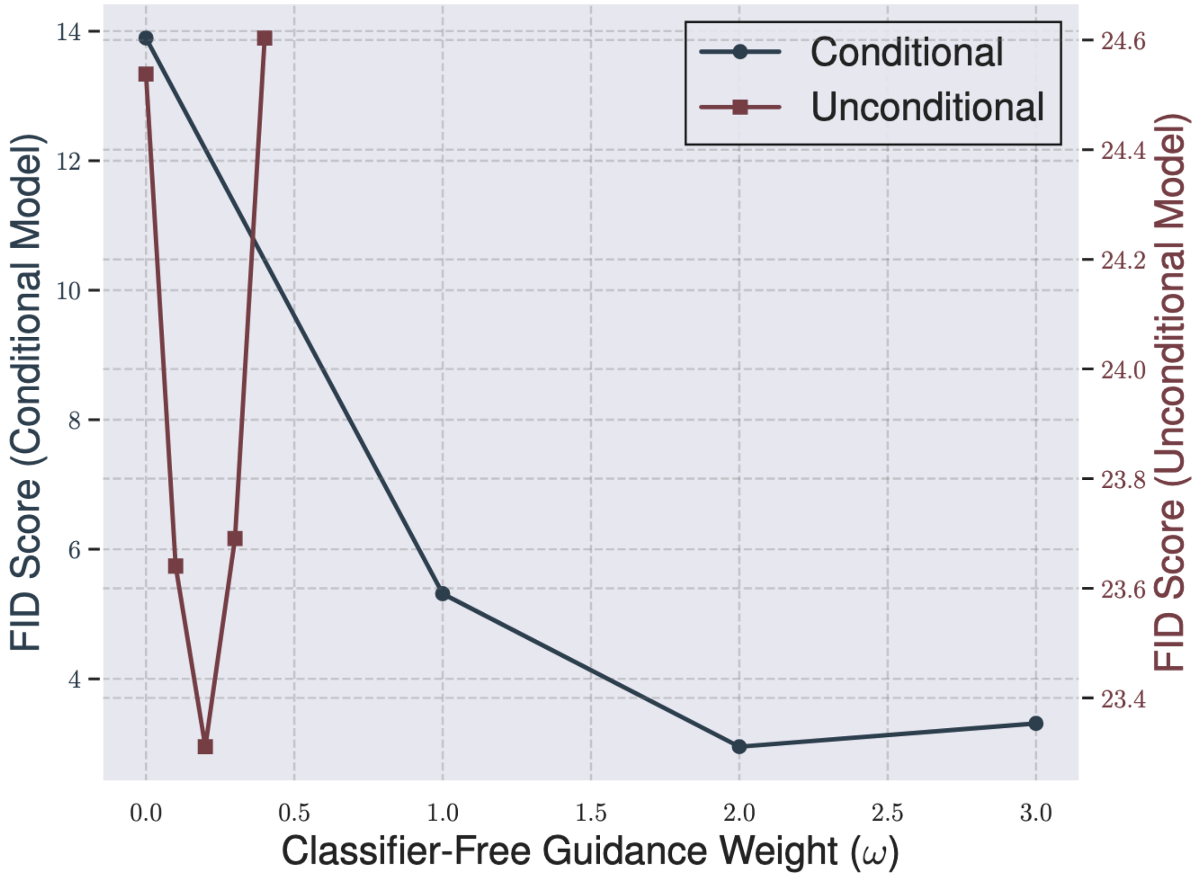


Table of Contents

Conclusion

- TarFlow uses transformers for an autoregressive NF block
    - Uses noise augmented training and denoising for better performance
    - Uses guidance for better performance
- TarFlow shows strong performance compared with prior NF models
    - But does not outperform diffusion models

Questions

- How does injecting that much noise during training not distort what distribution you're actually learning?
- Is the trade-off between tractable exact likelihood and sequential sampling fundamental to autoregressive normalizing flows, or can both likelihood and fast parallel generation be achieved?
- If TARFLOW relies heavily on Gaussian noise augmentation and Tweedie's formula for denoising, is this architecture simply converging on the mechanics of diffusion models from a different theoretical starting point?
- The main problem is why the author use causal transformer but not bidirectional transformer. Autoregression makes the Jacobian tractable and preserves exact invertibility, but the paper does not clearly establish whether this is the best factorization.