Normalizing Flows are Capable Generative Models

Shuangfei Zhai, Ruixiang Zhang, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Angel Bautista, Navdeep Jaitly, Josh Susskind
ICML 2025 · Apple

TLDR

- We show that Normalizing Flows trained with the change of variable formula can work surprisingly well as generative models

Background: Normalizing Flows

- Learn a deterministic function that transforms data to noise with a likelihood loss

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

- Allows for sampling by reversing the function starting from noise

$$z \sim \mathcal{N}(0,I),\ x = f^{-1}(z)$$

- Challenge: finding functions with easy to compute Jacobian determinants

Background: Autoregressive Flows

- Autoregressive affine transformations are invertible

forward: $z_i = (x_i - \mu(x_{<i})) \odot \exp(-a(x_{<i}))$

reverse: $x_i = z_i \odot \exp(a(x_{<i})) + \mu(x_{<i})$

- They also have tractable Jacobian determinants

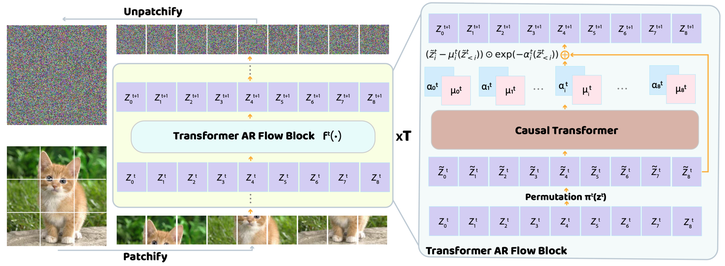
$$\log(|det(\frac{dz}{dx})|) = -\sum_i a(x_{<i})$$

Method: Transformer Autoregressive Flow (TarFlow)

Step 1: a powerful architecture

- Stacked autoregressive flows with alternating directions, denoted by $\pi^t(\cdot)$
- Each flow is implemented with a causal (Vision) Transformer
- All flows trained end to end with the likelihood loss

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



Step 2: Gaussian noise augmented training

- We found it crucial to add a small but non-negligible amount of Gaussian noise to the inputs for good sampling
- Gaussian noise densifies the training distributions and improves generalization

Model y instead of x: $x \sim p_{data}, \epsilon \sim \mathcal{N}(0, I\sigma^2), y = x + \epsilon$

Step 3: Score based denoising

- Because we model the noisy inputs, samples will appear noisy as well
- We can do this without training another model — with the help of Tweedie's formula
- If we know the density of y, then we can derive its score, which is exactly what we need for denoising

$$z \sim p_0, y = f^{-1}(z), x = y + \sigma^2 \nabla_y \log p_{model}(y)$$

Step 4: Guidance

Conditional model

- Extrapolate between the conditional and unconditional predictions

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

Unconditional model

- Create an unconditional prediction equivalent by injecting a temperature term τ to the attention layers

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

Results: SOTA Likelihood & Competitive FID

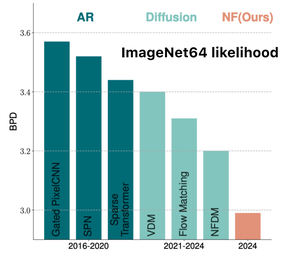
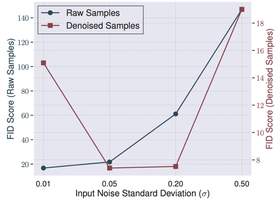


Table 3. Fréchet Inception Distance (FID) evaluation on Conditional ImageNet 64×64. We denote the TarFlow configuration in the format [P-Ch-T-K-p_e].

Model	Type	FID ↓
EDM (Karras et al., 2022)	Diff/FM	1.55
iDDPM (Nichol & Dhariwal, 2021)	Diff/FM	2.92
ADM(dropout) (Dhariwal & Nichol, 2021)	Diff/FM	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

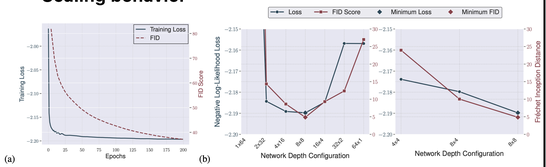
Ablations

Gaussian Noising

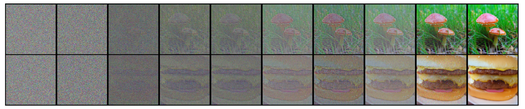


Guidance

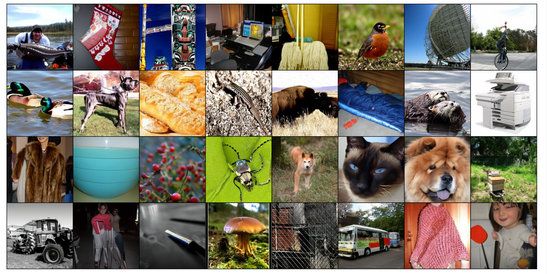
Scaling behavior



Sampling trajectory



ImageNet256 Samples



TM and © 2025 Apple Inc. All rights reserved.