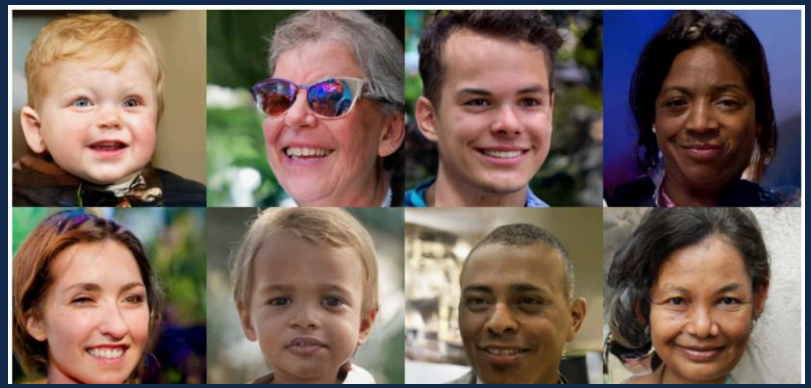
Very Deep VAEs Generalize Autoregressive Models and Can Outperform Them on Images

Rewon Child, OpenAI

ICLR 2021 · arXiv:2011.10650

Presented by Yuvraj Jain
September 3, 2026



Samples from the model on FFHQ-256. Figure 1, Child (2021).

Two ways to build a generative model of images

Both are trained to maximize log p(x), and both report negative log-likelihood (NLL) in bits per dimension. So they compare directly.

Autoregressive

$p(x) = \Pi_i \, p(x_i \mid x_{<i})$

Generate one value at a time, each one conditioned on everything already generated. No latent variables.

Latent variable (VAE)

sample z, then generate x from z

A compressed description of the scene comes first, the pixels follow. Trained on the ELBO, so the reported number is an upper bound on the true NLL.

The VAE reports a bound, the autoregressive model reports an exact value.

Autoregressive models won on likelihood for four years

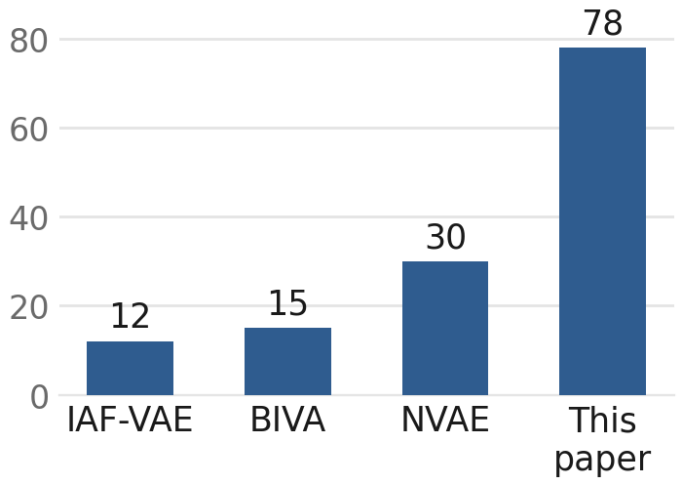
- Starting with PixelCNN in 2016, a convolutional model that predicts each pixel from the pixels before it
- That family held the best likelihood on every natural image benchmark
- Strange, because such a model has no notion of an underlying scene, only pixel-to-pixel dependencies
- Images are observations of scenes, so latent variables ought to be the better fit

---

Is the autoregressive assumption genuinely better, or were VAEs simply built wrong?

A hierarchical VAE samples in stages, not all at once

- A plain VAE samples one z. A hierarchical VAE samples a sequence z_0, z_1, ..., z_n
- Each group is conditioned on everything sampled before it
- For images, each group is a feature map: z_0 tiny and coarse at the top, z_n full resolution at the bottom
- One sampling step = one group: the network outputs μ and σ, then z is drawn



The three deepest hierarchical VAEs published before this paper, and this paper's model.

This sequence length is what we'll call depth: how many times the model can revise its decision before generating the image.

Four claims, in the order the paper makes them

1

Theory

A hierarchical VAE with enough depth can represent any autoregressive model, and can also find shorter generative paths when they exist.

2

Architecture

A minimal top-down ResNet VAE that trains stably past 70 steps of depth. Prior work topped out at 30.

3

Evidence

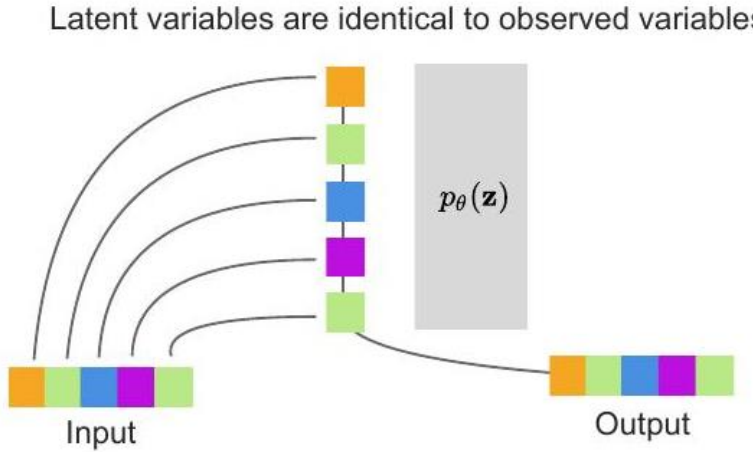
Holding parameters fixed, more depth means better likelihood. Depth is the causal variable, not capacity.

4

Results

Beats every PixelCNN-based model on CIFAR-10, ImageNet-32, ImageNet-64 and FFHQ, with fewer parameters and one-pass sampling.

Proposition 1, step one: make the encoder do nothing



$q(z_i = x_i \mid z_{<i}, x) = 1$

the encoder copies value i into latent i

$p(x_i = z_i \mid z) = 1$

the decoder copies it straight back out

$$p(z) = p(z_0)\, p(z_1|z_0) \cdots p(z_n|z_{<n})$$

already a chain of conditionals — substitute $z_i = x_i$ and it's the same chain over x

---

Encoder and decoder now do no work at all. The only component left is the prior p(z).

Figure 2, left.

An autoregressive model turns out to be a special case of a VAE

If autoregressive models are a special case of VAEs, why were they winning?

- Reaching that special case needs one sampling step per value in the image
- Every VAE built up to this point had about thirty sampling steps
- For a 32×32 image the theory asks for about three thousand (32 x 32 x 3)

VAEs were not worse. They were about a hundred times too shallow.

Do we really need three thousand steps?

- Three thousand is the worst case: every step just copies one pixel forward
- If a value is already predictable once a few earlier ones are known, it does not need its own step
- Values like that can be produced together, in the same step
- Images are mostly fine texture once the overall scene is set and texture is exactly this kind of predictable

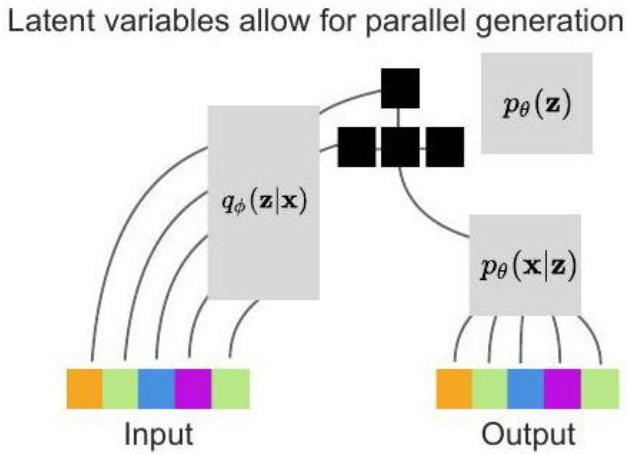


Figure 2, right.

So the real number of steps needed is somewhere between thirty and three thousand — probably much closer to the low end.

The architecture is deliberately plain

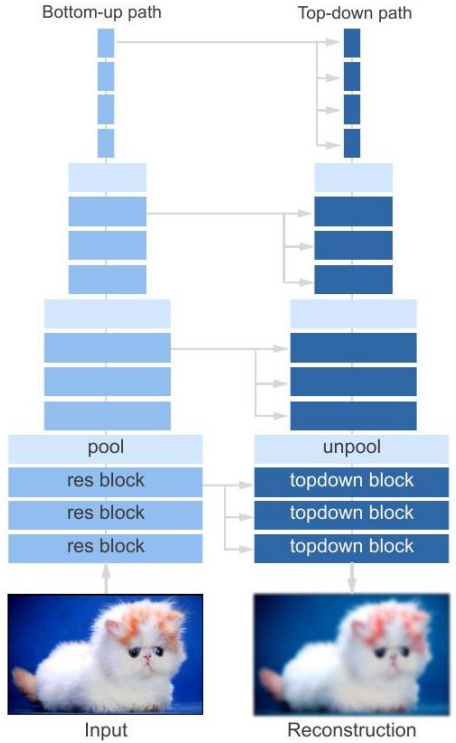


Figure 3, left half.

- Bottom-up: a deterministic pass over the image builds features. Training only
- Top-down: one latent group per block, coarse resolution to fine. Always
- At generation there is no image, so each group is drawn from the prior instead
- Only convolutions, nonlinearities and Gaussian sampling

Nothing clever, so that any gain can be attributed to depth rather than to architecture.

Depth had to be made trainable before it could be tested

- Scale down the last convolution in each residual block
- Change how feature maps are upsampled, which removed the collapse of unused layers
- Skip the rare update whose gradient explodes

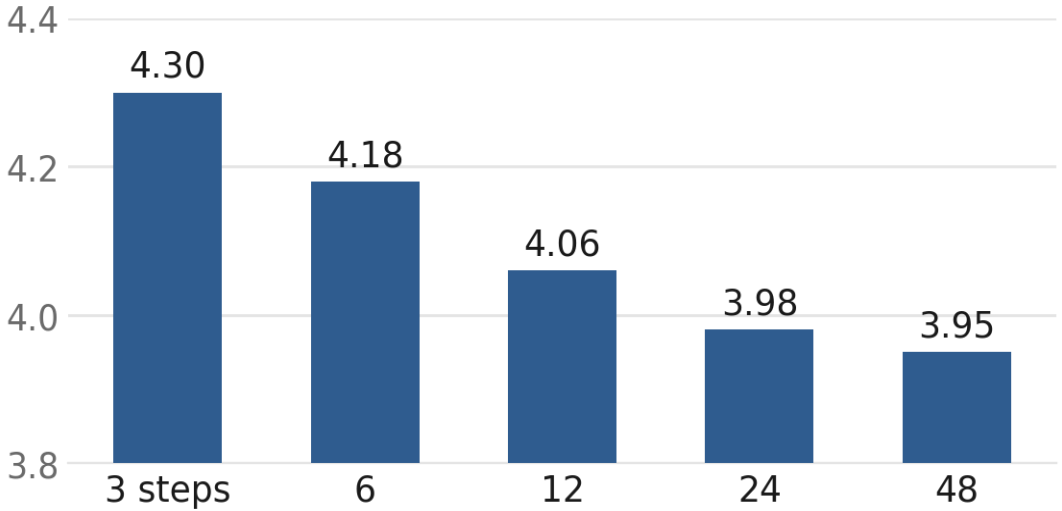
Sampling steps	Without scaling	With scaling
15	2.50	2.51
30	2.36	2.38
45	2.31	2.30
60	2.30	2.29
75	**Diverged**	**2.28**

Table 3. Test loss in bits per dimension, lower is better.

This is why nobody had run the experiment before: past about thirty steps, these networks simply diverge.

The key experiment: change depth, hold everything else fixed

Same 41M parameters and same 48 blocks in every run. Depth is reduced by feeding consecutive blocks the same input, so no weights are removed.



- Same capacity, same weights, only the number of sampling steps changes
- Still improving at 48 steps, past anything tried before

The gain cannot be explained by capacity. Depth is doing the work.

Table 1. Lower is better.

With enough depth, the VAE wins

Dataset	Model	Type	Params	Steps	NLL
CIFAR-10	PixelCNN++	AR	53M	many	2.92
	NVAE	VAE	131M	30	2.91
	Very Deep VAE	**VAE**	**39M**	**45**	**2.87**
ImageNet-32	Gated PixelCNN	AR	177M	many	3.83
	NVAE	VAE	268M	28	3.92
	Very Deep VAE	**VAE**	**119M**	**78**	**3.80**

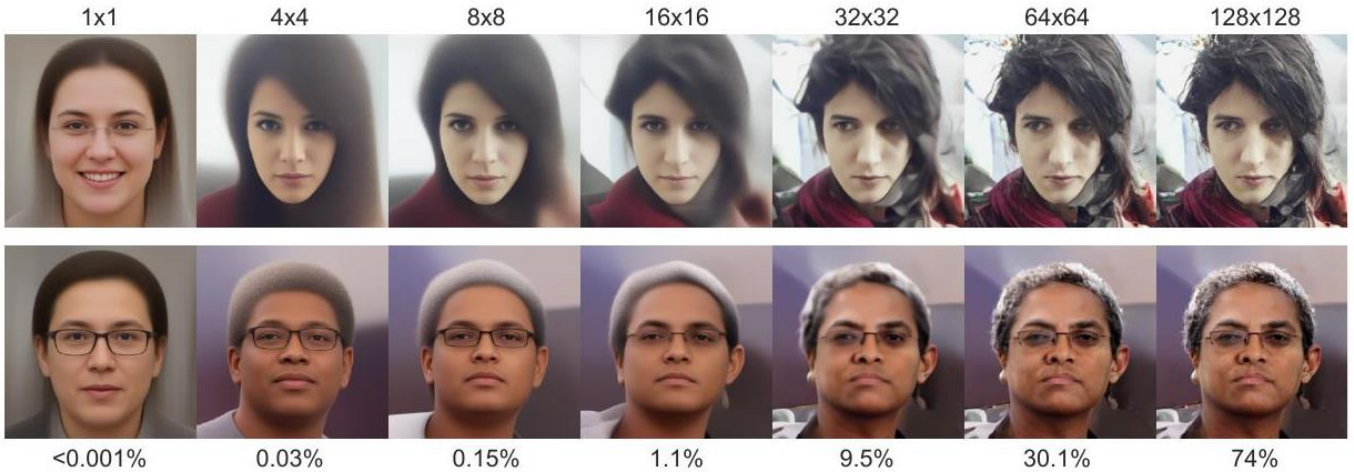
ImageNet-32 in one row: NVAE used 268M parameters and 28 steps to score 3.92. This one used 119M and 78 steps to score 3.80.

Honest caveat: this beats the PixelCNN family and every non-autoregressive model, but attention-based autoregressive models still score better on two of these datasets.

Selected rows from Table 2. Lower is better. Why NLL and not FID: this is a density-estimation comparison, and FID is not defined for autoregressive models.

The model discovers the coarse-to-fine hierarchy on its own

Each column: take a real image, keep its latents down to that resolution, and let the model invent everything finer.



- By 16×16, four columns in, the person is already fixed, using 1.1% of the latents
- Everything after that is texture, and it is independent enough to be emitted in parallel

---

This is why 62 sampling steps suffice for an image made of 196,608 values.

Assessment

What is strong

- One hypothesis, tested with the right control: capacity held fixed
- A deliberately plain architecture wins, so the credit goes to depth
- The theory is simple and actually explains the result

What is weak

- Proposition 2 assumes densities the model cannot actually express
- Likelihood is the only thing measured; sample quality never is
- Depth and the three training fixes changed together

The model at 1024×1024. The paper reports a likelihood for it and concedes it fails to capture the distribution. Nothing in the evaluation would have caught that.

Figure 11.

Student Questions

What modern products use VAE-based models? How much have VAE-based models improved in quality since this work?

Nearly every major image and video model today (Stable Diffusion, DALL·E 3, Sora) uses a VAE to compress pixels into a latent space for a diffusion model to generate in.

Are there any disadvantages in the quality or characteristics of the generated outputs when the model learns a hierarchical conditional dependency structure, even if the likelihood performance improves?

Yes…
Early commitments can't be revised.
Maximum likelihood training tends toward mode-covering, not mode-seeking

I would like to go over some of the theory in 2.1 and 2.2.

Covered in next slide

Questions?

Child, R. (2021). Very Deep VAEs Generalize Autoregressive Models and Can Outperform Them on Images. ICLR. Code: github.com/openai/vdvae

ADDITIONAL SLIDES (IF TIME PERMITS)

Proposition 1: the objective collapses into an autoregressive model

1. Why the bound is tight

$$\log p(x) = \text{ELBO} + KL(q(z|x) \,\|\, p(z|x))$$

The encoder is exactly the true posterior (Bayes' rule forces p(z|x) to the same point mass as q), so this KL is exactly 0.

2. Expanding the ELBO into three log terms — the algebra

$\text{ELBO} = E_q[\log p(x\|z)] - D_{KL}(q(z\|x) \,\|\, p(z))$	*the ELBO, as in class*
$= E_q[\log p(x\|z)] - E_q[\log q(z\|x) - \log p(z)]$	*$D_{KL}(a\|\|b) = E_a[\log a - \log b]$*
$= E_q[\log p(x\|z)] - E_q[\log q(z\|x)] + E_q[\log p(z)]$	*distribute the minus sign*
$= E_q[\log p(x\|z) - \log q(z\|x) + \log p(z)]$	*expectation is linear*
$= E_q[\log p(x\|z) + \log p(z) - \log q(z\|x)]$	*reorder the terms*

Decoder and encoder are both point masses, so both terms are log 1 = 0 — only log p(z) survives.

$$\log p(x) = \log p(z) = \Sigma_i \log p(z_i \mid z_{<i}) = \Sigma_i \log p(x_i \mid x_{<i})$$

That is an autoregressive model. The autoregression has moved into the prior.

	Model type	Params	Depth	Sampling	NLL
CIFAR-10					
PixelCNN++ (Salimans et al., 2017)	AR	53M*		D	2.92
PixelSNAIL (Chen et al., 2017)	AR			D	2.85
Sparse Transformer (Child et al., 2019)	AR	59M		D	**2.80**
VLAE (Chen et al., 2016)	VAE			D	$\leq$ 2.95
IAF-VAE (Kingma et al., 2016)	VAE		12	1	$\leq$ 3.11
Flow++ (Ho et al., 2019)	Flow	31M		1	$\leq$ 3.08
BIVA (Maaløe et al., 2019)	VAE	103M	15	1	$\leq$ 3.08
NVAE (Vahdat & Kautz, 2020)	VAE	131M	30	1	$\leq$ 2.91
Very Deep VAE **(ours)**	VAE	39M	45	1	$\leq$ **2.87**
ImageNet-32					
Gated PixelCNN	AR	177M*	10	D	3.83
Image Transformer (Parmar et al., 2018)	AR			D	**3.77**
BIVA	VAE	103M*	15	1	$\leq$ 3.96
NVAE	VAE	268M	28	1	$\leq$ 3.92
Flow++	Flow	169M		1	$\leq$ 3.86
Very Deep VAE **(ours)**	VAE	119M	78	1	$\leq$ **3.80**
ImageNet-64					
Gated PixelCNN	AR	177M*		D	3.57
SPN (Menick & Kalchbrenner, 2018)	AR	150M		D	3.52
Sparse Transformer	AR	152M		D	**3.44**
Glow (Kingma & Dhariwal, 2018)	Flow			1	3.81
Flow++	Flow	73M		1	$\leq$ 3.69
Very Deep VAE **(ours)**	VAE	125M	75	1	$\leq$ **3.52**
FFHQ-256 (5 bit)					
NVAE	VAE		36	1	$\leq$ 0.68
Very Deep VAE **(ours)**	VAE	115M	62	1	$\leq$ **0.61**
FFHQ-1024 (8 bit)					
Very Deep VAE **(ours)**	VAE	115M	72	1	$\leq$ **2.42**