FEEC UNICAMP

Presentation 3
IA376N 2s2025

NVAE: A Deep Hierarchical Variational Autoencoder (NeurIPS 2020)

Henrique Parede de Souza - 260497

Overview

- Authors
- Problems of VAE
- Nouveau VAE
- Hierarchical Architecture
- Limitations
- Improvements
- Results
- Discussion & Conclusion

NVAE: A Deep Hierarchical Variational Autoencoder

Arash Vahdat, Jan Kautz
NVIDIA
{avahdat, jkautz}@nvidia.com

Abstract

Normalizing flows, autoregressive models, variational autoencoders (VAEs), and deep energy-based models are among competing likelihood-based frameworks for deep generative learning. Among them, VAEs have the advantage of fast and tractable sampling and easy-to-access encoding networks. However, they are currently outperformed by other models such as normalizing flows and autoregressive models. While the majority of the research in VAEs is focused on the statistical challenges, we explore the orthogonal direction of carefully designing neural architectures for hierarchical VAEs. We propose Nouveau VAE (NVAE), a deep hierarchical VAE built for image generation using depth-wise separable convolutions and batch normalization. NVAE is equipped with a residual parameterization of Normal distributions and its training is stabilized by spectral regularization. We show that NVAE achieves state-of-the-art results among non-autoregressive likelihood-based models on the MNIST, CIFAR-10, CelebA 64, and CelebA HQ datasets and it provides a strong baseline on FFHQ. For example, on CIFAR-10, NVAE pushes the state-of-the-art from 2.98 to 2.91 bits per dimension, and it produces high-quality images on CelebA HQ as shown in Fig. 1. To the best of our knowledge, NVAE is the first successful VAE applied to natural images as large as 256×256 pixels. The source code is available at https://github.com/NVlabs/NVAE.

Authors

Arash Vahdat

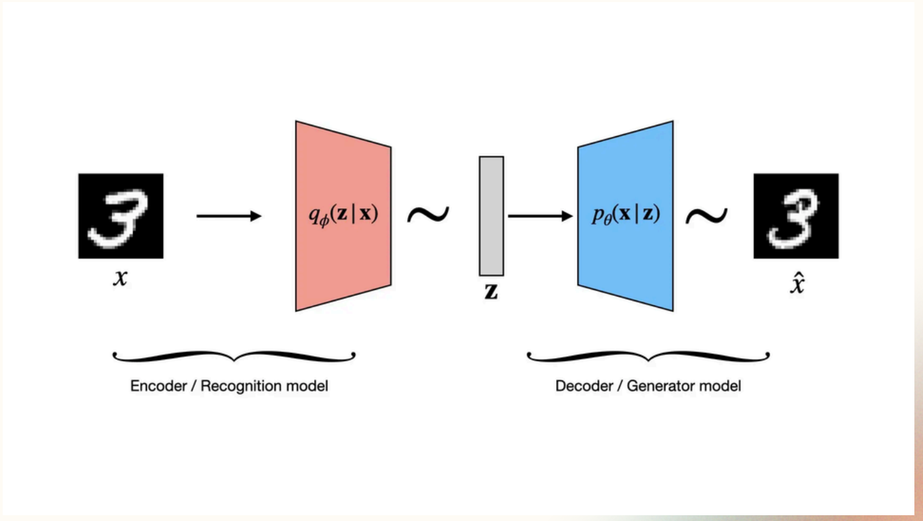
Jan Kautz

- Vahdat is Research Director of fundamental generative AI research (**GenAIR**) team at NVIDIA Research.
- Jan Kautz is Vice President of Learning and Perception Research at NVIDIA.
- Co-authored papers on Generative Models like NVAE and Diffusion Models.

Main areas of study: Generative Models, Diffusion Models, Computer Vision, Image Synthesis

Problem

VAEs are currently **outperformed** by normalizing flows and autoregressive models.



Source

- Besides the **fast sampling** and **easy-to-access encoding layers**, VAEs have been outperformed by recent models.
- Most of the research regarding VAEs' improvement only focus on the **statistical challenges**.
- The role of **neural network architectures** is frequently overlooked.

Frequent Problems

Encoder Overffiting

- The marginal log-likelihood, estimated by non-encoder-based methods, is not sensitive to the encoder overfitting.

Lack of long-range correlations

- The neural networks for VAEs should model long-range correlations in data, requiring the networks to have large receptive fields.

Deep hierarchical VAEs training instability

- Due to the unbounded Kullback–Leibler (KL) divergence in the variational lower bound, training very deep hierarchical VAEs is often unstable.

Batch Normalization omission

- Current SOTA VAEs omit batch normalization to combat the sources of randomness that could potentially amplify their instability.

Nouveau VAE (NVAE)

Deep **hierarchical** VAE with network architecture that produces **high-quality images**.

Figure 1: 256×256-pixel samples generated by NVAE, trained on CelebA HQ [28].

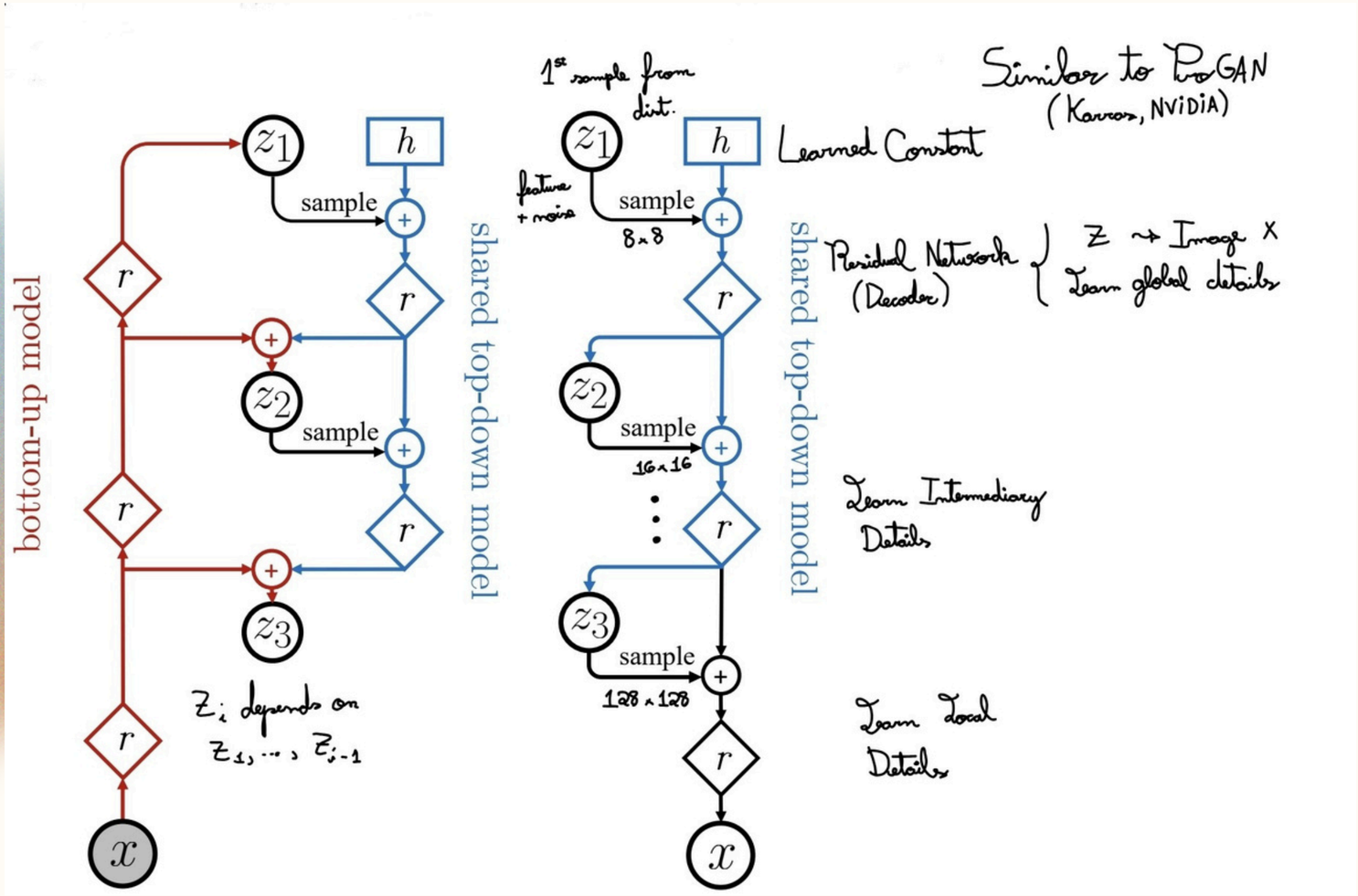
Stated NVAE as the **new SOTA** among non-autoregressive models, being the first successful application of VAEs to images as large as 256×256 pixels.

Based on **depthwise convolutions**.

Architecture

Hierarchical Decoder

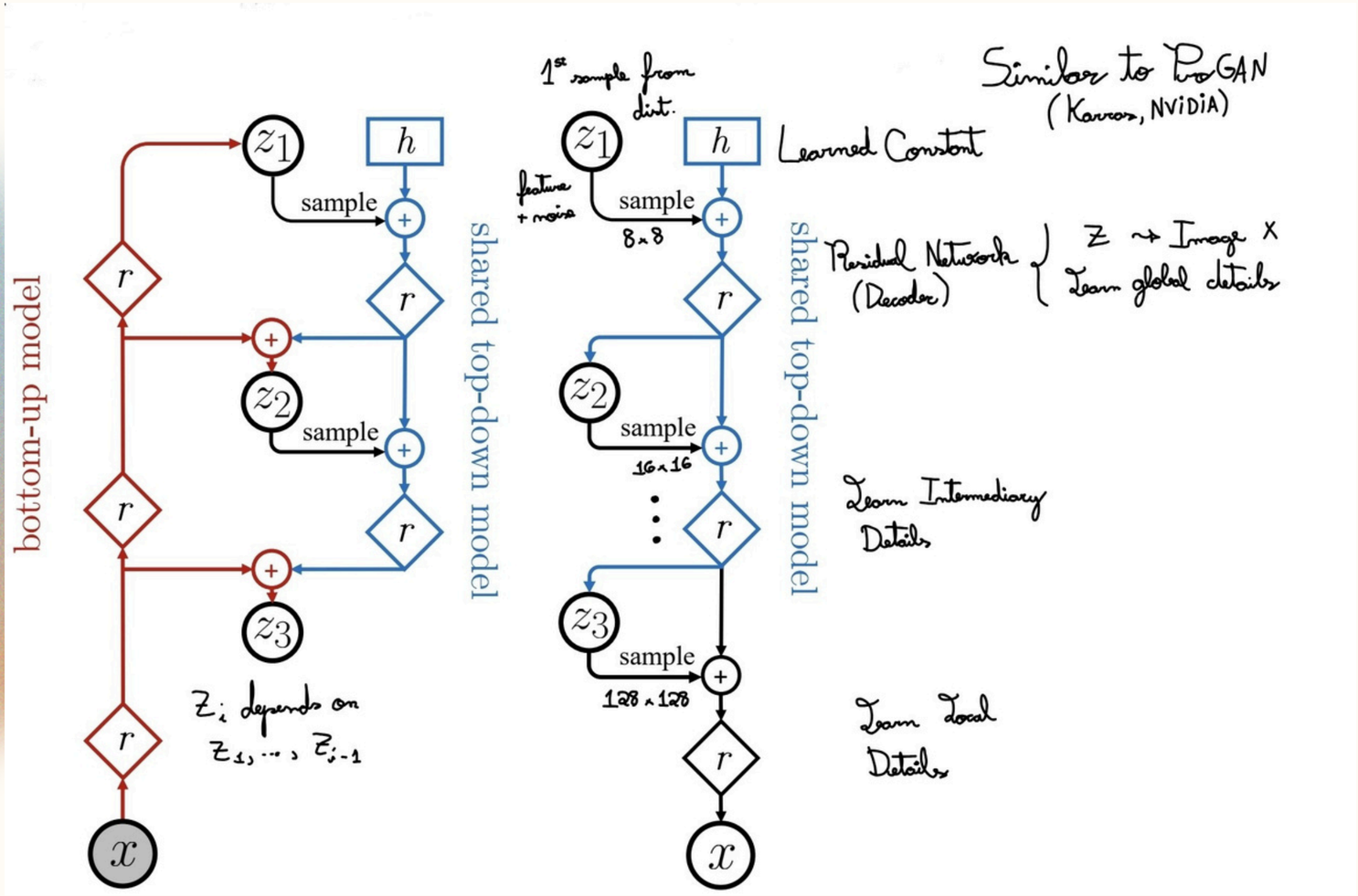
- Sample z_i from learned distribution (feature + noise).
- Add to the previous residual image.
- For the next layer, increase the z_i size (similar to ProGAN).



Architecture

Hierarchical Encoder

- z_i is generated by the combination of $x's$ encoded representation and the previous residual output from decoder block.



Main Challenges

- Designing expressive neural networks specifically for VAEs.
- Model the **long-range correlations** in data, beyond hierarchical multi-scaling approach.
- Training a large number of hierarchical groups and image sizes while maintaining training **stability**.

Residual Cells

For the Generative Module

- Improve long-range correlations by **increasing the kernel** sizes in the convolutional path.
- However, large filter come with the cost of **large parameter sizes** and **computational complexity**.

Usage of Depthwise Convolution (DC)

- DC outperform regular convolutions while keeping the number of parameters and the computational complexity orders of magnitudes smaller.
- However, DC have limited expressivity as they operate in each channel separately.

Batch Normalization

- Adjust the momentum hyperparameter of BN.
- Apply a regularization on the norm of scaling parameters in BN layers.

Residual Cells

For the Generative Module

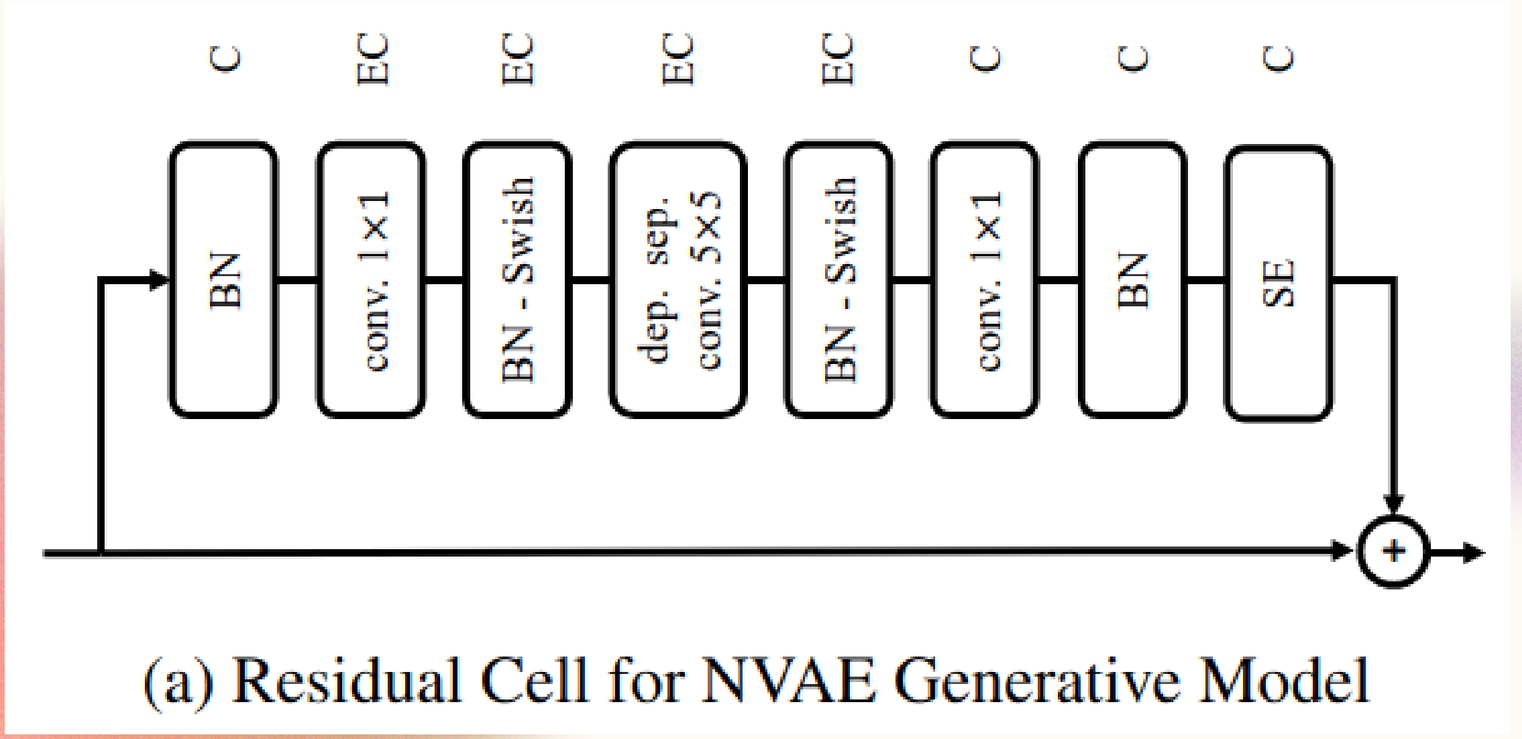
- Improve long-range correlations by **increasing the kernel** sizes in the convolutional path.
- However, large filter come with the cost of **large parameter sizes** and **computational complexity**.

Squeeze and Excitation (SE)

- Simple channel-wise gating layer that has been used widely in classification problems.

Swish Activation Fuction

$$f(u) = \frac{u}{1 + e^{-u}}$$



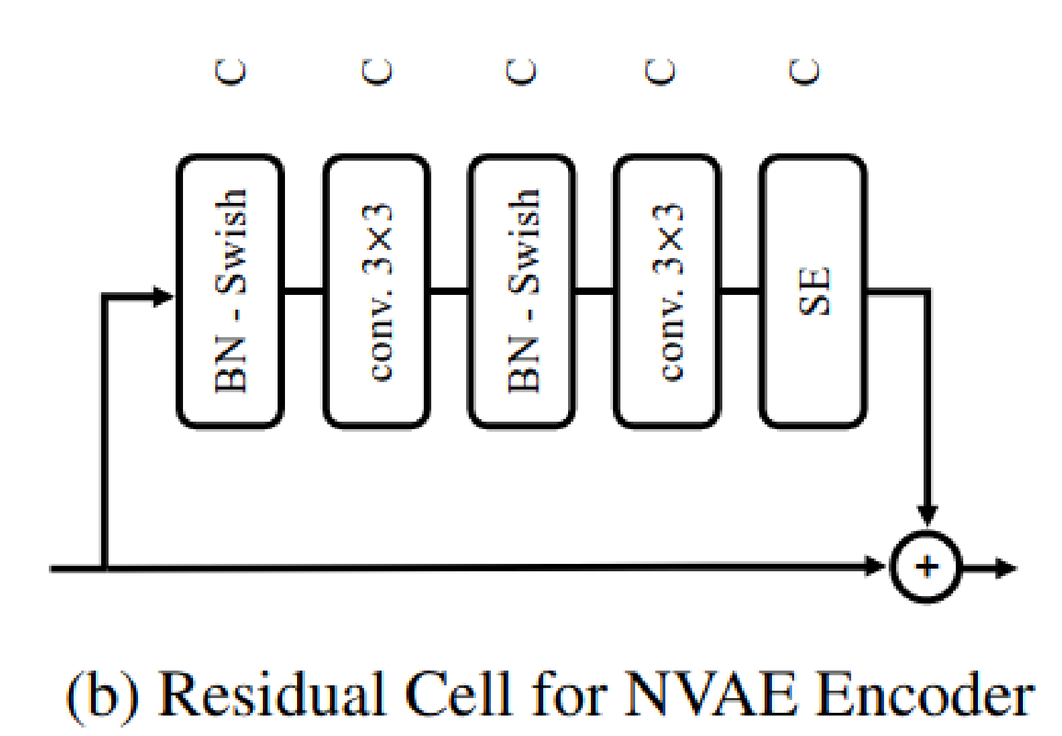
(a) Residual Cell for NVAE Generative Model

This block represents a residual NN of the decoder.

Residual Cells

For the Encoder Module

- Depthwise convolutions **didn’t improve** encoder’s performance.
- Since regular convolutions require less memory, they used a simpler architecture using BN, SE and Swish.



(b) Residual Cell for NVAE Encoder

This block represents a residual NN of the encoder.

Memory Usage

For the Generative Module

- High memory requirement imposed by the expanded features.
- These two tricks **double the training throughput** when using a larger batch size

Mixed-precision floating-points

- Defined the model in mixed-precision using the NVIDIA APEX library.
- Convolutions that can safely be cast to half-precision floats.
- Reduce GPU memory by 40%.

Backward pass cache

- In the residual cells, a copy of feature maps for each operation is stored for the backward pass.
- **Gradient check-pointing**: fusion of BN and Swish, storing only one feature map for the backward pass, instead of two.
- The additional BN computation does not change the training time significantly, but reduces 18% GPU memory.

Unbounded KL Term

Training deep hierarchical VAEs is challenging due to unstable KL optimization.

Residual Normal Distributions

- Residual distribution that parameterizes $q(z|x)$ relative to $p(x)$.
- Ensures posterior shifts consistently with the prior.
- Smooths KL gradients: easier optimization and more stable training.

Spectral Regularization (SR)

- Regularizes the Lipschitz constant of encoder layers.

$$\mathcal{L}_{SR} = \lambda \sum_{i} s^{(i)}$$

where s(i) is the largest singular value of the ith conventional layer

Results

- bits/dimension (bpd), lower is better.
- NVAE outperforms previous non-autoregressive models on most datasets.

Method	MNIST 28×28	CIFAR-10 32×32	ImageNet 32×32	CelebA 64×64	CelebA HQ 256×256	FFHQ 256×256
NVAE w/o flow	**78.01**	2.93	-	2.04	-	0.71
NVAE w/ flow	78.19	**2.91**	3.92	**2.03**	**0.70**	**0.69**
VAE Models with an Unconditional Decoder						
BIVA [36]	78.41	3.08	3.96	2.48	-	-
IAF-VAE [4]	79.10	3.11	-	-	-	-
DVAE++ [20]	78.49	3.38	-	-	-	-
Conv Draw [42]	-	3.58	4.40	-	-	-
Flow Models without any Autoregressive Components in the Generative Model						
VFlow [59]	-	2.98	-	-	-	-
ANF [60]	-	3.05	3.92	-	0.72	-
Flow++ [61]	-	3.08	**3.86**	-	-	-
Residual flow [50]	-	3.28	4.01	-	0.99	-
GLOW [62]	-	3.35	4.09	-	1.03	-
Real NVP [63]	-	3.49	4.28	3.02	-	-
VAE and Flow Models with Autoregressive Components in the Generative Model						
δ-VAE [25]	-	2.83	3.77	-	-	-
PixelVAE++ [35]	78.00	2.90	-	-	-	-
VampPrior [64]	78.45	-	-	-	-	-
MAE [65]	77.98	2.95	-	-	-	-
Lossy VAE [66]	78.53	2.95	-	-	-	-
MaCow [67]	-	3.16	-	-	0.67	-
Autoregressive Models						
SPN [68]	-	-	3.85	-	0.61	-
PixelSNAIL [34]	-	2.85	3.80	-	-	-
Image Transformer [69]	-	2.90	3.77	-	-	-
PixelCNN++ [70]	-	2.92	-	-	-	-
PixelRNN [41]	-	3.00	3.86	-	-	-
Gated PixelCNN [71]	-	3.03	3.83	-	-	-

Results

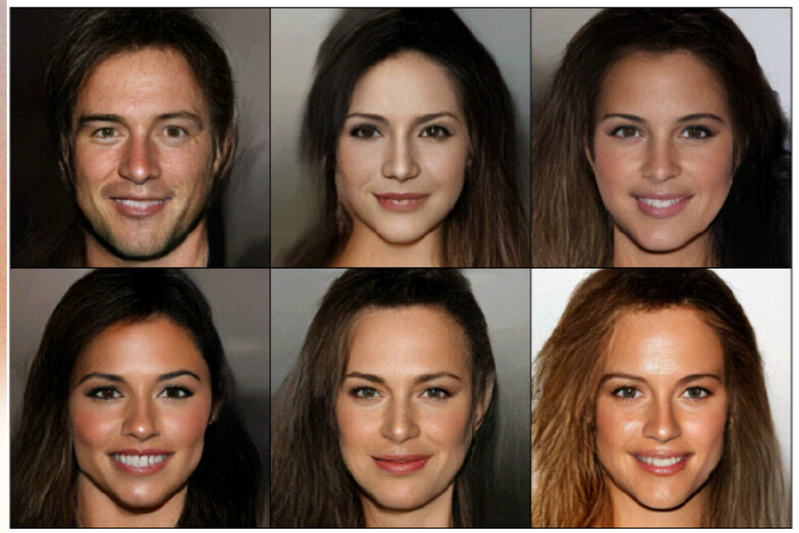
- (d) (e): NVAE generated
- (f) (g): Other AR models

- NVAE has less artifacts than other models.

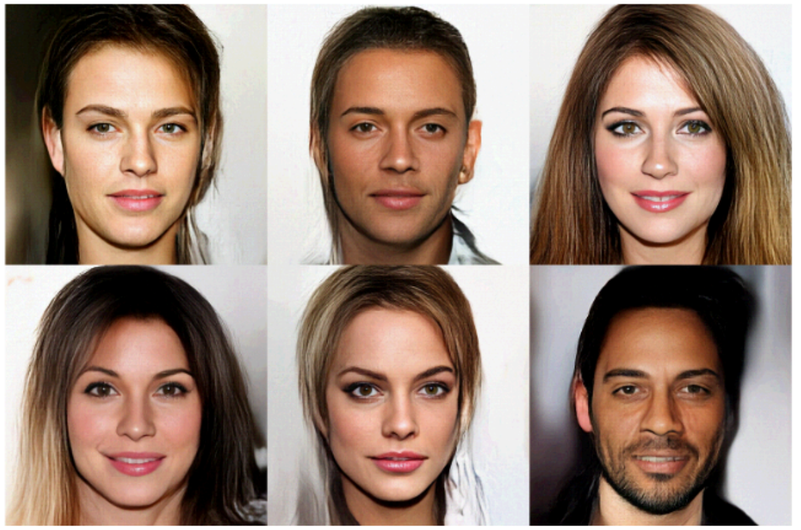
- NVAE produces high quality and diverse samples even with small temperatures.

(d) CelebA HQ ($t = 0.6$)

(e) FFHQ ($t = 0.5$)



(f) MaCow [67] trained on CelebA HQ ($t = 0.7$)



(g) Glow [62] trained on CelebA HQ ($t = 0.7$)

Results

Reconstructed images in NVAE are indistinguishable from the training images.

(a) Reconstruction results (best seen when zoomed in).

Conclusion

New SOTA for non-regressive models

- NVAE achieves SOTA results on MNIST, CIFAR-10, CelebA 64, and CelebA HQ-256.
- Provides a strong baseline on FFHQ-256.

First VAE that can produce large high-quality images

Difficult points

Understand
statistical metrics

Diagrams
interpretation

AI Perspective

Main contributions

GPT5

🚀 Main Contributions

- **1. Novel Architecture Design**
  - Deep **hierarchical VAE** with **depthwise separable convolutions**
  - **Swish + BatchNorm + Squeeze-and-Excitation** blocks for stability & expressivity
- **2. Residual Parameterization of Latents**
  - Approximate posterior defined **relative** to the prior → easier KL optimization
- **3. Spectral Regularization**
  - Controls Lipschitz constant → stabilizes training with many latent groups
- **4. Efficiency Tricks**
  - **Mixed precision** + **gradient checkpointing** → lower memory & faster training
- **5. Performance**
  - **State-of-the-art** among non-autoregressive likelihood models
  - Works on large images (CelebA-HQ, FFHQ)
  - Fast sampling (≈56 ms per 256×256 image)

AI Perspective

Limitations & Future work

⚠️ **Limitations / Open Questions**

- Training still **unstable** for very deep hierarchies
- **KL term remains unbounded** — SR only mitigates
- **BatchNorm behavior** affects sample diversity — not fully understood
- **No autoregressive prior** → may limit expressiveness
- High **computational cost** and **manual architecture tuning**

GPT5

Thank you!