

Auto regressive를 능가하는 Deep hierarchical VAE를 제작함.

그리고 UPVAE는 출처인듯, auto regressive와 빠르고 좋은 model을 만들 수 있다고 한다.

* Introduction.

Auto regressive, VAE, normalizing flow의 NLL은 model의 비교 평가에 좋다.

PixelCNN 보다, auto regressive(AT)는 높은 성능을 보였음.

AT는 natural image (latent or observed scene)의 variable 간의 결합성을 학습하여, long-term dependency를 포함된다.

VAE는 사실적으로 image가 생성되는 방식에 가깝지만,

ImageNet에서 VAE보다 AT이 더 성능이 좋다.

AT이 실제로 더 좋은 것인지, 개선된 VAE가 더 좋은지에 대해서는 많은 연구가 있다.

AT이 VAE보다 latent data의 복잡한 학습과 더 나은 때문이

복잡한 synthesis, high-dim data의 small architecture를 가능케 함.

논문에서 VAE는 최초 AT과 동일한 성능을 냈다고 한다.

↳ AT는 강제로 prior, restricted posterior를 넣기 때문이다.

↳ observed variable만 출력함.

↳ Deep hierarchy VAE라는 시설(Section 3)

Contribution.

1. 이를 것으로看他 deep hierarchy가 VAE의 성능을 높이는지 정도.

2. no layer 이상으로 학습할 수 있는 model 소개.

3. model capacity의 관계없이 depth가 성능을 향상시킬 때, VAE가 모든 benchmark에서 PixelCNN을 능가.

4. PixelCNN의 배수 더 작은 param, 수천배 빠른 synthesis, 더 큰 image로 확장

* Preliminaries

* VAE

→ An Introduction to Variational Autoencoder. 읽기.

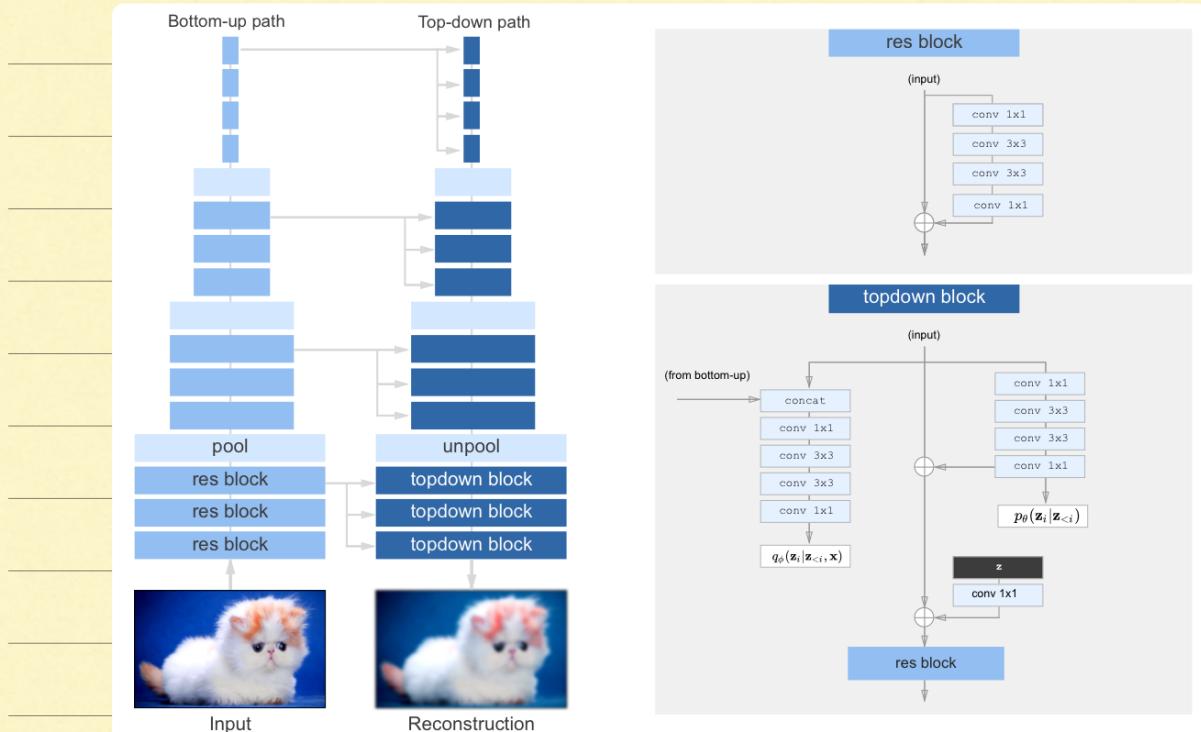


Figure 3: A diagram of our top-down VAE architecture. Residual blocks are similar to bottleneck ResNet blocks (He et al., 2016). Each convolution is preceded by the GELU nonlinearity (Hendrycks & Gimpel, 2016). $q_\phi(\cdot)$ and $p_\theta(\cdot)$ are diagonal Gaussian distributions. \mathbf{z} is sampled from $q_\phi(\cdot)$ during training, and $p_\theta(\cdot)$ when sampling. We use average pooling and nearest-neighbor upsampling for pool and unpool layers.

* Hierarchical VAE

높은 성능의 generation을 위해 복잡한 latent distribution을 요구할수록 성능이 낮아진다.

simple latent로 복잡한 latent를 만드는 간단한 방법이 hierarchical

이미지에서 모든 hierarchy layer는 feature map을 만들고.

\mathbf{z}_0 은 network의 Top part low-resolution 혹은 low variable, \mathbf{z}_N 은 bottom part high-res. high variable.

유명한 구조는 LVAE. (Top-down)

↳ prior posterior approximation을 같은 수준으로 latent를 만든다.

$$p_\theta(\mathbf{z}) = p_\theta(\mathbf{z}_0)p_\theta(\mathbf{z}_1|\mathbf{z}_0)\dots p_\theta(\mathbf{z}_N|\mathbf{z}_{<N})$$

$$q_\phi(\mathbf{z}|\mathbf{x}) = q_\phi(\mathbf{z}_0|\mathbf{x})q_\phi(\mathbf{z}_1|\mathbf{z}_0, \mathbf{x})\dots q_\phi(\mathbf{z}_N|\mathbf{z}_{<N}, \mathbf{x})$$

$D \rightarrow N$ 가능.

보통 feature를 generate하기 위해 bottom-up으로 deterministic한 data 생성.

그리고, Top-down으로 latent group을 생성한다.

↳ approxim posterior, prior, recon network $P_\theta(x|z)$ 사이에 공유하는
feature를 생성하기 위해 feed forward network 사용.

논문에서는 이 구조가 가장 효과적이며, 간단하고, 생물의 시각 과정과 유사하다고 여긴다.

x Why depth matters for hierarchical VAEs.

충분한 빌드 시간이 있는 Hierarchy VAE는 observed Variable에 대해 입력의 order를 학습할 수 있다.

그 블록이 존재한다면, 더 효과적으로 학습할 수 있다.

- Definition

N stochastic layer deep hierarchical VAE

Independent $p(x|z)$

Top-down factorized prior, approxim posterior

- Proposition.

1. $N=1$ data dim 일 때 N -layer VAE는 autoregressive model과 같다.

2. N -layer VAE는 N -dimension latent density의 보편적인 approximator.

Proposition 1의 경우 \mathbb{H} autoregressive가 더 성능이 좋은지 미리 나타낸다.

↳ dependency 푸면 깊어 deeper.

32x32 일 때, 1 step generate라면, 300Layer depth가 있어야 한다.

Proposition 1의 경우 $N=D$ 는 극단적인 경우고, 더 짧게 learnable 할 수 있다고 한다.

$\mathbf{z} \in \mathbb{R}^k, k < D$ 가 있다면, G 는 더 짧은 압축된 data를 사용할 수 있으며.

Proposition 2 k -layer VAE가 이런 variable의 prior와 posterior를 학습할 수 있다고 명시한다.

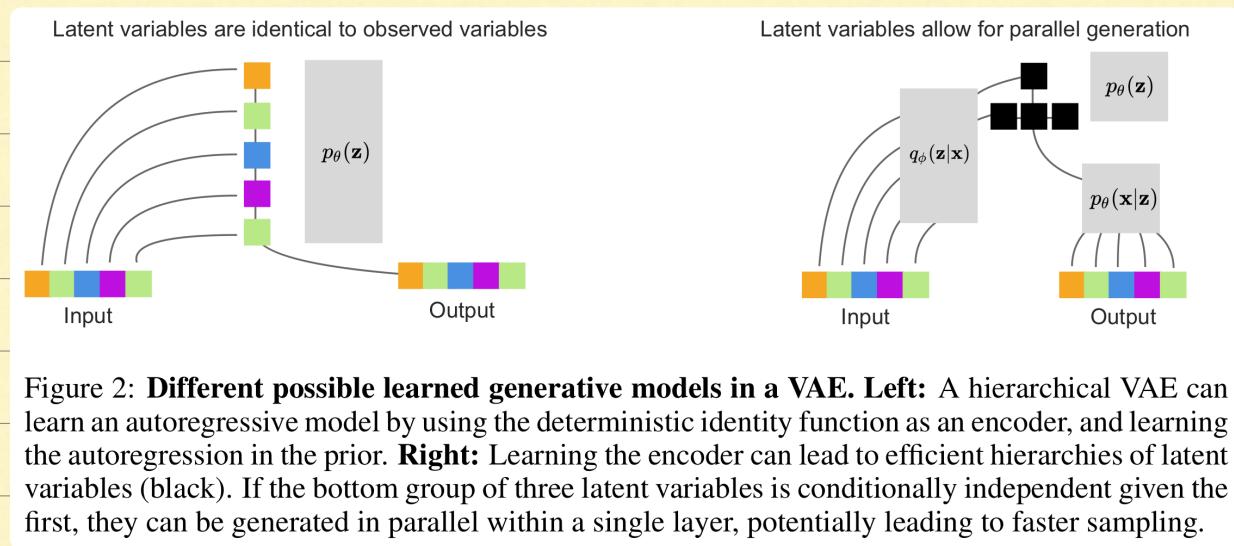


Figure 2: **Different possible learned generative models in a VAE.** **Left:** A hierarchical VAE can learn an autoregressive model by using the deterministic identity function as an encoder, and learning the autoregression in the prior. **Right:** Learning the encoder can lead to efficient hierarchies of latent variables (black). If the bottom group of three latent variables is conditionally independent given the first, they can be generated in parallel within a single layer, potentially leading to faster sampling.

이런 짧은 G 는 2가지 방식으로 나타난다.

1. model이 특정 변수가 다른 천들이 모두 independent인 것을 안다면,

model은 single layer에서 병렬적으로 변수를 생성할 수 있다. $q_\theta(\mathbf{z}_n | \mathbf{z}_{<n}, \mathbf{x}) = \prod_i q_\theta(z_n^{(i)} | \mathbf{z}_{<n}, \mathbf{x})$ 이다.

↳ image에서 공간적으로 독립적인 texture가 있다면 이런 efficient-hierarchy가 나타난다고 가정

2. model은 data의 low-dimension representation을 학습할 수 있다.

한 연구에서 \mathbb{R}^D 의 embedded는 k -dimension manifold인 low-dimension distribution에 학습될 때,

VAE는 latent의 k dim만 activate하는 것을 보였다.

이는 manifold가 D dimension이 아닌 이상, 더 작은 layer가 필요하다는 것을 알 수 있다.

추가로 data의 lowest k 를 알 수는 없지만 현재 대부분의 VAE보다 같다.

하지만, VAE는 30 layer까지 가능하지만 더 짧은 layer가 성능을 높인다고 가정한다.

* An Architecture for very deep VAEs

* Architectural components and initialization.

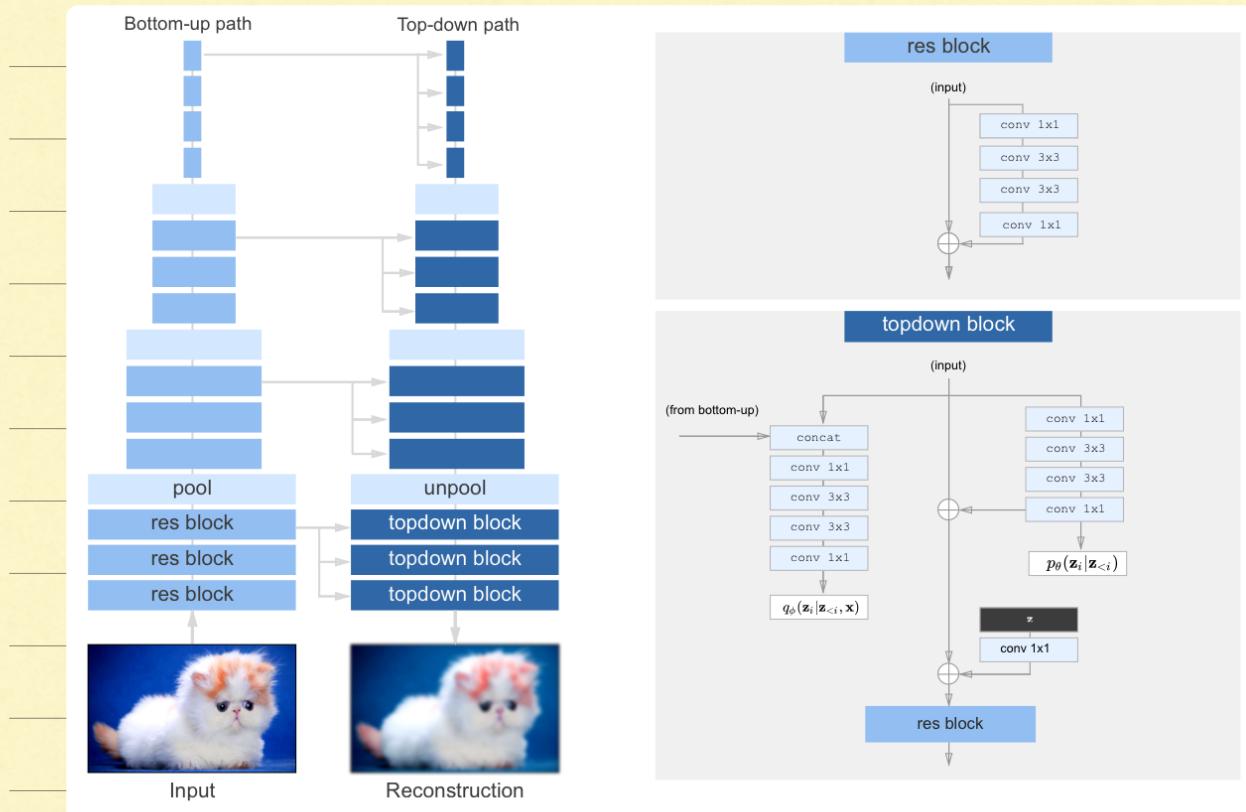


Figure 3: A diagram of our top-down VAE architecture. Residual blocks are similar to bottleneck ResNet blocks (He et al., 2016). Each convolution is preceded by the GELU nonlinearity (Hendrycks & Gimpel, 2016). $q_\phi(\cdot)$ and $p_\theta(\cdot)$ are diagonal Gaussian distributions. \mathbf{z} is sampled from $q_\phi(\cdot)$ during training, and $p_\theta(\cdot)$ when sampling. We use average pooling and nearest-neighbor upsampling for pool and unpool layers.

LVAE랑 기본적으로 비슷하지만 Resblock의 bottleneck이 없다.

↳ stochastic layer의 대체 prior와 posterior는 diag Gaussian distribution

No layer depth을 줄여 residual bottleneck block의 마지막 conv output을 $\frac{1}{M}$ 으로 scaling

↳ 안정성, 성능↑

unpool하는 nearest + resblock 사용.

↳ Transparency하는 low resolution latent의 posterior collapse 방지.

* Stabilizing training with gradient skipping

VAE는 optimization이 어렵다.

alt recon 또는 KL loss의 경우 gradient가 급격히 높아져 나쁘게 보이는 것

max gradient를 넘으면 update를 skip하는 방식으로 divergence를 막고, train을 smooth하게 만든다.

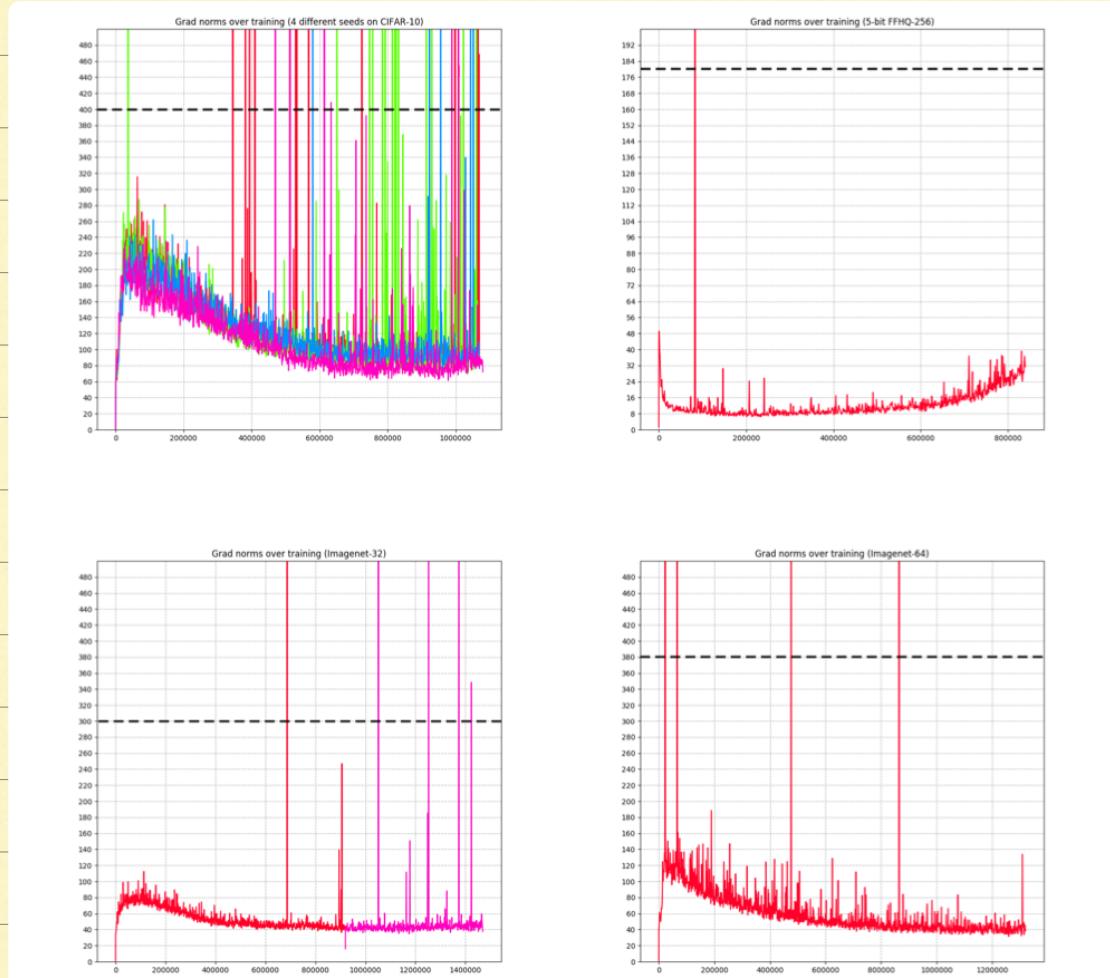


Figure 6: Effect of gradient skipping. We plot the max gradient norm encountered per 500 updates for our best models across datasets. The dashed black line indicates the “skip threshold”, or value above which the update is skipped. We choose a high threshold that affects fewer than 0.01 percent of training updates. Without this skip heuristic, networks will diverge when extreme updates are encountered. These updates can have norm as high as $1e15$.

⇒ Experiments

* Statistical depth, independent of capacity, improves performance

- Depth가 대한 성능 변화.

48 layer network에서 각각의 대로 conditioning 하는 대신,

output variable을 independent 하도록 layer를 grouping 한다.

↳ i 번째 TD의 입력이 x_i 일 때, $x_{i+1} \sim x_{i+k}$ 를 x_i 로 쌓았을 때 연속된 k개의 독립 block을 만든다

$$\hookrightarrow \text{설명 } x_{i+1} = x_i + f(\text{block}(x_i))$$

→ param 수의 영향을 고려하지 않고, stochastic depth를 확인한다.

↳ 48개의 layer의 성능과 상관관계를 보여줌.

Table 1: **Loss by network with different configurations of stochastic layers on ImageNet-32** (similar trends appear on CIFAR-10). **Left:** Networks with equal number of layers, but with lower stochastic depth as described in Section 5.1. Increasing depth up to 48 layers still shows gains, which is farther than previous work has explored. **Right:** Networks with 48 layers, but distributed at different resolutions. We find higher resolutions benefit more from layers.

Depth	Params	Test Loss	Distribution of 48 layers					Test Loss
			32x32	16x16	8x8	4x4	1x1	
3	41M	4.30						
6	41M	4.18	10	10	10	10	8	3.98
12	41M	4.06	12	12	10	8	6	3.97
24	41M	3.98	14	14	10	6	4	3.96
48	41M	3.95	16	16	10	4	2	3.95

- Scale.

Table 2: **Our main results on standard benchmark datasets.** Very deep VAEs outperform PixelCNN-based autoregressive models with fewer parameters while maintaining fast sampling. “Depth” refers to the number of stochastic layers for hierarchical VAEs (although BIVA and IAF-based networks have additional statistical dependencies). Sampling refers to the number of network evaluations per sample, and D designates the dimensionality of the data. An asterisk (*) denotes our estimate of parameters. Samples for ImageNet and CIFAR-10 are in the Appendix.

	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
VLAЕ (Chen et al., 2016)	VAE			D	≤ 2.95
IAF-VAE (Kingma et al., 2016)	VAE		12	1	≤ 3.11
Flow++ (Ho et al., 2019)	Flow	31M		1	≤ 3.08
BIVA (Maaløe et al., 2019)	VAE	103M	15	1	≤ 3.08
NVAE (Vahdat & Kautz, 2020)	VAE	131M	30	1	≤ 2.91
Very Deep VAE (ours)	VAE	39M	45	1	$\leq \boxed{2.87}$
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	≤ 3.96
NVAE	VAE	268M	28	1	≤ 3.92
Flow++	Flow	169M		1	≤ 3.86
Very Deep VAE (ours)	VAE	119M	78	1	$\leq \boxed{3.80}$
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	≤ 3.69
Very Deep VAE (ours)	VAE	125M	75	1	$\leq \boxed{3.52}$
FFHQ-256 (5 bit)					
NVAE	VAE		36	1	≤ 0.68
Very Deep VAE (ours)	VAE	115M	62	1	$\leq \boxed{0.61}$
FFHQ-1024 (8 bit)					
Very Deep VAE (ours)	VAE	115M	72	1	≤ 2.42

↳ Autoregressive 모델은 성능이 좋고, 다른 모델의 경우 params이 적다.

↳ stochastic depth) 다른 모델과의 차이점.

➤ Very deep VAEs learn an efficient hierarchical ordering

VAE의 경우 AR 만들 때 깊이가 깊을수록, parallel하게 합성할 수 있는 conditionally independent variable이

hierarchical latent를 학습할 수 있는데 이에 대한 문제가 있음.

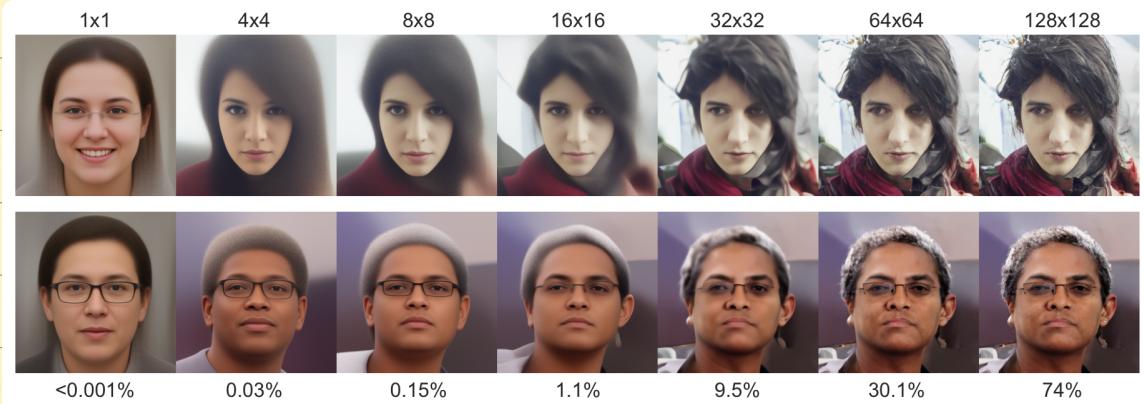


Figure 4: Cumulative percentage of latent variables at a given resolution, and reconstructions of samples on FFHQ-256. We sample latent variables from the approximate posterior until the given resolution, and sample the rest from the prior at low temperature. This shows what images are likely given a subset of latent variables. Low-resolution latents comprise a small fraction of the total latents, but encode significant portions of the global structure. This suggests deep VAEs learn efficient, hierarchical representations of the data.

low resolution latent는 전부나 latents의 1% 미만이지만 global feature를 거의 결정.

high resolution은 spatially independent 한 것으로 보여, parallel하게 생성할 수 있다.

↳ 더 나은 log likelihood, 빠른 sampling.

↳ 또한 절약은 param (30% 정도)

↳ 더 적은 global-dependency, simpler to learn

high-res init layer → 많은 수로 이득 (to the right)

↳ global feature > local feature 보다 작은 부분이다.

* Very deep VAEs are easily scaled to high dimensional data

AR은 scaling의 문제로 좋음

↳ sampling 시 접근 가능한 memory를 linear화로 즐기.

VAE는 scaling이 좋다.

↳ network의 upsampling 적용 방식을.

↳ forward 시 단계별로 만날.

Table 4: **Key hyperparameters for experiments.** We detail here the main hyperparameters used in training. FFHQ-1024 has reduced hidden size for higher resolutions; see code for details.

Parameter	CIFAR-10	ImageNet-32	ImageNet-64	FFHQ-256	FFHQ-1024
Num layers	45	78	75	62	72
Hidden size	384	512	512	512	Varies
Bottleneck size	96	128	128	128	Varies
Latent dim per layer	16	16	16	16	16
Batch size	32	256	128	32	32
Learning rate	0.0002	0.00015	0.00015	0.00015	0.00007
Optimizer	Adam	Adam	Adam	Adam	Adam
Skip threshold	400	300	380	180	500
Weight Decay	0.01	0.0	0.0	0.0	0.0
EMA rate	0.0002	0.00015	0.00015	0.00015	0.00015
Training iterations	1.1M	1.7M	1.6M	1.7M	1.7M
GPUs	2 x V100	32 x V100	32 x V100	32 x V100	32 x V100
Training time	6 days	2.5 weeks	2.5 weeks	2.5 weeks	2.5 weeks
Parameters	39M	119M	125M	115M	115M

* Related Work

같은 network가 더 성능이 좋고, new architecture, NLL의 이득에 대한 경쟁화.

BIVA, NVAE 등에서 간접되는 depth의 효용성 입증

Diff도 바운더지로 specific analytical posterior를 찾는다.

↳ VAE는 단일 forward.

Appendix A4.5 및 IAFel 소개 및 언급

* Conclusion.

VAE는 AR보다 성능이 좋아야 한다.

* Appendix A.

* Ablation of architectural components

upsampling layer et residual의 posterior collapse의 영향을 바운다.

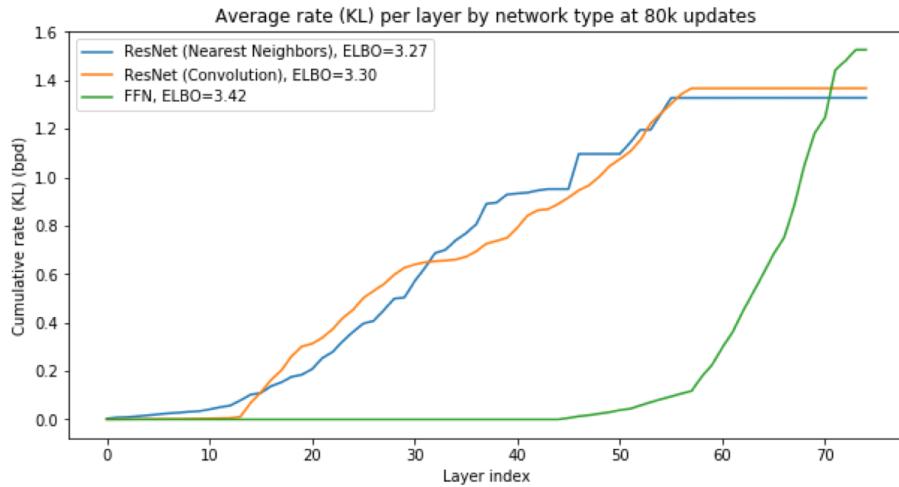


Figure 5: **Relationship between architecture and posterior collapse.** We visualize the cumulative KL divergence (or “rate”, in bits per dimension) for several different architectures across a 73 layer network on ImageNet-32. When residual connections are removed from the “res block” in the top-down path (Figure 3), the model encodes no information in the first 45 layers of the network and the loss is highest (“FFN”). When a learned convolutional upsampler is used as the “unpool” layer, the first 13 layers of the network encode no information. When nearest-neighbor upsampling is used, the first layers all encode information, and the loss is the lowest.

↳ 73 layer의 대비서, 각각 kl을 의미

↳ FFN은 이미 가장 높아 보임.

Table 3: **Effects of scaling residual initialization on very deep VAEs.** We trained networks with varying depths for 80k iterations. Scaling the last layer in the residual block by $\frac{1}{\sqrt{N}}$ results in higher losses for shallower networks, but lower losses and greater stability for deeper networks. The number of updates which are skipped because the gradient norm would destabilize the network is significantly reduced with scaling.

Depth	Without scaling		With scaling	
	Loss	Skipped Updates	Loss	Skipped Updates
15	2.50	13	2.51	0
30	2.36	41	2.38	1
45	2.31	48	2.30	0
60	2.30	76	2.29	1
75	Diverged	-	2.28	0

↳ $\frac{1}{\sqrt{N}}$ 의 unstable update를 줄여준다.

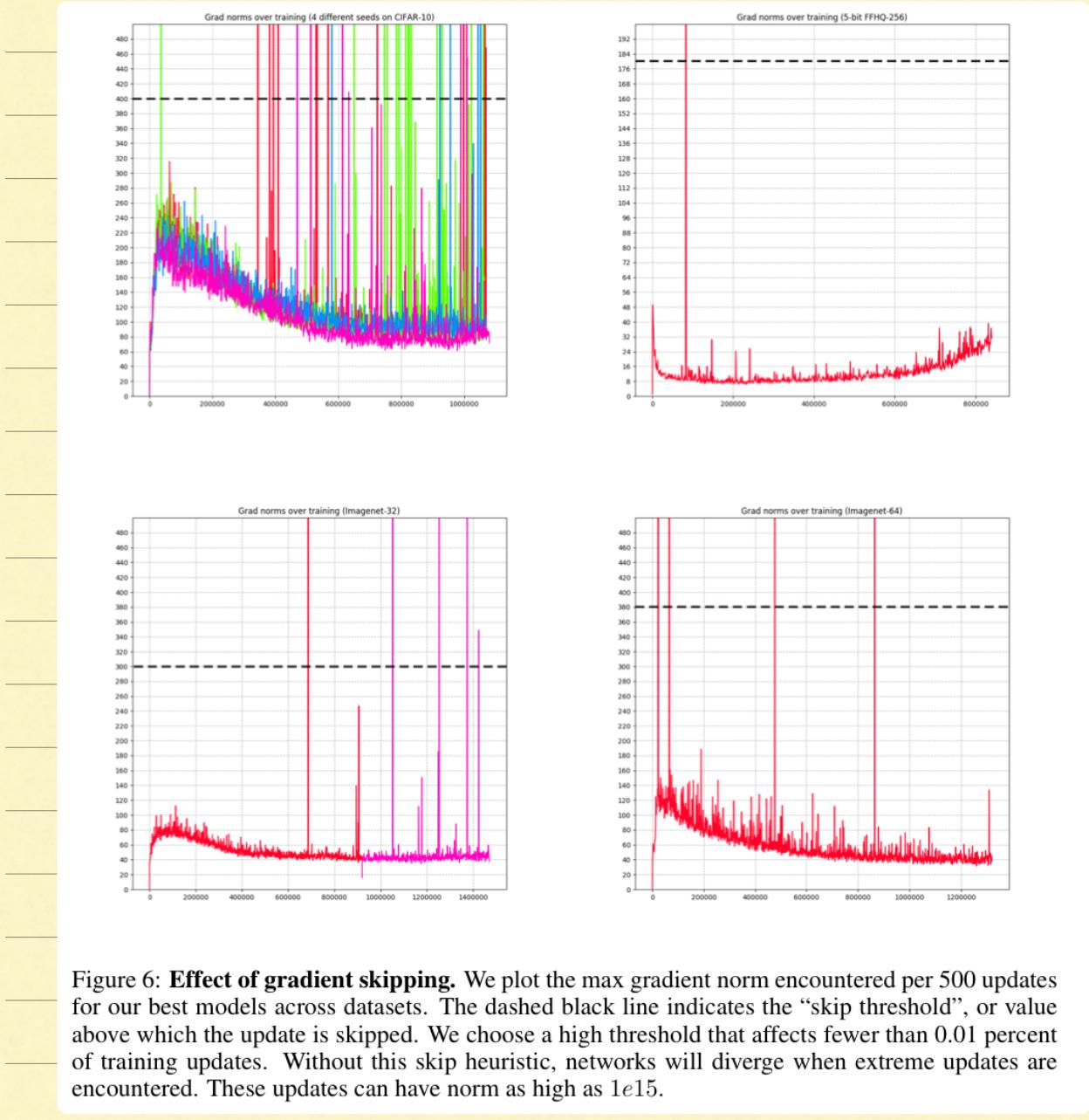


Figure 6: Effect of gradient skipping. We plot the max gradient norm encountered per 500 updates for our best models across datasets. The dashed black line indicates the “skip threshold”, or value above which the update is skipped. We choose a high threshold that affects fewer than 0.01 percent of training updates. Without this skip heuristic, networks will diverge when extreme updates are encountered. These updates can have norm as high as $1e15$.

↳ update skipping이不稳定을 예방.

↳ 적은 skip한 정도로 threshold를 조건 설정

* **Proposition 1.** N -layer VAE generalize AR model when N is the data dimension.

Observed Variable의 대체 임의의 순서를 갖는 AR이 단순히 주어진 order의 observed Variable의 경우

학습하는 approximate posterior el generator를 갖는 N -layer VAE의 경우 보여줄

generality를 잊지 않고, 각 vector latent variable \mathbf{z} : i는 initial element만 갖다고 가정, notation 단순화

$$p_{\theta}(\mathbf{z}) = p_{\theta}(z_0)p_{\theta}(z_1|z_0)\dots p_{\theta}(z_N|z_{<N}) \quad (2)$$

$$q_{\phi}(\mathbf{z}|\mathbf{x}) = q_{\phi}(z_0|\mathbf{x})q_{\phi}(z_1|z_0, \mathbf{x})\dots q_{\phi}(z_N|z_{<N}, \mathbf{x}) \quad (3)$$

in case prior, approxim posterior distribution 같음

Proof.

$$q(z_i = x_i | z_{<i}, \mathbf{x}) = 1, p(x_i = z_i | \mathbf{z}) = 1 \text{ 일때,}$$

eq 1의 ELBO는 $p(\mathbf{z}|\mathbf{x})=q(\mathbf{z}|\mathbf{x})$ 를 암시한다고 알려져 있다.

$$\log p_{\theta}(\mathbf{x}) \geq E_{\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})} \log p_{\theta}(\mathbf{x}|\mathbf{z}) - D_{KL}[q_{\phi}(\mathbf{z}|\mathbf{x}) || p_{\theta}(\mathbf{z})] \quad (1)$$

$$\log q(\mathbf{z}|\mathbf{x}) = \log p(x|\mathbf{z}) = 0 \text{ 이므로.}$$

$$\text{ELBO는 } \log P_{\theta}(\mathbf{x}) = \log P_{\theta}(\mathbf{z}) = \sum_{i=1}^N \log P_{\theta}(z_i | z_{<i}) = \sum_{i=1}^N \log P_{\theta}(x_i | x_{<i})$$

↳ observed variable의 경우, AR과 같다.

⇒ **Proposition 2.** N -layer VAEs are universal approximators of N -dimensional latent density

↳ Hierarchy VAE는 depthwise AR flow를 학습하고, 특성 조건에서는 충분한 capacity, $\alpha\eta\eta$

N -dim의 latent variable의 density를 표현할 수 있다.

Proof.

$P_\theta(z)$ 가 prior라고 했을 때, $P_\theta(z)$ 는 알려진 base density P_0 으로부터 ϵ 의 reparameterize trick으로 표현 가능

$$P_\theta(z) = P_0(\epsilon) \left| \det \frac{\partial f(\epsilon, \theta)}{\partial \epsilon} \right|, \quad f \text{는 } \epsilon \rightarrow z \text{에 대한 } NN$$

↳ ARolI, Jacobian이 lower triangle이거나 대칭이

$P_\theta(z)$ 는 $P(z)$ 로 근사될 수 있다.

같은 logic이 $q_\phi(z|x)$ 의 $p(z|x)$ 에도 적용됨.

이는 f 가 임의의 확률 밀도의 inverse CDF를 적용할 수 있을지에 달렸고,

Gaussian distribution의 실례로 VAE의 express의 제한을 든다.

↳ elementwise depth의 문제이기 때문에 신경안쓰고 future work.

↳ inverse CDF는 sampling $\eta\eta$ 의 특정 pdf를 따르는 sampling을 하기 위함.

→ A Note on IAF.

IAF는 VAE의 universal approxim posterior라는 점에서 deep VAE의 유사체다.

→ IAF의 특징

IAF의 masked AR component는 spatially stochastic dependency를 만들

↳ depth of dependency와 deep VAE의 chain inductive bias

I2I로 IAF는 각 component의 동일한 computation, param.

↳ deep VAE는 stepwise의 차례, local, global의 구조.

↳ 이미지에서 실제로 다른 동작, IAF는 deep VAE의 image decomposition을 가능하게 하는 매개변수

→ IAF는 서로 상호 보완적

✗ A Note on learning hierarchical features

Gibbs sampling의 data recovery에 충분하여, hierarchical VAE가 활용성이 있다는 연구도 있겠지만

Gibbs는 느리고, hierarchical 더 간단.