long-range sequence에는 transformer가 SOTA다.

CNN의 inductive bias는 model을 효과적이지만, long-range interactive에는 비현실적인 compute cost가 든다.

∴ CNN과 transformer의 장점 결합하고 싶음

↳ CNN으로 context를, transformer로 구성요소를 뽑음.

Conditional에도 동작을 잘한다.

* Introduction.

Transformer는 NLP에서 SOTA고, 다른 분야에도 적용되고 있다.

CNN과 다르게 내장된 inductive bias가 없기 때문에 입력의 complex한 관계를 학습할 수 있다.

그러나, 이때문에 모든 pair간 계산이 필요해서 quadratic cost가 든다.

∴ 고해상도에는 시간과 에너지가 많이 든다.

Transformer가 conv의 구조를 학습하는 경향이 있다.

→ 매 epoch마다 image의 구조를 scratch부터 다시 배우는가?

→ 아니면 transformer의 유연성은 유지하며 image의 inductive bias를 효율적으로 encoding 하는가?

저자는 low-level image structure에서 CNN 등은 효과적이지만, structural assumption에는 아니라고 한다.

게다가 CNN은 locality inductive bias 뿐만아니라, spatial invariance도 가지고 있다. (weight sharing)

따라서 conv는 문맥상 image를 얻기 위한 codebook을 학습하고 장거리 분포로 (시각적인 상호작용) 학습하기 위해 transformer가 학습된다.

또한 디테일 local part간 low-level distribution을 capture할 필요가 없도록 D를 활용한다.

그리고 conditional도 가능하다.

* Related Work

- The Transformer Family

Transformer는 attention으로 input 간의 상관없이 interactive 하다.

NLP에서는 token으로 input이 주어지고, image나 audio도 비슷한 signal로 주어진다.

Transformer는 attention과 fc로 구성된다.

Attention은 Q, k, V로 구성된다.

$Q \in R^{N \times d_k}$, $k \in R^{N \times d_k}$, $V \in R^{N \times d_v}$

$\text{Attn}(Q, K, V) = \text{softmax}\left(\frac{Qk^t}{\sqrt{d_k}}\right)V$

Attention은 모든 input seq 끼리 계산하기 때문에 quadratic cost가 든다.

↳ long-interactive에는 좋지만 속도에 대해 trade-off가 있다.

이런 문제를 해결하기 위해서 attention의 receptive field를 제한하거나, pixel independent의 assumption을 넣는다.

- Convolutional approach.

CNN의 경우 image size에 대해서는 compute cost가 linear하고, kernel에는 quadratic 하지만 size가 상수로 가짐.

Receptive field는 compute의 효율적이지만 계속해 너무 강함

low-res에서는 transformer가 더 강한 성능을 보임.

∴ 두 개를 섞어 씀.

- Two-stage approach.

먼저 image를 encoding 하고, encoding에 대한 probability model을 학습한다.

VAE를 사용하여, data의 representation을 학습하고, 다시 VAE의 분포를 학습하는 것의 이점을 보여준 논문이 있다.

2-stage에서 conditional normalizing flow는 unconditional normalizing flow와 비슷한 성능을 보인다.

GAN의 훈련 효율성을 높이기 위해, AE의 representation과

학습된 G에 대해 image로 decoding되는 low-res의 wavelet coefficient에 대해 GAN을 학습한다.

VQVAE에서는 image의 discrete한 representation을 학습하기 위해 conv 층에 활용했다.

계층적인 학습으로 VQVAE를 발전시킨 model도 있다.

하지만 long-range에 대해 문제가 있는 conv의 예전의 단점이다.

32x32 image로 encoding하여 학습하는 것도 cost가 부담되기 때문에, VQVAE도 (32x32)로 encoding 한다.

가능한한 작은 pixel에 대해 spatially invariant하도록 작은 receptive field를 갖는 shallow VQVAE를 썼다.

└ 이때 encoding에서 compute cost를 줄이기 위해 큰 해상도를 encode했으므로 decoder에서는 fine feature를

잘 잡기 위해 작은 receptive field를 쓴다.

하지만 저자는 context를 보는 encoder가 transformer로 학습되면,

high-res를 합성할 수 있다고 한다.

* Approach.

Quadratic한 compute cost 때문에 Transformer를 활용한 vision은 high-res를 다루지 못했다.

고해상도 이미지는 전체의 맥락을 이해해야한다. 따라서 pixel 단위보다 인지적으로 의미는 codebook을 사용한다.

Codebook을 사용해 composition의 length를 줄일 수 있고, transformer로 global한 interactive를 효율적으로 학습할 수 있다.

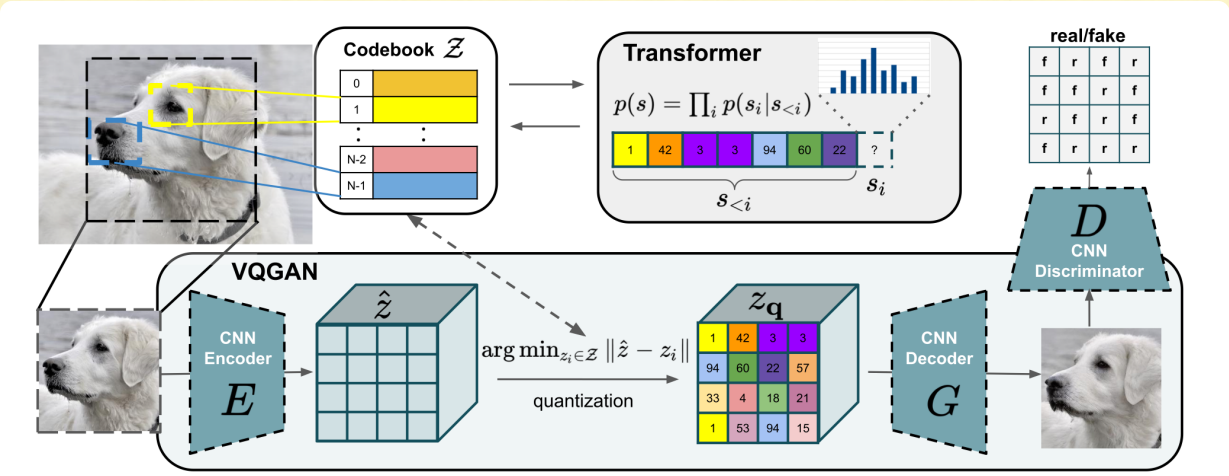


Figure 2. Our approach uses a convolutional *VQGAN* to learn a codebook of context-rich visual parts, whose composition is subsequently modeled with an autoregressive transformer architecture. A discrete codebook provides the interface between these architectures and a patch-based discriminator enables strong compression while retaining high perceptual quality. This method introduces the efficiency of convolutional approaches to transformer based high resolution image synthesis.

- Learning a Effective codebook of Image Constituents for Use in Transformer.

Transformer를 사용하려면 sequence를 input으로 받아야한다.

pixel sequence를 복잡하게 받는 대신, codebook을 받는다. $z_q \in \mathbb{R}^{h \times w \times n_z}$

⇒ VQVAE 방식

먼저, E와 G를 학습시킨다.

↳ E로 z를 만들고, z_q로부터 input과 비교할 $\hat{x} = G(z_q)$를 생성한다.

$$z_q = q(\hat{z}) = \left(\underset{z_k \in \mathcal{Z}}{\arg\min} \|\hat{z}_{ij} - z_k\| \right) \in \mathbb{R}^{h \times w \times n_z}$$

$$\hat{x} = G(z_q) = G(q(E(x)))$$

미분 불가능한 q의 경우 backpropagation은 straight-through gradient estimate로 근사한다.

↳ dec에서 enc로 gradient copy.

$$\mathcal{L}_{VQ}(E,G,\mathcal{Z}) = \|x-\hat{x}\|^2 + \|sg[E(x)]-z_q\|_2^2 + \beta\|sg[z_q]-E(x)\|_2^2$$

$\mathcal{L}_{recon}$ / stop-gradient. / commitment loss

* Learning a perceptual rich codebook

latent의 구성으로 image representation 분포에 transformer를 사용하려면, codebook 훈련이 잘되어야 한다.

따라서 VQVAE에 DeL perceptual loss를 쓴다.

$$\mathcal{L}_{GAN}(\{E,G,\mathcal{Z}\},D) = [\log D(x) + \log(1-D(x))]$$

$Q = \{E,G,\mathcal{Z}\}$ 에 대한 optimal compression을 찾기위해

$$Q^* = \arg\min_{E,G,\mathcal{Z}} \max_{D} \mathbb{E}_{x\sim p(x)}\left[\mathcal{L}_{VQ}(E,G,\mathcal{Z}) + \lambda\mathcal{L}_{GAN}(\{E,G,\mathcal{Z}\},D)\right]$$

$$\downarrow$$

$$\lambda = \frac{\nabla_{G_L}[\mathcal{L}_{rec}]}{\nabla_{G_L}[\mathcal{L}_{GAN}] + \delta}$$

G_L: decoder last layer에 대한 input의 gradient.

모든 곳의 context를 위해 가장 low-res에 대해서만 attention 진행.

- Learning the Composition of Images with Transformer.

* Latent Transformer.

E,G가 훈련되면, 이제 codebook으로 image를 만들 수 있다.

정확히는, 양자화된 image x는 $z_q = q(E(x))$로 주어지고.

$s_{ij} = k$ such that $(z_q)_{ij} = z_k$

해당되는 codebook의 mapping 후 decoder로 복원한다. $\hat{x} = G(z_q)$

index s가 주어지면 Transformer는 다음 가능한 index를 예측한다.

$p(s_i | s_{<i})$ 에 대해

$p(s) = \prod_i p(s_i | s_{<i})$

$\mathcal{L}_{Tr} = E_{x \sim p(x)}[-\log p(s)]$

- Conditional Synthesis

Condition을 주어서 image 생성하기.

$p(s|c) = \prod_i p(s_i | s_{<i}, c)$

condition r은 codebook과 같이 새로 학습되며, 단순히 s 앞에 붙여 구현합니다.

$p(s_i | s_{<i}, r)$

- Generating high-resolution Image.

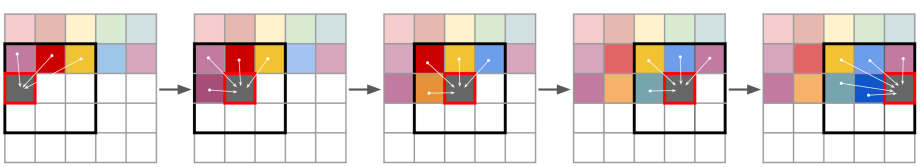


Figure 3. Sliding attention window.

Sequence h·w 의 limit를 준다.

mega-pixel을 gen하려면, patch로 잘라내어, s를 최대로 한 후 생성한다.

이때도 fig3 과 같이 sliding window로 동작한다.

VQGAN은 dataset이 spatial invariant하고, spatial conditioning이 가능하다면 model은 매우 faithful하다.

실제로 엄격한 조건이 아니고, uncondition에도 coco-GAN 처럼 condition을 넣을 수 있다.

* Experiments.

Conv와 Transformer 이점을 섞는 방식의 능력 평가.

codebook의 영향을 조사한다.

$|Z|=1024$이고, seq-len : 16x16으로 학습

- Transformer in the Latent Space.

Transformer는 image의 autoregressive에 큰 발전, but low-res 나 shallow한 pixel에는 제약이 있었음.

각 (un-) conditional에서 Tf와 conv 접근의 차이를 조사함

Tf와 pixelSnail의 조건을 맞춰 평가.

- Result

Negative Log-Likelihood (NLL)			
Data / # params	Transformer *P-SNAIL steps*	Transformer *P-SNAIL time*	PixelSNAIL *fixed time*
RIN / 85M	**4.78**	4.84	4.96
LSUN-CT / 310M	**4.63**	4.69	4.89
IN / 310M	**4.78**	4.83	4.96
D-RIN / 180 M	**4.70**	4.78	4.88
S-FLCKR / 310 M	**4.49**	4.57	4.64

D-RIN → Depth condition

S-FLCKR → semantic condition

Table 1. Comparing Transformer and PixelSNAIL architectures across different datasets and model sizes. For all settings, transformers outperform the state-of-the-art model from the PixelCNN family, PixelSNAIL in terms of NLL. This holds both when comparing NLL at fixed times (PixelSNAIL trains roughly 2 times faster) and when trained for a fixed number of steps. See Sec. 4.1 for the abbreviations.

→ Tf가 우수함.

- Unified Model for Image Synthesis Tasks.

Transformer 다양성과 일반성이 좋다.

256x256 img, 16x16 latent size에서 conditional synthesis 에 대해 실험

i) Semantic Image Synthesis

ii) Structure-to-image.

iii) Pose-guided Synthesis

iv) Stochastic superresolution.

v) Class-conditional image Synthesis

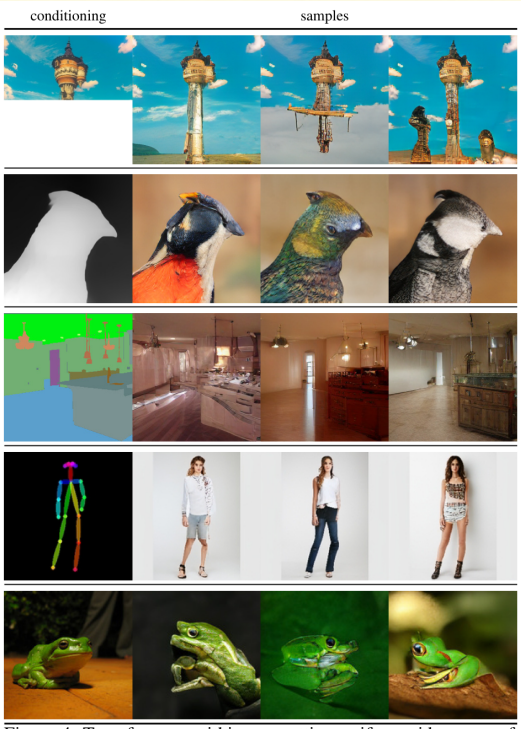


Figure 4. Transformers within our setting unify a wide range of image synthesis tasks. We show 256 × 256 synthesis results across different conditioning inputs and datasets, all obtained with the same approach to exploit inductive biases of effective CNN based *VQGAN* architectures in combination with the expressivity of transformer architectures. Top row: Completions from unconditional training on ImageNet. 2nd row: Depth-to-Image on RIN. 3rd row: Semantically guided synthesis on ADE20K. 4th row: Pose-guided person generation on DeepFashion. Bottom row: Class-conditional samples on RIN.

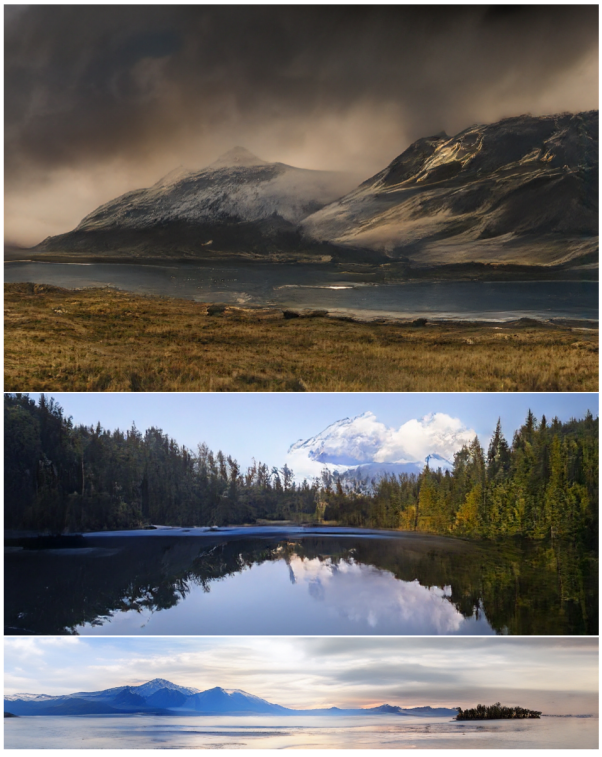


Figure 5. Samples generated from semantic layouts on S-FLCKR. Sizes from top-to-bottom: 1280 × 832, 1024 × 416 and 1280 × 240 pixels. Best viewed zoomed in. A larger visualization can be found in the appendix, see Fig 29.

예시들은 같은 방법으로 생성됐고,

Tf의 이점으로 task에 유연하지만 seq가 짧다.

따라서 별도 mechanism으로 생각

Figure 6. Applying the sliding attention window approach (Fig. 3) to various conditional image synthesis tasks. Top: Depth-to-image on RIN, 2nd row: Stochastic superresolution on IN, 3rd and 4th row: Semantic synthesis on S-FLCKR, bottom: Edge-guided synthesis on IN. The resulting images vary between 368 × 496 and 1024 × 576, hence they are best viewed zoomed in.

- High-resolution Synthesis

Sliding Window는 256×256 이미지를 기본으로

하지만 임의의 size와 ratio에도 작동할 수 있다.

Dataset의 공간적으로 벗어 나아야한다.

256의 경우 $m=4$ (size $h = H/2^m$), block = 16×16의

S-FLCKR의 경우 $m=5$,

condition을 같이 Tf의 계열

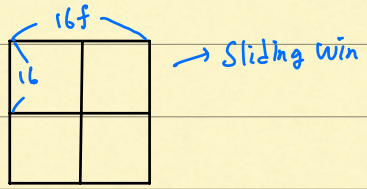
- Building Context-Rich Vocabularies

context-rich vocab의 효율을 측정하기 위해 m은 변하지만 Tf는 고정된 실험을 진행한다.

└ encoding 되는 context의 양 조절

image $H \times W$는 $H/f \times W/f$ 사이즈의 zp로 된다. (content size.)

Tf에도 16×16으로 훈련시키되, 16보다 crop해서 사용



- Results

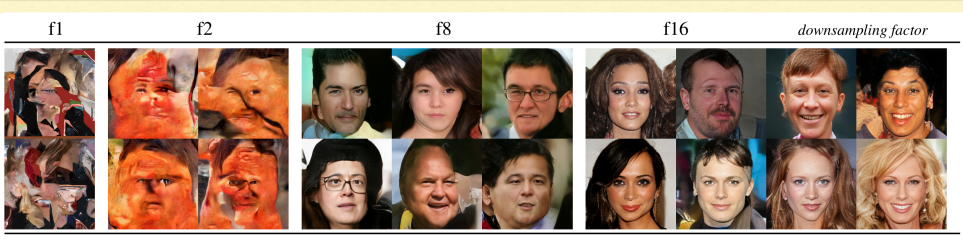


Figure 7. Evaluating the importance of effective codebook for HQ-Faces (CelebA-HQ and FFHQ) for a fixed sequence length $|s| = 16 \cdot 16 = 256$. Globally consistent structures can only be modeled with a context-rich vocabulary (right). All samples are generated with temperature $t = 1.0$ and top-k sampling with $k = 100$. Last row reports the speedup over the f1 baseline which operates directly on pixels and takes 7258 seconds to produce a sample on a NVIDIA GeForce GTX Titan X.

확실히 receptive field가 작으면 안된다.

f=8의 경우 몰려서 생기고, f=16일때 (full receptive) 등식이 잘 된다.

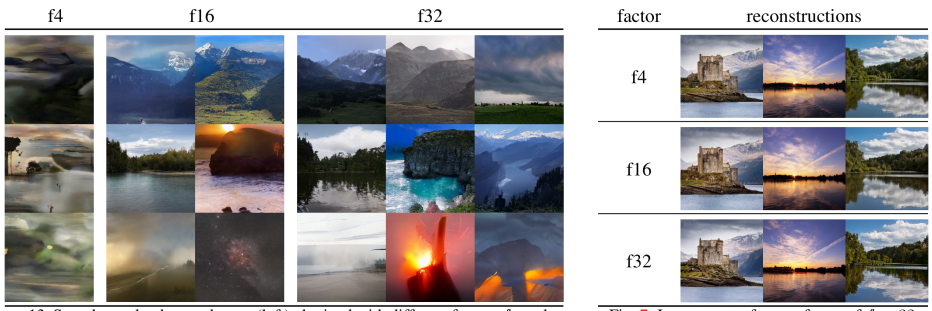


Figure 13. Samples on landscape dataset (left) obtained with different factors f, analogous to Fig. 7. In contrast to faces, a factor of $f = 32$ still allows for faithful reconstructions (right). See also Sec. C.

S-FLCKR에서도 강은 결과.

정량적인 비교를위해 image에서 바로 학습하는 것과, codebook에서 학습하는것 사이에 차이를 봤다.

↳ IS에서 정확하고, FID에서 비슷

-Benchmarking Image Synthesis Results.

각 방법별 비교.

Dataset	**ours**	SPADE [53]	Pix2PixHD (+aug) [75]	CRN [9]
COCO-Stuff	22.4	22.6/23.9(*)	111.5 (54.2)	70.4
ADE20K	35.5	33.9/35.7(*)	81.8 (41.5)	73.3

Table 2. FID score comparison for semantic image synthesis (256×256 pixels). (*): Recalculated with our evaluation protocol based on [50] on the validation splits of each dataset.

⇒ semantic synthesis

CelebA-HQ 256×256		FFHQ 256×256	
Method	FID ↓	Method	FID ↓
GLOW [37]	69.0	VDVAE ($t = 0.7$) [11]	38.8
NVAE [69]	40.3	VDVAE ($t = 1.0$)	33.5
PIONEER (B.) [23]	39.2 (25.3)	VDVAE ($t = 0.8$)	29.8
NCPVAE [1]	24.8	VDVAE ($t = 0.9$)	28.5
VAEBM [77]	20.4	*VQGAN*+P.SNAIL	21.9
Style ALAE [56]	19.2	BigGAN	12.4
DC-VAE [54]	15.8	**ours** (k=300)	9.6
ours (k=400)	10.2	U-Net GAN (+aug) [66]	10.9 (7.6)
PGGAN [31]	8.0	StyleGAN2 (+aug) [34]	3.8 (3.6)

Table 3. FID score comparison for face image synthesis. CelebA-HQ results reproduced from [1, 54, 77, 24], FFHQ from [66, 32].

⇒ Unconditional face Synthesis

⇒ 특히된 method에도 비빌 수 있음.

ㄱ Transformer top-k sampling을 수행함.

- Class-Conditional Synthesis on ImageNet.

VQGAN은 f=16, 코드=16384 (128x128)로 학습됐다.

D>1 VQGAN을 더 사실적으로 만든다.

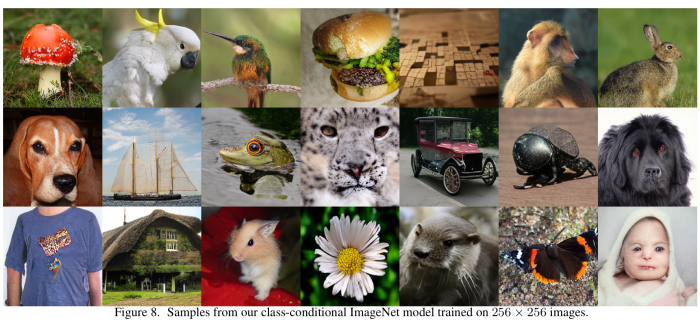


Figure 8. Samples from our class-conditional ImageNet model trained on 256 × 256 images.

Model	acceptance rate	FID	IS
mixed k, $p = 1.0$	1.0	17.04	70.6 ± 1.8
$k = 973$, $p = 1.0$	1.0	29.20	47.3 ± 1.3
$k = 250$, $p = 1.0$	1.0	15.98	78.6 ± 1.1
$k = 973$, $p = 0.88$	1.0	15.78	74.3 ± 1.8
$k = 600$, $p = 1.0$	0.05	5.20	280.3 ± 5.5
mixed k, $p = 1.0$	0.5	10.26	125.5 ± 2.4
mixed k, $p = 1.0$	0.25	7.35	188.6 ± 3.3
mixed k, $p = 1.0$	0.05	5.88	304.8 ± 3.6
mixed k, $p = 1.0$	0.005	6.59	402.7 ± 2.9
DCTransformer [48]	1.0	36.5	n/a
VQVAE-2 [61]	1.0	∼31	∼45
VQVAE-2	n/a	∼10	∼330
BigGAN [4]	1.0	7.53	168.6 ± 2.5
BigGAN-deep	1.0	6.84	203.6 ± 2.6
IDDPM [49]	1.0	12.3	n/a
ADM-G, no guid. [15]	1.0	10.94	100.98
ADM-G, 1.0 guid.	1.0	4.59	186.7
ADM-G, 10.0 guid.	1.0	9.11	283.92
val. data	1.0	1.62	234.0 ± 3.9

Table 4. FID score comparison for class-conditional synthesis on 256 × 256 ImageNet, evaluated between 50k samples and the training split. Classifier-based rejection sampling as in VQVAE-2 uses a ResNet-101 [22] classifier. BigGAN(-deep) evaluated via https://tfhub.dev/deepmind truncated at 1.0. "Mixed" k refers to samples generated with different top-k values, here $k \in \{100, 200, 250, 300, 350, 400, 500, 600, 800, 973\}$.

다른 autoregressive보다 성능이 좋다

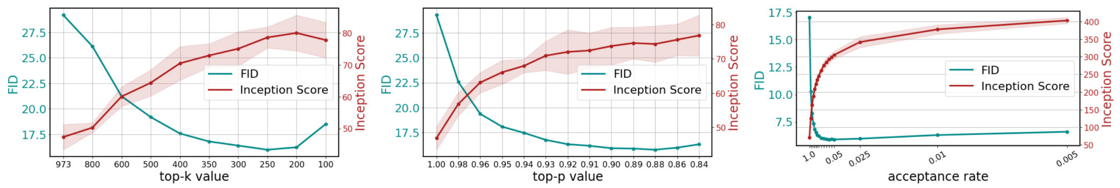


Figure 9. FID and Inception Score as a function of top-k, nucleus and rejection filtering.

- How good is VQGAN

GAN이 아닌 경우 (D가 없음) 성능비교를 위해 실험함.

Model	Codebook Size	dim $\mathcal{Z}$	FID/val	FID/train
VQVAE-2	64×64 & 32×32	512	n/a	~ 10
DALL-E [59]	32×32	8192	32.01	33.88
VQGAN	16×16	1024	7.94	10.54
VQGAN	16×16	16384	4.98	7.41
*VQGAN**	32×32	8192	1.49	3.24
VQGAN	64×64 & 32×32	512	1.45	2.78

Table 5. FID on ImageNet between reconstructed validation split and original validation (FID/val) and training (FID/train) splits. *trained with Gumbel-Softmax reparameterization as in [59, 29].

⇒ 역시 Codebook size나 크기가 클수록 성능↑

Figure 12. Comparing reconstruction capabilities between VQVAEs and *VQGAN*s. Numbers in parentheses denote compression factor and codebook size. With the same compression factor and codebook size, *VQGAN*s produce more realistic reconstructions compared to blurry reconstructions of VQVAEs. This enables increased compression rates for *VQGAN* while retaining realistic reconstructions. See Sec. C.

* Conclusion.

Tf를 사용해도 고해상도를 얻으며, quadratic cost를 해결

CNN과 Tf의 장점을 합쳐고 general한 방식이라는 것이 의미가 있다.

×Appendix

A. Changelog (이전 버전과의 차이점)

B. Implementation Details.

-Hyperparam

Experiment	n_{layer}	# params $[M]$	n_z	$\lvert\mathcal{Z}\rvert$	dropout	length(s)	n_e	m
RIN	12	85	64	768	0.0	512	1024	4
c-RIN	18	128	64	768	0.0	257	768	4
D-RINv1	14	180	256	1024	0.0	512	768	4
D-RINv2	24	307	256	1024	0.0	512	1024	4
IN	24	307	256	1024	0.0	256	1024	4
c-IN	24	307	256	1024	0.0	257	1024	4
c-IN (big)	48	1400	256	16384	0.0	257	1536	4
IN-Edges	24	307	256	1024	0.0	512	1024	3
IN-SR	12	153	256	1024	0.0	512	1024	3
S-FLCKR, $f = 4$	24	307	256	1024	0.0	512	1024	2
S-FLCKR, $f = 16$	24	307	256	1024	0.0	512	1024	4
S-FLCKR, $f = 32$	24	307	256	1024	0.0	512	1024	5
(FacesHQ, $f = 1)^*$	24	307	–	512	0.0	512	1024	–
FacesHQ, $f = 2$	24	307	256	1024	0.0	512	1024	1
FacesHQ, $f = 4$	24	307	256	1024	0.0	512	1024	2
FacesHQ, $f = 8$	24	307	256	1024	0.0	512	1024	3
FacesHQ**, $f = 16$	24	307	256	1024	0.0	512	1024	4
FFHQ**, $f = 16$	28	355	256	1024	0.0	256	1024	4
CelebA-HQ**, $f = 16$	28	355	256	1024	0.0	256	1024	4
FFHQ (big)	24	801	256	1024	0.0	256	1664	4
CelebA-HQ (big)	24	801	256	1024	0.0	256	1664	4
COCO-Stuff	32	651	256	8192	0.0	512	1280	4
ADE20K	28	405	256	4096	0.1	512	1024	4
DeepFashion	18	129	256	1024	0.0	340	768	4
LSUN-CT	24	307	256	1024	0.0	256	1024	4
CIFAR-10	24	307	256	1024	0.0	256	1024	1

Table 8. Hyperparameters. For every experiment, we set the number of attention heads in the transformer to $n_h = 16$. n_{layer} denotes the number of transformer blocks, # params the number of transformer parameters, n_z the dimensionality of codebook entries, $|\mathcal{Z}|$ the number of codebook entries, dropout the dropout rate for training the transformer, length(s) the total length of the sequence, n_e the embedding dimensionality and m the number of downsampling steps in the *VQGAN*. D-RINv1 is the experiment which compares to Pixel-SNAIL in Sec. 4.1. Note that the experiment (FacesHQ, $f = 1)^*$ does not use a learned *VQGAN* but a fixed k-means clustering algorithm as in [8] with $K = 512$ centroids. A prefix "c" refers to a class-conditional model. The models marked with a '$**$' are trained on the same *VQGAN*.

-Architecture (VQGAN)

Encoder	Decoder
$x \in \mathbb{R}^{H\times W\times C}$	$z_{\mathbf{q}} \in \mathbb{R}^{h\times w\times n_z}$
Conv2D $\to \mathbb{R}^{H\times W\times C'}$	Conv2D $\to \mathbb{R}^{h\times w\times C''}$
$m\times$ { Residual Block, Downsample Block} $\to \mathbb{R}^{h\times w\times C''}$	Residual Block $\to \mathbb{R}^{h\times w\times C''}$
Residual Block $\to \mathbb{R}^{h\times w\times C''}$	Non-Local Block $\to \mathbb{R}^{h\times w\times C''}$
Non-Local Block $\to \mathbb{R}^{h\times w\times C''}$	Residual Block $\to \mathbb{R}^{h\times w\times C''}$
Residual Block $\to \mathbb{R}^{h\times w\times C''}$	$m\times$ { Residual Block, Upsample Block} $\to \mathbb{R}^{H\times W\times C'}$
GroupNorm, Swish, Conv2D $\to \mathbb{R}^{h\times w\times n_z}$	GroupNorm, Swish, Conv2D $\to \mathbb{R}^{H\times W\times C}$

Table 7. High-level architecture of the encoder and decoder of our *VQGAN*. The design of the networks follows the architecture presented in [25] with no skip-connections. For the discriminator, we use a patch-based model as in [28]. Note that $h = \frac{H}{2^m}$, $w = \frac{W}{2^m}$ and $f = 2^m$.

Transformer의 구조는 기본적으로 GPT-2의 capacity는 layer 수를 변형하여 조절

C. On Context-rich Vocabularies

factor m은 encoder에서 downsampling이 몇 번 이루어지는가? f = downsampling factor. ($f = 2^m$)

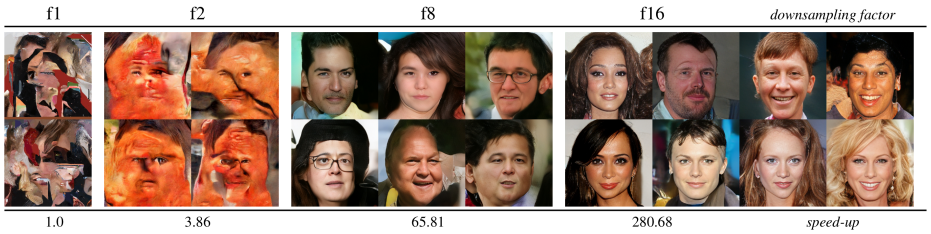


Figure 7. Evaluating the importance of effective codebook for HQ-Faces (CelebA-HQ and FFHQ) for a fixed sequence length $|s| = 16 \cdot 16 = 256$. Globally consistent structures can only be modeled with a context-rich vocabulary (right). All samples are generated with temperature $t = 1.0$ and top-k sampling with $k = 100$. Last row reports the speedup over the f1 baseline which operates directly on pixels and takes 7258 seconds to produce a sample on a NVIDIA GeForce GTX Titan X.

→ 성능에 중요함 (크면 클수록 → long-term interaction을 함께 학습가능)

→ 하지만 f가 크면 compression이 더 많이 되므로, 따라서 너무 크면 recon이 안좋

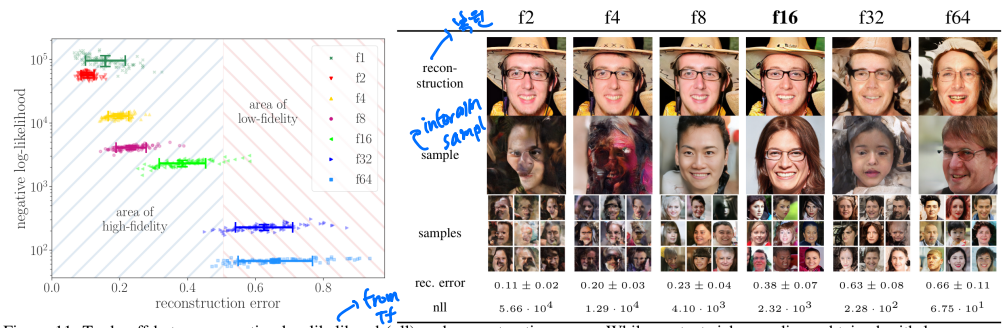


Figure 11. Trade-off between negative log-likelihood (nll) and reconstruction error. While context-rich encodings obtained with large factors f allow the transformer to effectively model long-range interactions, the reconstructions capabilities and hence quality of samples suffer after a critical value (here, $f = 16$). For more details, see Sec. C.

NLL은 image의 representation distribution의 modeling 능력에 대해 나타낸다.

f가 너무 크면 long-range에 대한 TF가 훈련이 잘되어서 NLL은 작지만 compression이 너무 많이 된다.

f가 작으면 recon은 잘되지만 sample quality↓

∴ 그 f로 perceptualy faithful한 recon을 encoding 해야한다.

Figure 12. Comparing reconstruction capabilities between VQVAEs and *VQGAN*s. Numbers in parentheses denote compression factor and codebook size. With the same compression factor and codebook size, *VQGAN*s produce more realistic reconstructions compared to blurry reconstructions of VQVAEs. This enables increased compression rates for *VQGAN* while retaining realistic reconstructions. See Sec. C.

↳ VQGAN, VQVAE, DALL-E 의 recon능력 비교.

f가 크면 압축적으로 덜 자세히 f가 크면, 성능↑

f의 크기는 dataset에 의존적이고.

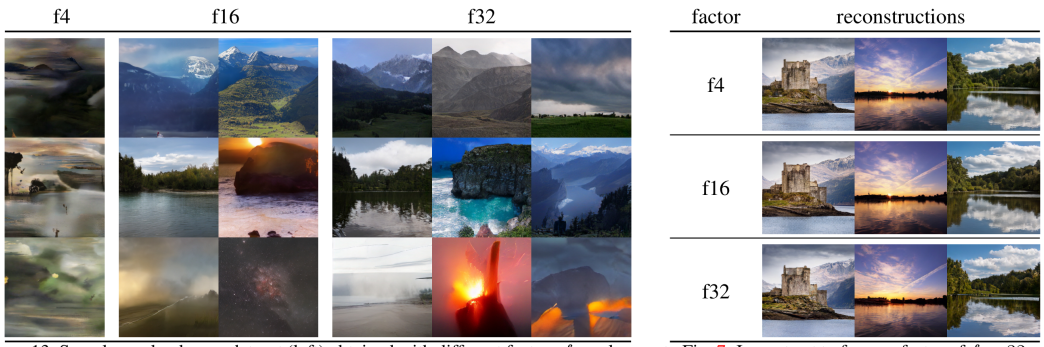


Figure 13. Samples on landscape dataset (left) obtained with different factors f, analogous to Fig. 7. In contrast to faces, a factor of $f = 32$ still allows for faithful reconstructions (right). See also Sec. C.

알고리즘 다르게 평가에서는 perceptualy 로 민감해서 f32가 성능이 좋다

* D. Additional Results.

- Qualitative Comparison.

Model	acceptance rate	FID	IS
mixed k, $p = 1.0$	1.0	17.04	70.6 ± 1.8
$k = 973$, $p = 1.0$	1.0	29.20	47.3 ± 1.3
$k = 250$, $p = 1.0$	1.0	15.98	78.6 ± 1.1
$k = 973$, $p = 0.88$	1.0	15.78	74.3 ± 1.8
$k = 600$, $p = 1.0$	0.05	5.20	280.3 ± 5.5
mixed k, $p = 1.0$	0.5	10.26	125.5 ± 2.4
mixed k, $p = 1.0$	0.25	7.35	188.6 ± 3.3
mixed k, $p = 1.0$	0.05	5.88	304.8 ± 3.6
mixed k, $p = 1.0$	0.005	6.59	402.7 ± 2.9
DCTransformer [48]	1.0	36.5	n/a
VQVAE-2 [61]	1.0	∼31	∼45
VQVAE-2	n/a	∼10	∼330
BigGAN [4]	1.0	7.53	168.6 ± 2.5
BigGAN-deep	1.0	6.84	203.6 ± 2.6
IDDPM [49]	1.0	12.3	n/a
ADM-G, no guid. [15]	1.0	10.94	100.98
ADM-G, 1.0 guid.	1.0	4.59	186.7
ADM-G, 10.0 guid.	1.0	9.11	283.92
val. data	1.0	1.62	234.0 ± 3.9

Table 4. FID score comparison for class-conditional synthesis on 256×256 ImageNet, evaluated between 50k samples and the training split. Classifier-based rejection sampling as in VQVAE-2 uses a ResNet-101 [22] classifier. BigGAN(-deep) evaluated via https://tfhub.dev/deepmind truncated at 1.0. "Mixed" k refers to samples generated with different top-k values, here $k \in \{100, 200, 250, 300, 350, 400, 500, 600, 800, 973\}$.

→ class마다 다른 성능

Dataset	ours-previous (+R)	BigGAN (-deep)	MSP
IN 256, 50K	19.8 (11.2)	7.1 (7.3)	n.a.
IN 256, 18K	23.5	9.6 (9.7)	50.4

Dataset	ours-previous	ours-new
CelebA-HQ 256	10.7	10.2
FFHQ 256	11.4	9.6

Table 6. Results from a previous version of this paper, see also Sec. A. Left: Previous results on class-conditional ImageNet synthesis with a slightly different implementation and evaluated against 50k and 18k training examples instead of the whole training split. See Tab. 4 for new, improved results evaluated against the whole training split. Right: Previous results on face-synthesis with a slightly different implementation compared to the new implementation. See also Tab. 3 for comparison with other methods.

⇒ 예시는 fig 14 ~ 17.

Semantic Synthesis

Dataset	**ours**	SPADE [53]	Pix2PixHD (+aug) [75]	CRN [9]
COCO-Stuff	22.4	22.6/23.9(*)	111.5 (54.2)	70.4
ADE20K	35.5	33.9/35.7(*)	81.8 (41.5)	73.3

Table 2. FID score comparison for semantic image synthesis (256×256 pixels). (*): Recalculated with our evaluation protocol based on [50] on the validation splits of each dataset.

⇒ 예시는 fig 40 ~ 41.

- Comparison to Image-GPT.

ImageGPT와 비교.

↳ TF가 image Synthesis와 image representation 모두에서 효과적이다.

↳ but 192x192 size의 한계

VQGAN은 high-res 학습 가능

⇒ 예시: fig 27, 28

- Additional high-res results

Depth-to-Image, Edge-to-Image, Semantic synthesis, unconditional LSUN, Pose-to-image 등이 fig 29 ~ 44 에 있음.

E. Nearest Neighbors of Sample.

GAN과 같이 likelihood-based gen model은 overfitting을 잡아내는 장점이 있다.

이를 위해 큰 model (overfit이 되기 쉬움) 에서 val에 대해 NLL이 가장 큰 epoch과

train에 대해 NLL이 가장 큰 epoch에서 각 dataset에 대해 유리.

각 epoch에서 sample을 통해, 가까운 사진 검색

train은 train set을 재구성. val은 새로운 data

∴ early-stopping이 과적합 방지,

↳ 위치는 dataset 크기에 비례

FID는 과적합을 방지를 못하고, val NLL은 품질은 보장 ×

둘다 정규화는 matrix이 됨.

VQGAN은 top-k로 다양성 확보.

F. On the Ordering of Image Representations.

NLP의 경우 인간이 순서를 정하지만, image는 명확 ×

특히 sliding의 경우 항이 우선시 되기 때문에 5가지 순법의 성능조사.

i) row major : 왼쪽 위에서 오른쪽 아래.

ii) Spiral out : 중심부터 나선형.

iii) Z-curve

iv) subsample :

v) alternative : 행을 채우는 방향은 교대로

vi) Spiral in : 밖부터 in으로

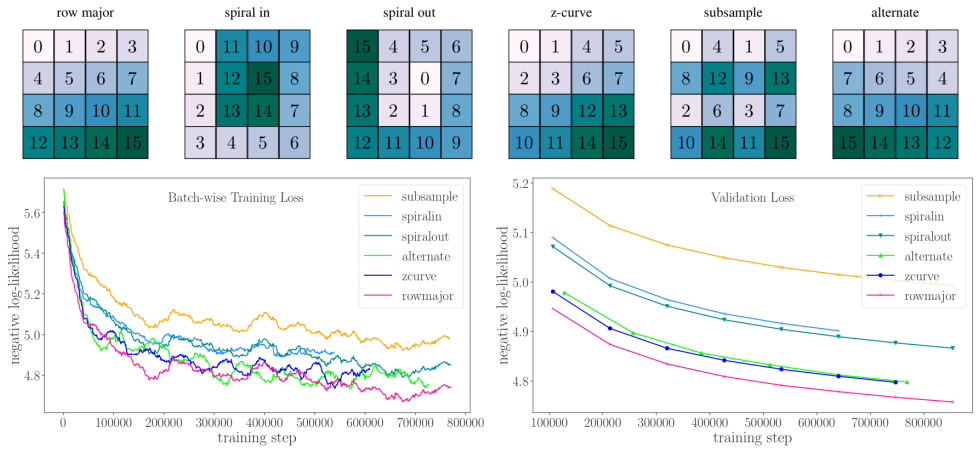


Figure 47. Top: All sequence permutations we investigate, illustrated on a 4×4 grid. Bottom: The transformer architecture is permutation invariant but next-token prediction is not: The average loss on the validation split of ImageNet, corresponding to the negative log-likelihood, differs significantly between different prediction orderings. Among our choices, the commonly used row-major order performs best.

Subsample과 spiral in의 평균 NLL이 가장 낮지만, 만족할 결과를 생성하지는.

다른 variant과 더 안정이 좋은 sample을 생성한다.

Subsample이 베틀적이고 bias의 도움이 되지도 않는다.

autoregressive codebook이 permutation-invariant이 아니지만, major row가 가장 성능 ↑.

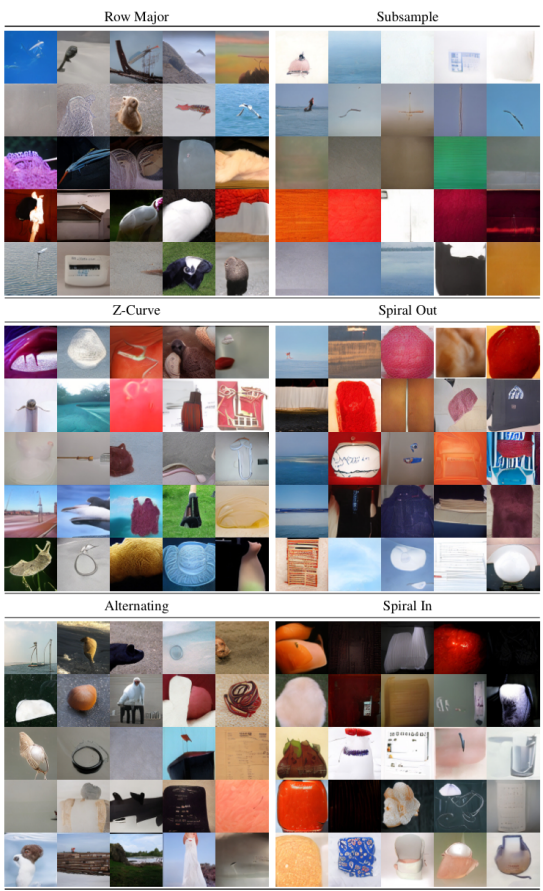


Figure 48. Random samples from transformer models trained with different orderings for autoregressive prediction as described in Sec. F.