Taming Transformers for High-Resolution Image Synthesis (VQ-GAN)

Members: Momal, Dane, Sanjay, Kasun, Mehdi
Group - I

Contents

Introduction

Architecture

Loss Functions

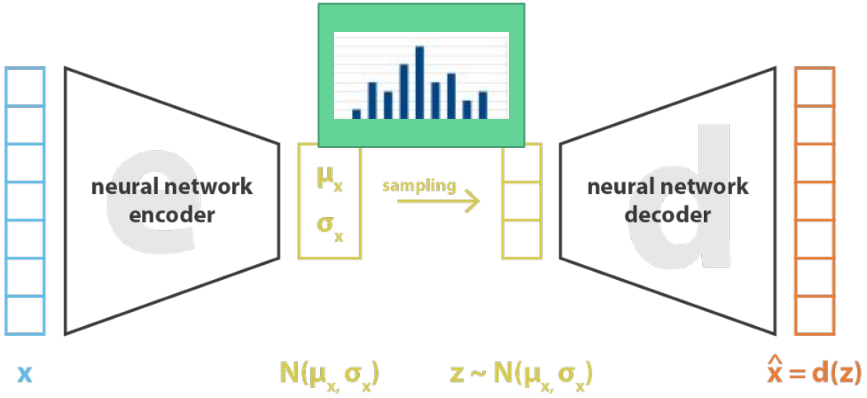
Conditional Synthesis

Experiments and Conclusion

Introduction

Two-Stage Approaches

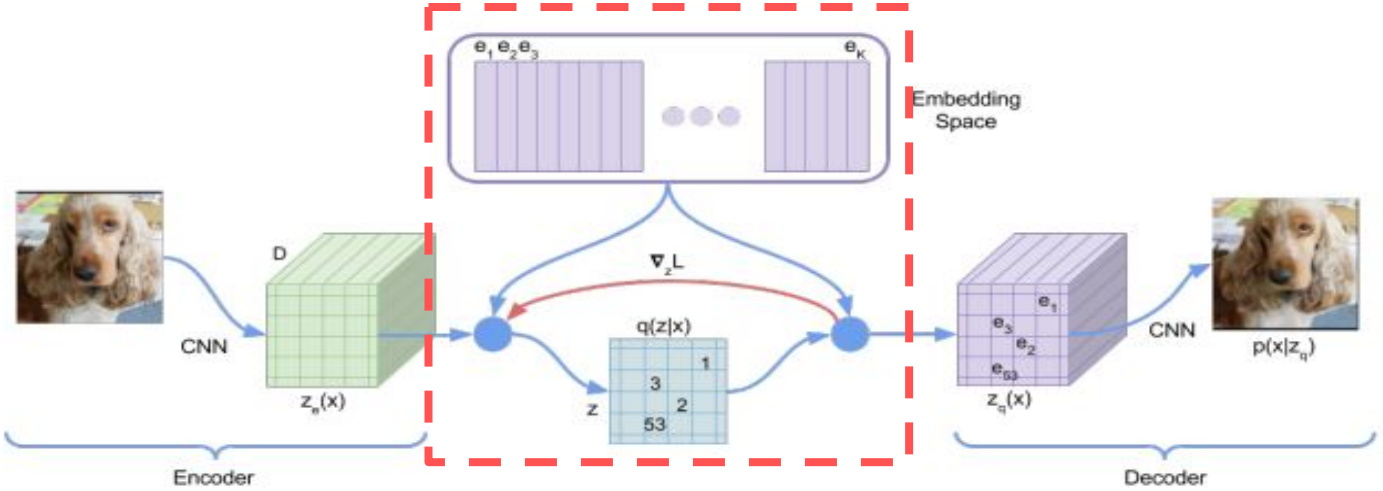
- First learn an encoding of data with VAE (encoder/decoder)
- Then learn a probabilistic distribution of the encoding



https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

VQ-VAE

- VQ-GAN built on top of VQ-VAE (Vector Quantized Variational Auto-Encoder)
- Learns discrete representations of images
- Only works on low resolution
- Global interactions aren't captured well by convolution density estimation



Hypothesis

- Low-level image structure is well described by a local connectivity
- Higher semantic levels are not locally connected



Idea

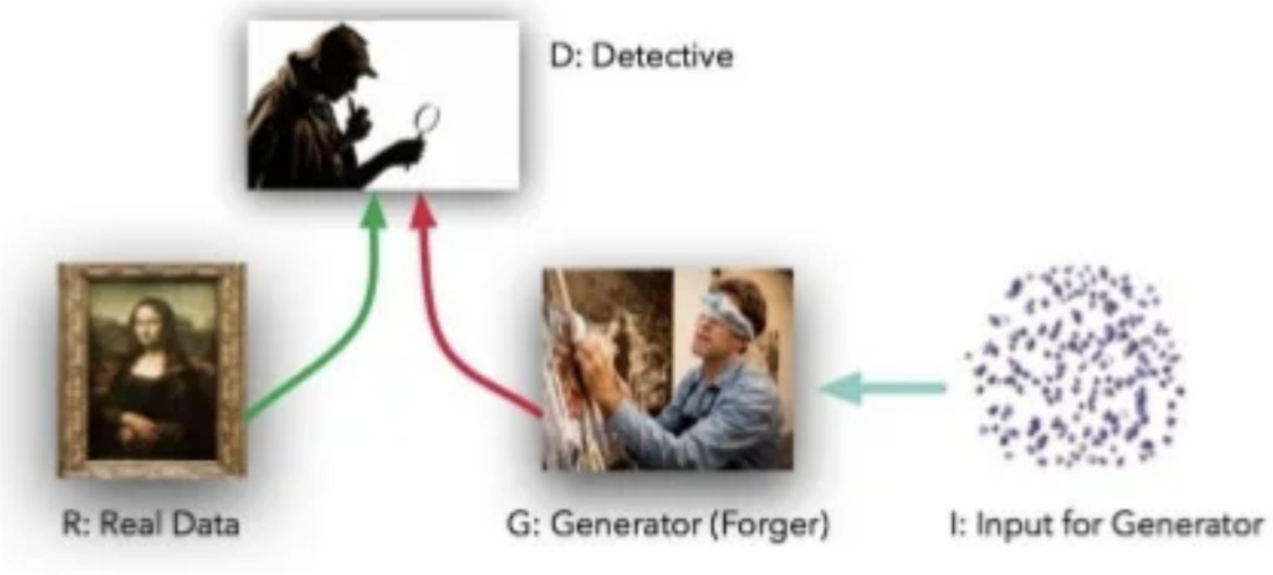
- Utilize transformers
- Take convolutional and transformer architecture together
- Use a convolutional approach to learn a codebook
- Learn global compositions
- Use an adversarial approach to ensure codebook learns important parts

VQ-GAN Architecture

Preliminaries

- Adversarial Training Procedure
- Autoregressive Models
- Quantization (Codebook)

Adversarial Training Procedure



Shamane Siriwardhana

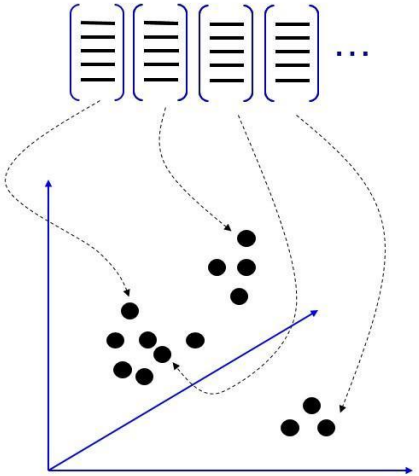
Autoregressive Models

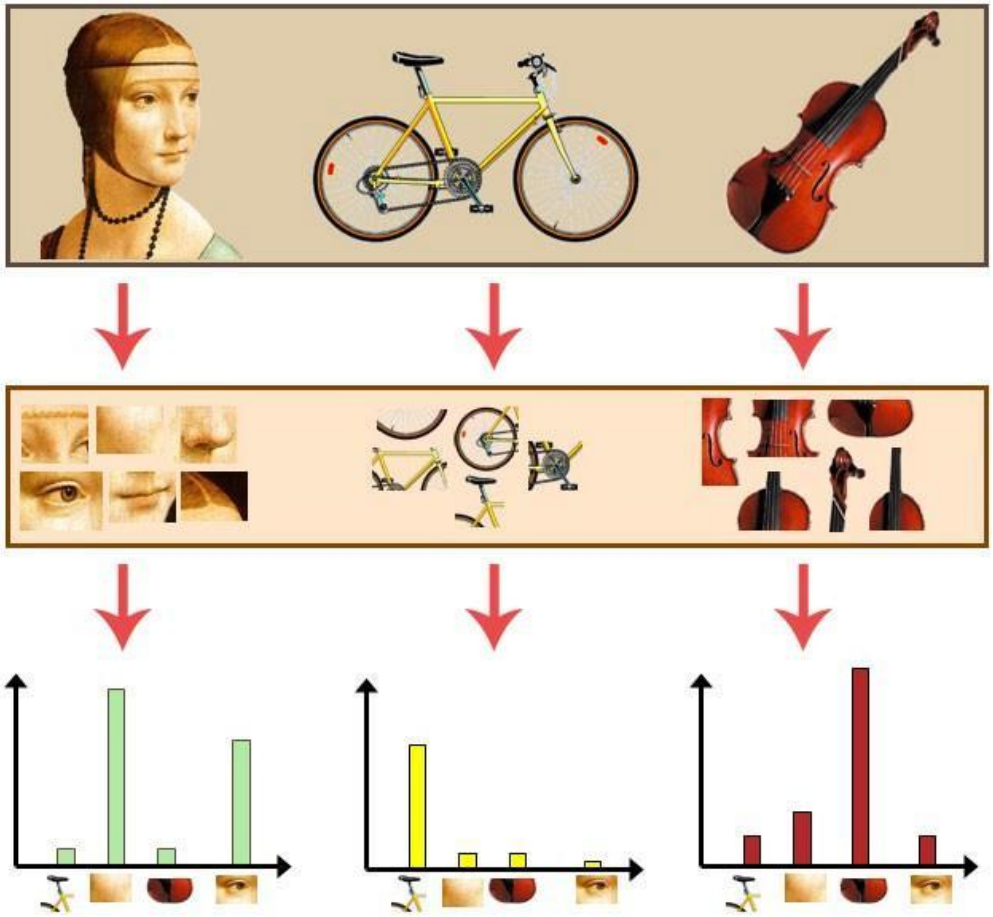
- The notation AR(p) indicates an autoregressive model of order p. The AR(p) model is defined as

$$X_t = c + \sum_{i=1}^{p} \phi_i X_{t-i} + \epsilon_t$$

Codebook explanation

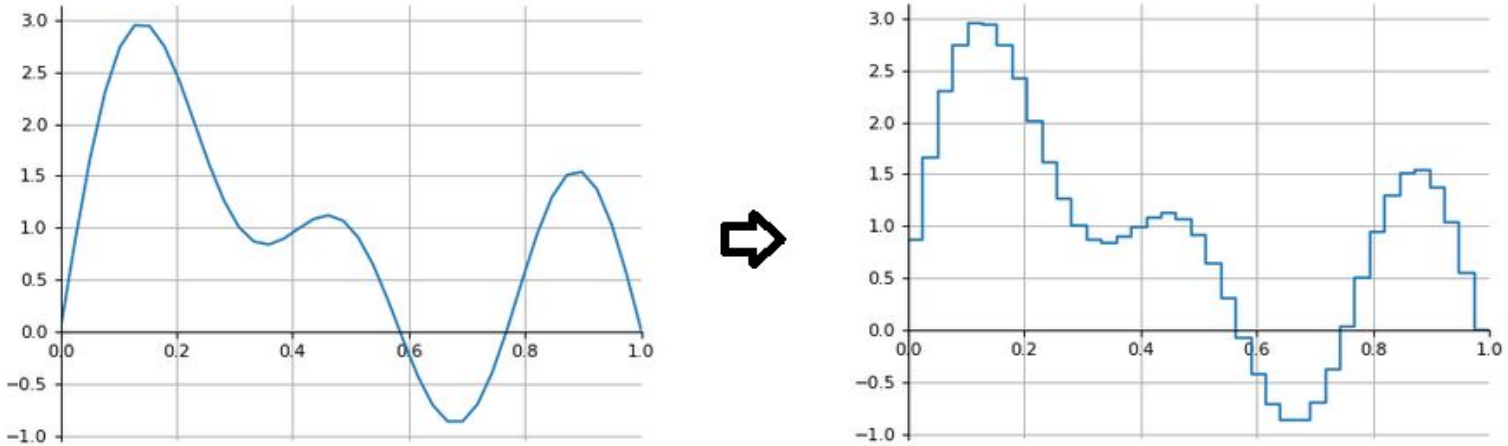
- Bag of visual words



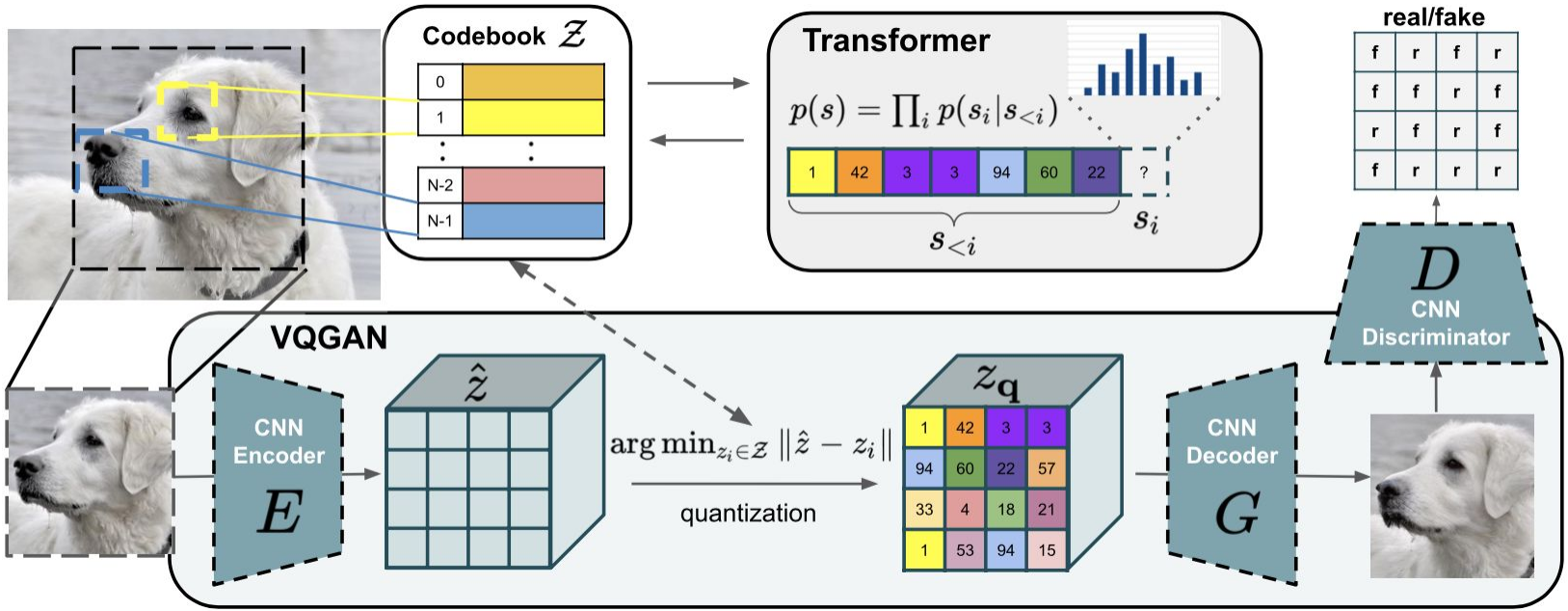


https://towardsdatascience.com/bag-of-visual-words-in-a-nutshell-9ceea97ce0fb

Quantization

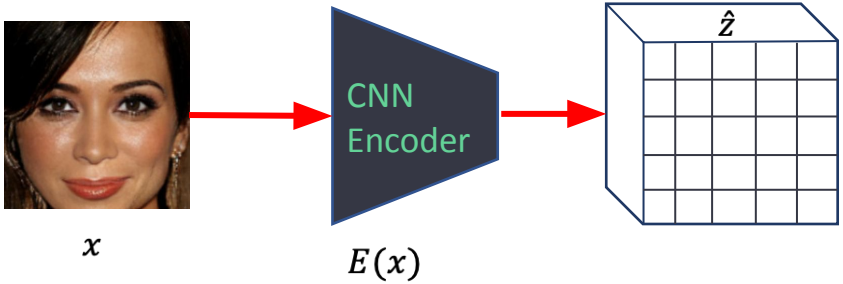


Architecture



Architecture Breakdown

Encoder Part

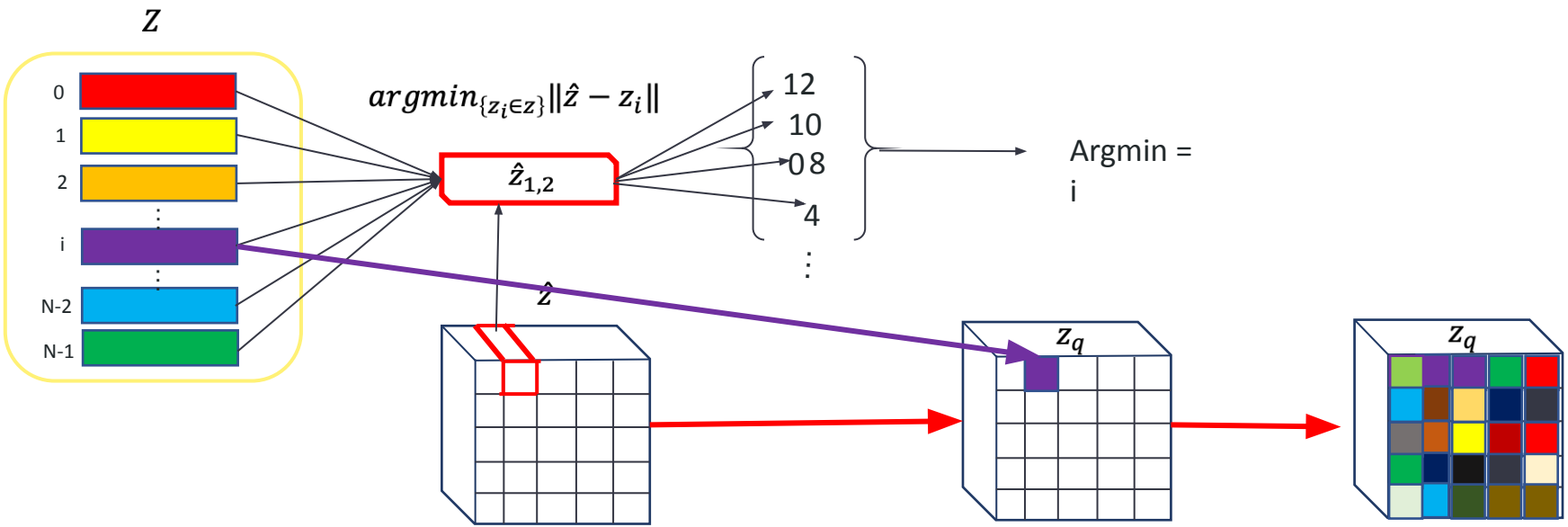


Encoder
$x \in \mathbb{R}^{H \times W \times C}$
Conv2D $\rightarrow \mathbb{R}^{H \times W \times C'}$
$m\times$ { Residual Block, Downsample Block} $\rightarrow \mathbb{R}^{h \times w \times C''}$
Residual Block $\rightarrow \mathbb{R}^{h \times w \times C''}$
Non-Local Block $\rightarrow \mathbb{R}^{h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

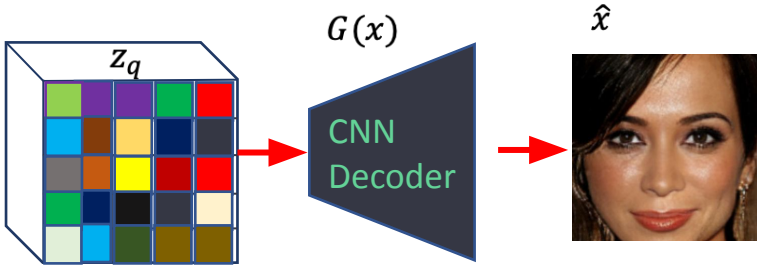
Quantization Part

- 1 - $z_q = q(\hat{z}) := \left(argmin_{z_k \in Z} \left\| \hat{z}_{ij} - z_k \right\|_2\right) \in \mathbb{R}^{h \times w \times n_z}$

  (Nearest Neighbor!)



Decoder Part

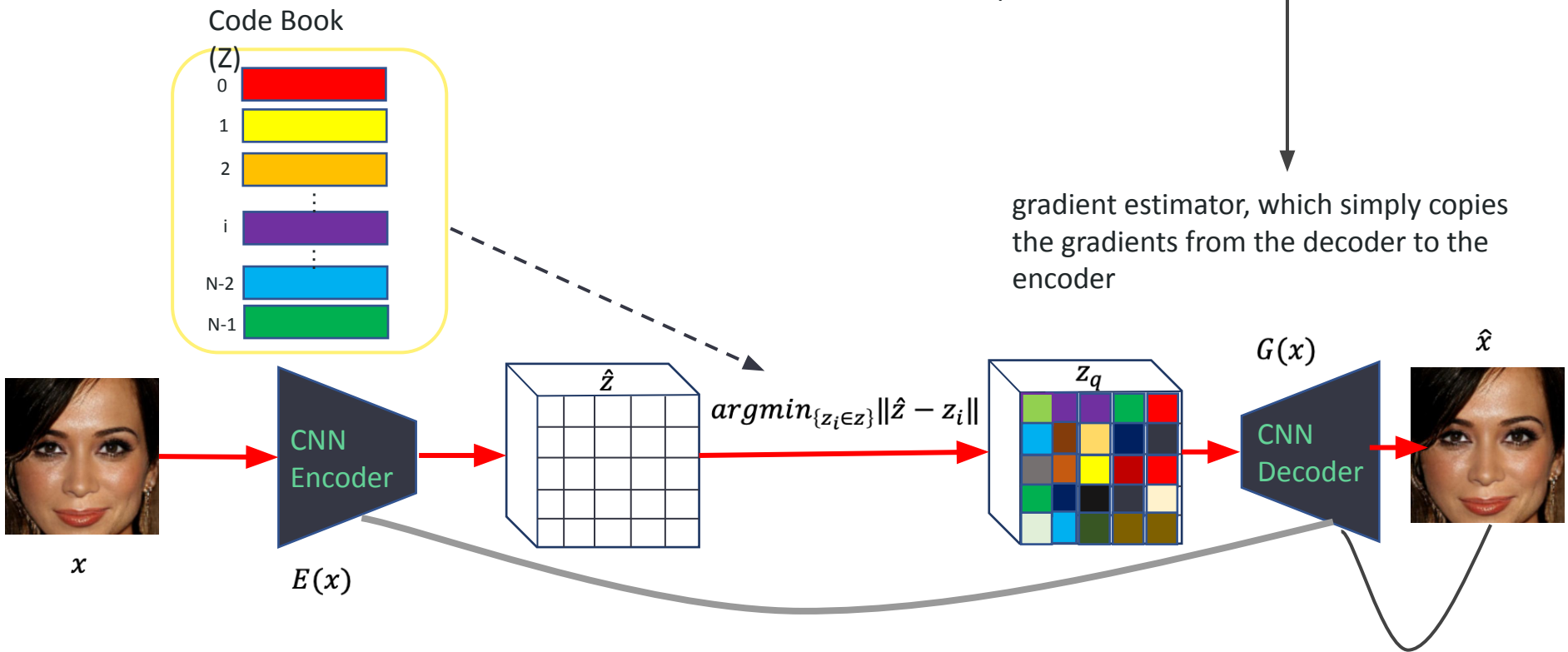


Decoder
$z_{\mathbf{q}} \in \mathbb{R}^{h \times w \times n_z}$
Conv2D $\rightarrow \mathbb{R}^{h \times w \times C''}$
Residual Block $\rightarrow \mathbb{R}^{h \times w \times C''}$
Non-Local Block $\rightarrow \mathbb{R}^{h \times w \times C''}$
Residual Block $\rightarrow \mathbb{R}^{h \times w \times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

Training The Codebook, Encoder and Decoder

- 2- $\hat{x} = G(z_q) = G(q(E(x))$ → non-differentiable because of quantization

gradient estimator, which simply copies the gradients from the decoder to the encoder



Loss Function

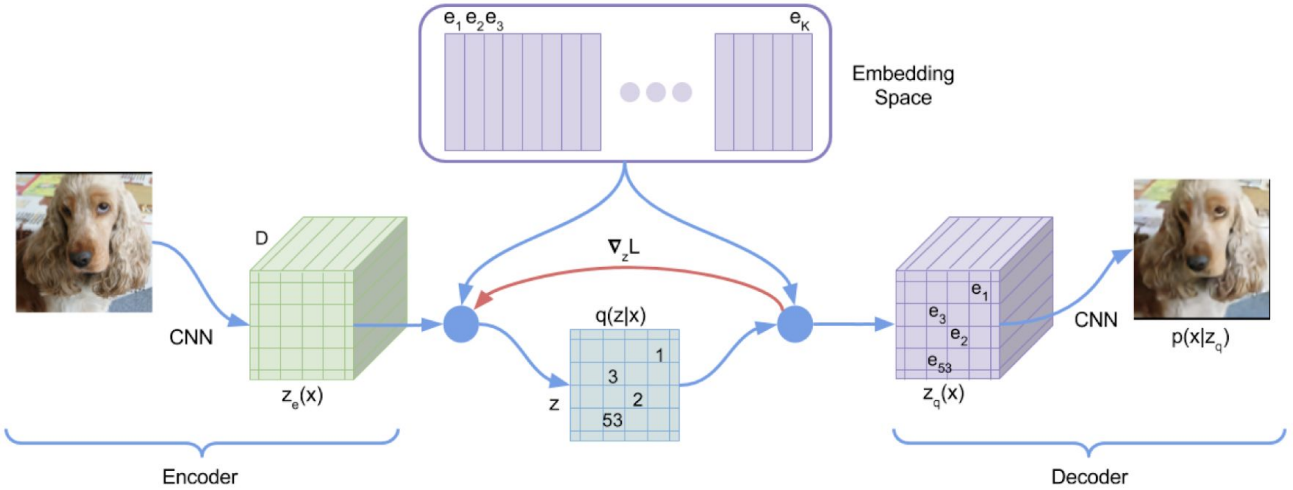
$$\mathcal{L}_{VQ}(E, G, Z) = \|x - \hat{x}\| + \left\|sg[E(x)] - z_q\right\|_2^2 + \left\|sg[z_q] - E(x)\right\|_2^2$$

Reconstruction Loss

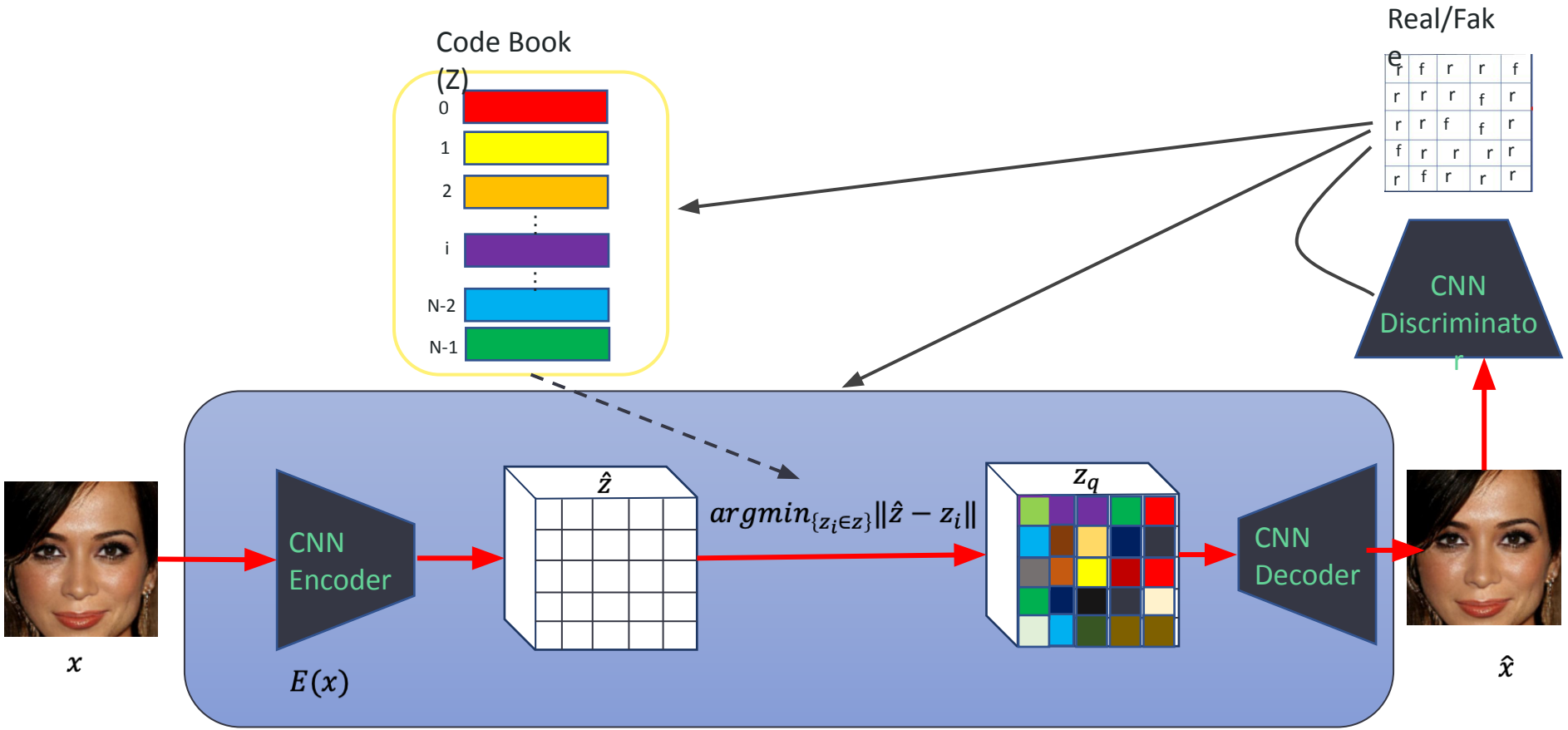
Euclidean distance of stop-gradient of encoder output and zq

commitment loss
Euclidean distance of stop-gradient of zq and the encoder
(will train the encoder)

VQ-VAE


e_1 e_2 e_3
e_K
Embedding Space
D
CNN
$z_e(x)$
$\nabla_z L$
q(z|x)
z
1
3
2
53
e_1
e_3
e_2
e_{53}
$z_q(x)$
CNN
$p(x|z_q)$
Encoder
Decoder

Adversarial Training Part for Richer Codebook



Learning the Composition of Images with Transformers (Autoregressive)

- Latent Transformers
  - The quantized encoding of an image x is given by
  - Which is equivalent to sequence of indices from codebook
  - Transformer basically generates the sequence when some indices of the sequence are given
  - This can be formulated as an autoregressive next index prediction

$$z_{\mathbf{q}} = \mathbf{q}(E(x)) \in \mathbb{R}^{h \times w \times n_z}$$

$$s \in \{0, \ldots, |\mathcal{Z}|-1\}^{h \times w}$$

Learning the Composition of Images with Transformers

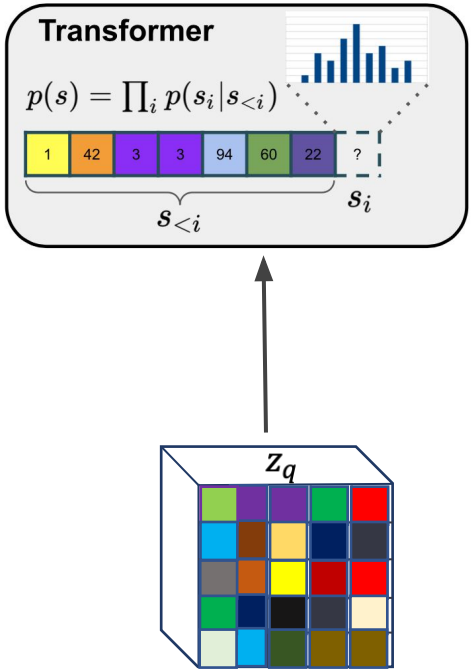
- Given indices s<i, the transformer learns to predict the distribution of possible next indices to compute the likelihood of the full representation as:

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

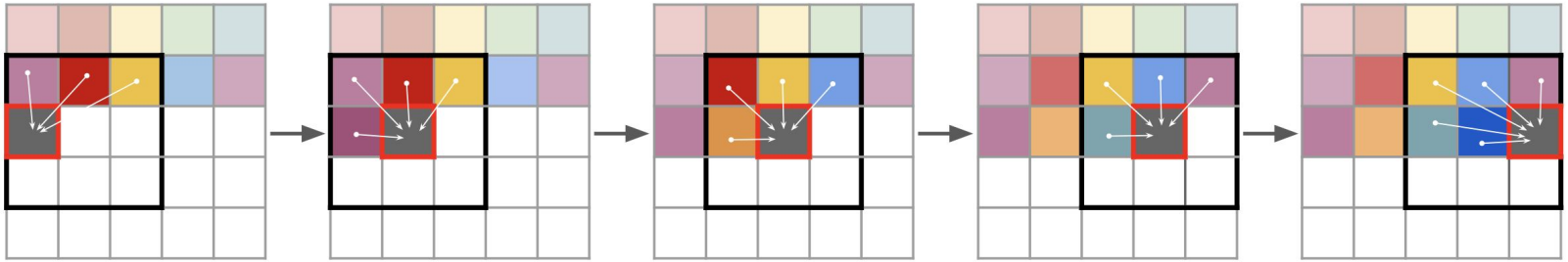
- The transformer is trained directly by maximizing the log likelihood:

$$\mathcal{L}_{transformer} = \mathbb{E}_{x \sim p(x)}[-\log p(s)]$$

Transformer Training

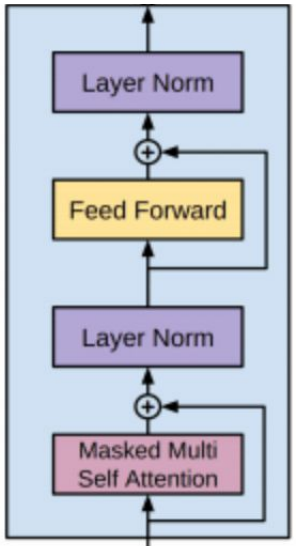


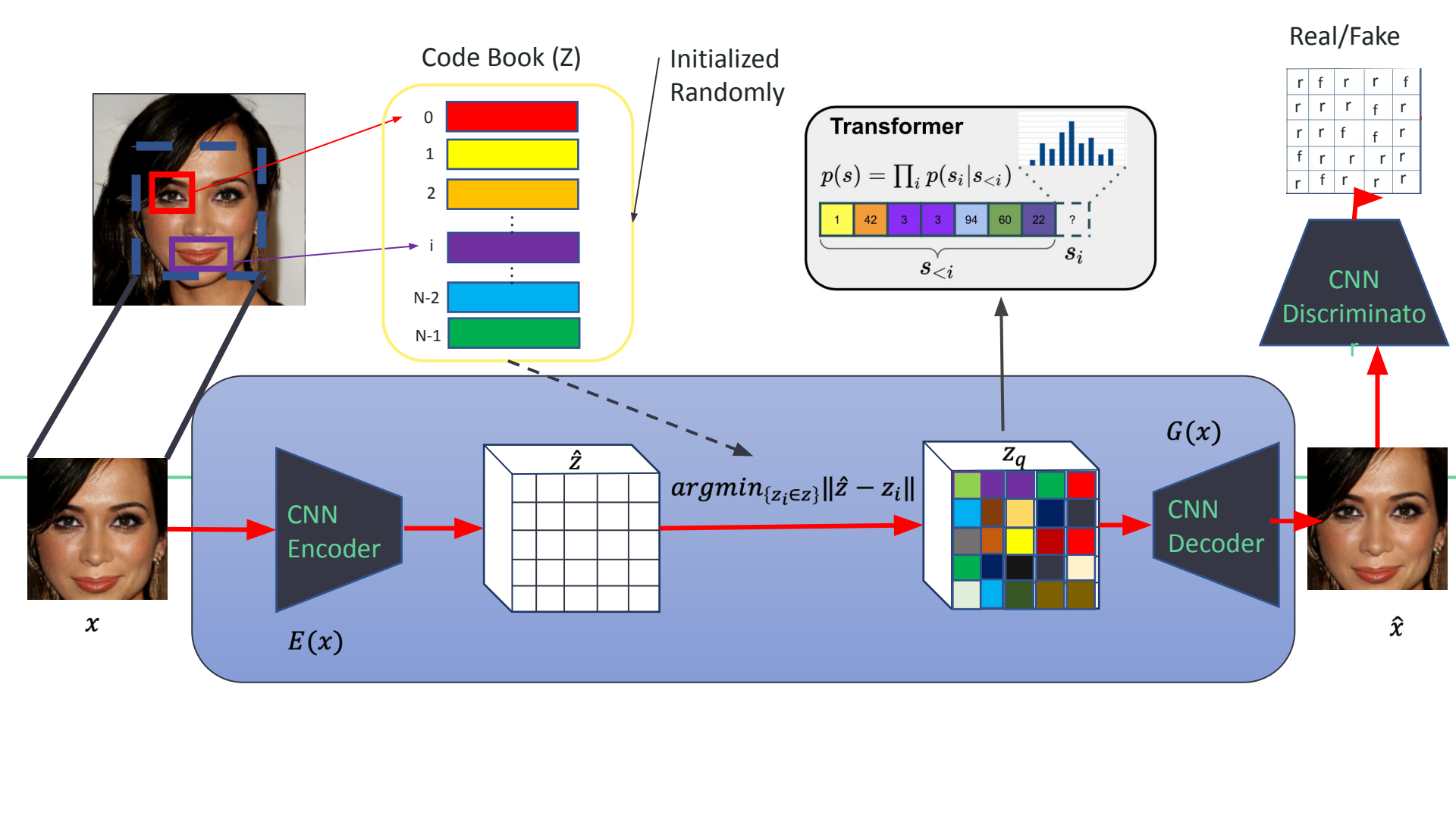
Sliding Window Transformer



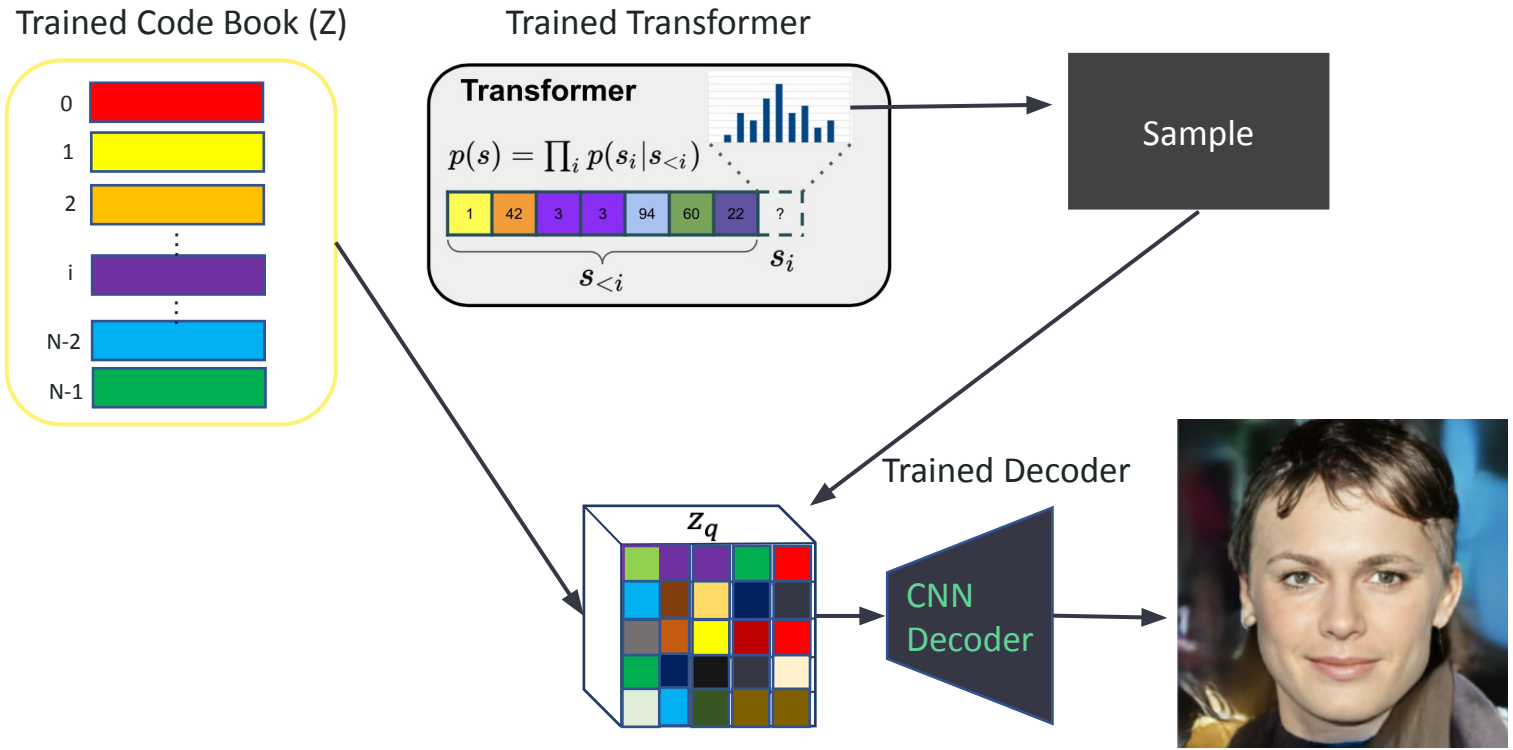
Transformer architecture of GPT2

Unsupervised Language model (NLP) developed by OpenAI

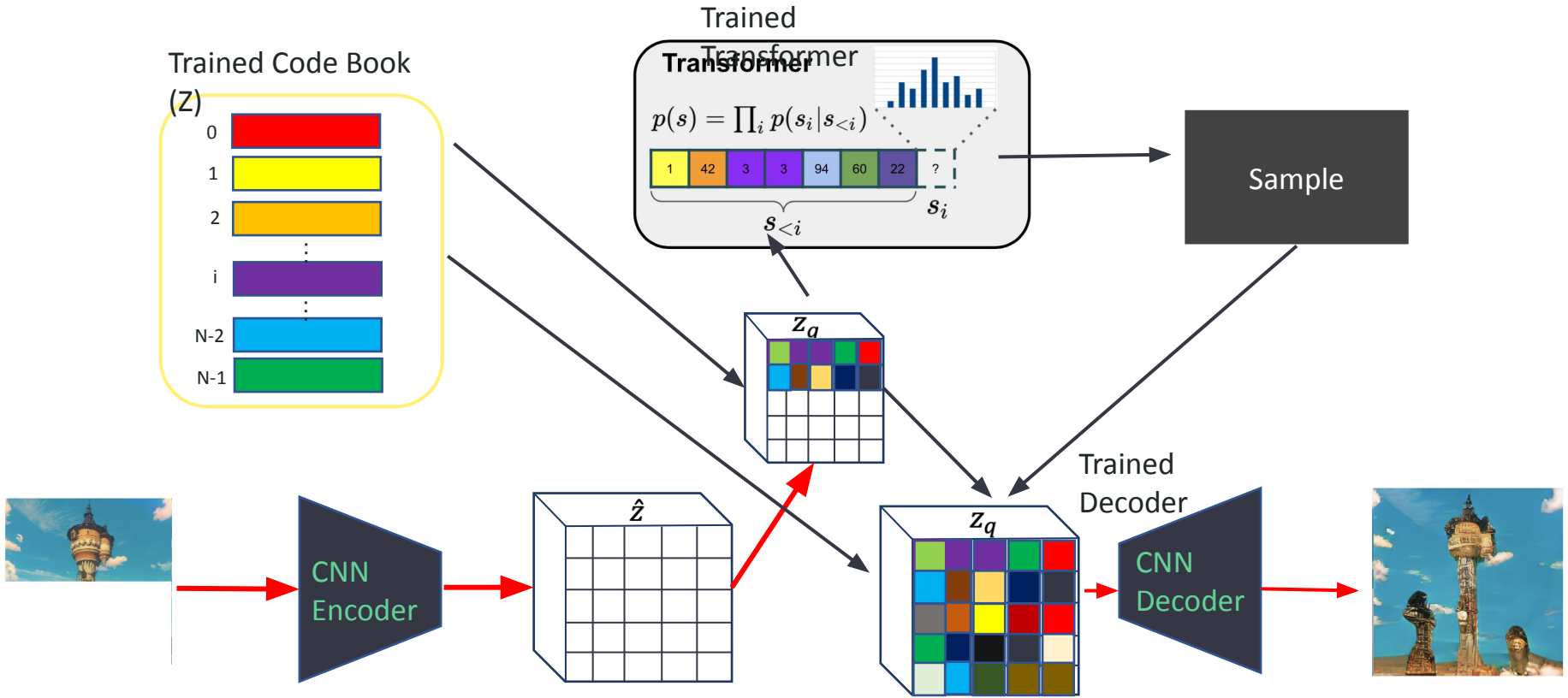



Code Book (Z)
Initialized Randomly
0
1
2
i
N-2
N-1
Transformer
$p(s) = \prod_i p(s_i|s_{<i})$
1 42 3 3 94 60 22 ?
$s_{<i}$
s_i
Real/Fake
r f r r f
r r r f r
r r f f r
f r r r r
r f r r r
CNN Discriminator
CNN Encoder
$E(x)$
$\hat{z}$
$argmin_{\{z_i \in z\}} \|\hat{z} - z_i\|$
z_q
$G(x)$
CNN Decoder
x
$\hat{x}$

Data Synthesis (unconditional)



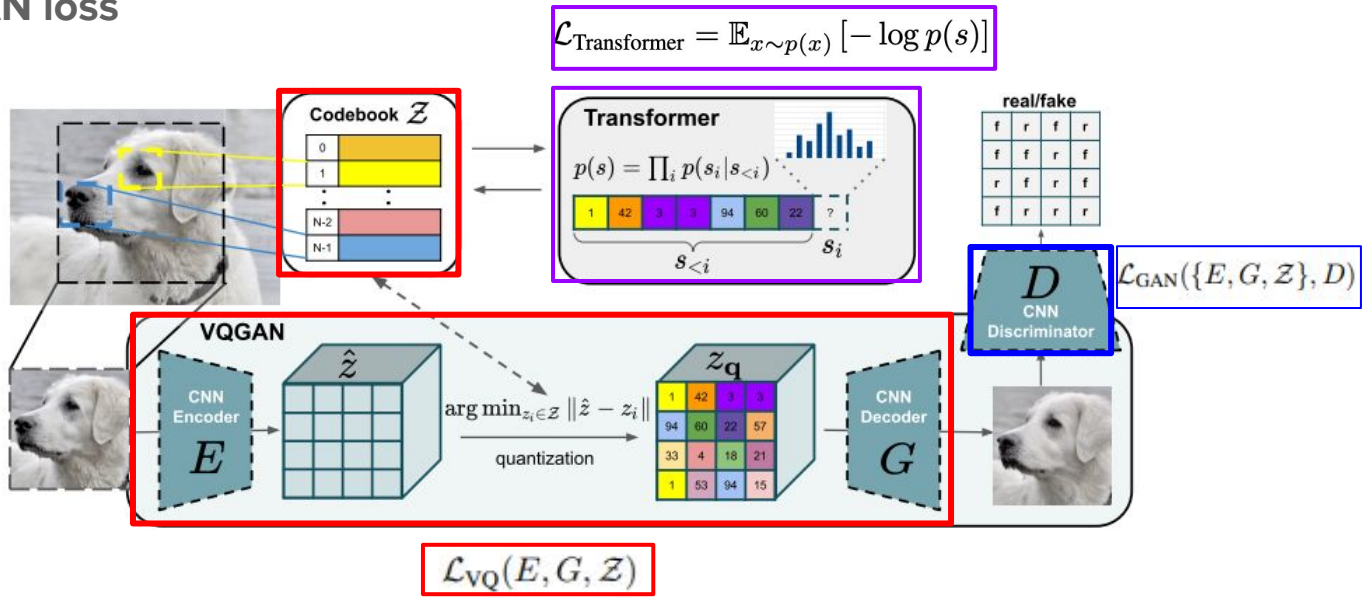
Data Synthesis (conditional)



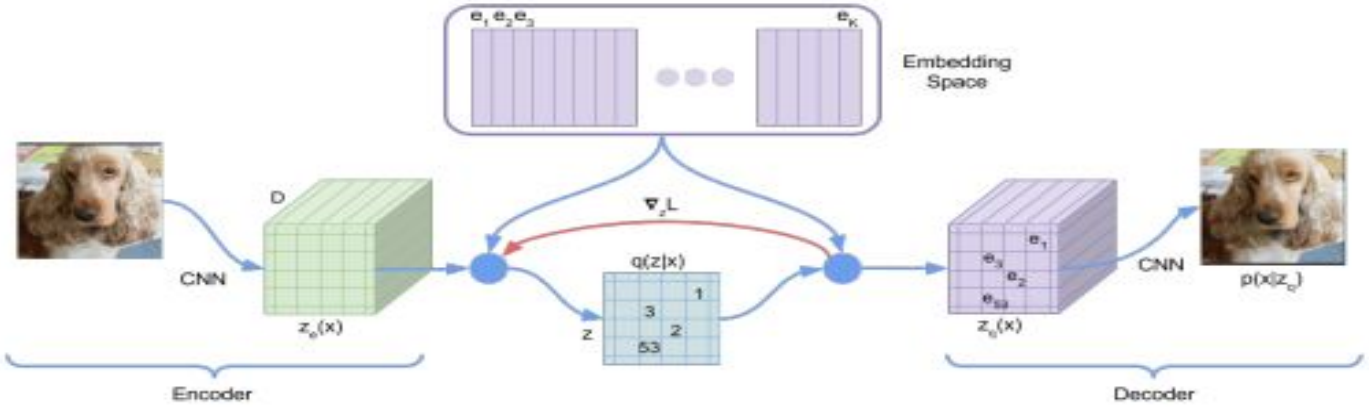
VQ-GAN Loss functions

Loss Function

- VQ-GAN losses
  - **Lvq - VAE loss**
  - **Lgan - GAN loss**
- Transformer Loss

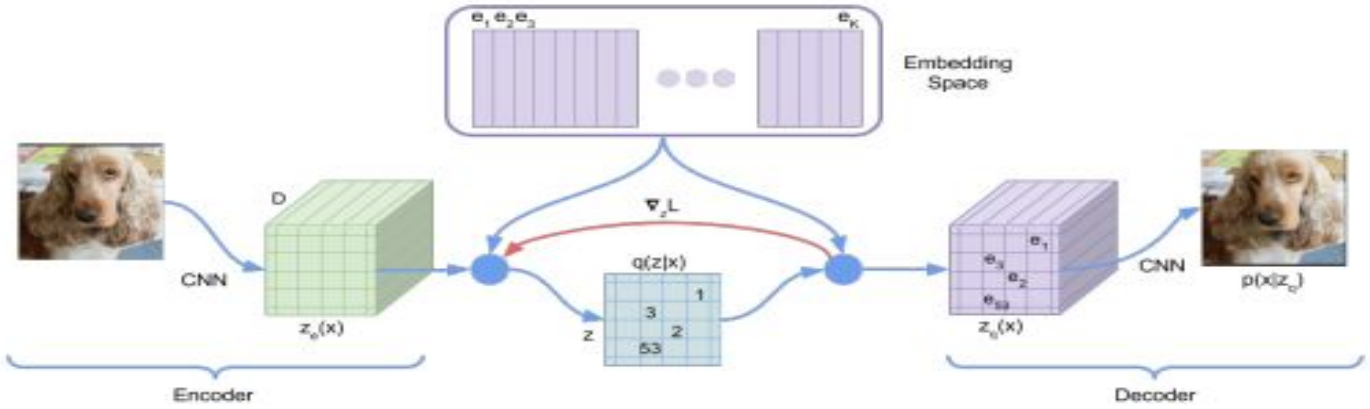


VAE Loss - Lvq $\mathcal{L}_{\mathbf{VQ}}(E, G, \mathcal{Z})$

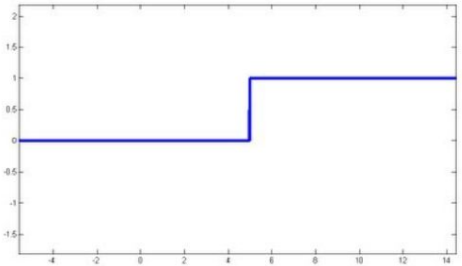


$$L = \log p(x|z_q(x)) + \|\text{sg}[z_e(x)] - e\|_2^2 + \beta\|z_e(x) - \text{sg}[e]\|_2^2,$$

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

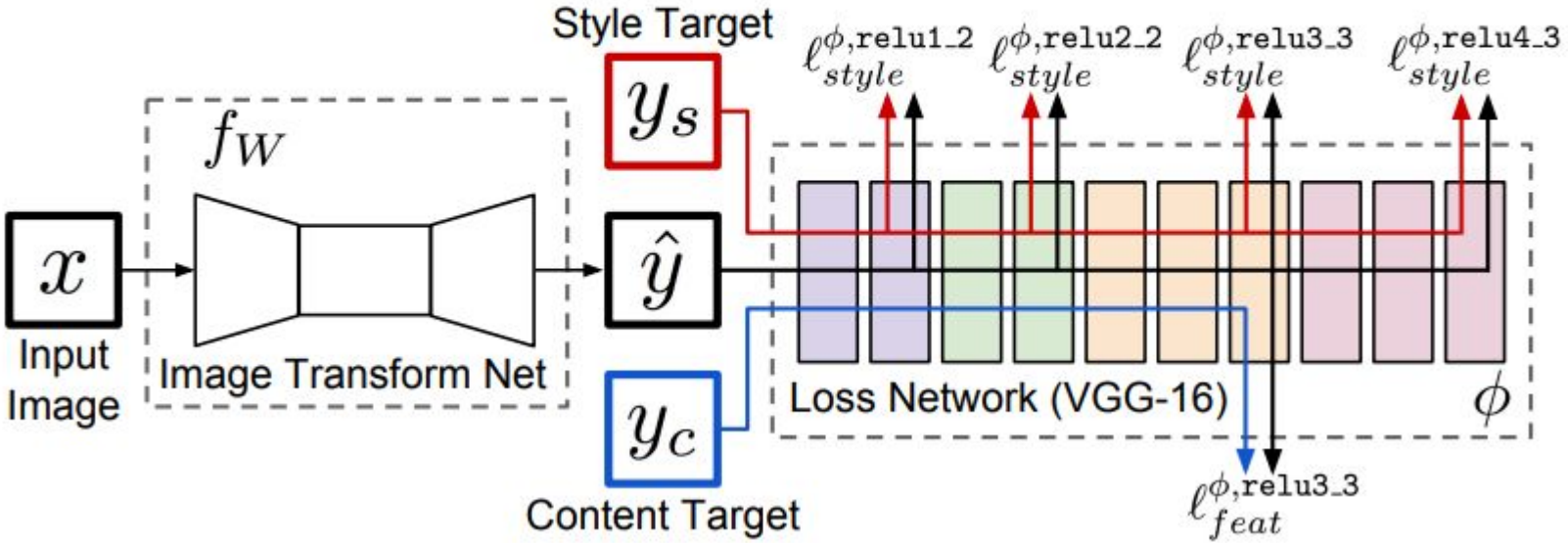


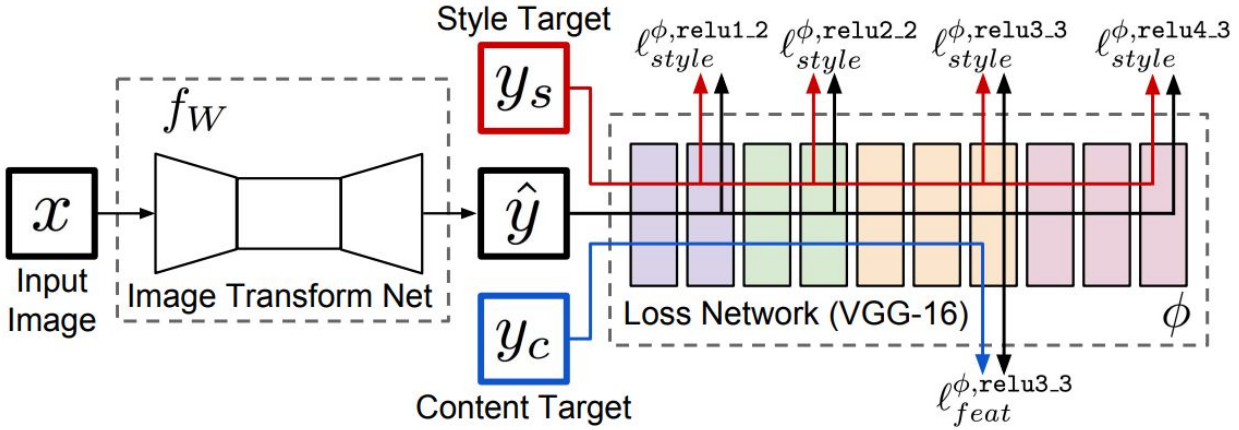
Replace with
Perceptual loss

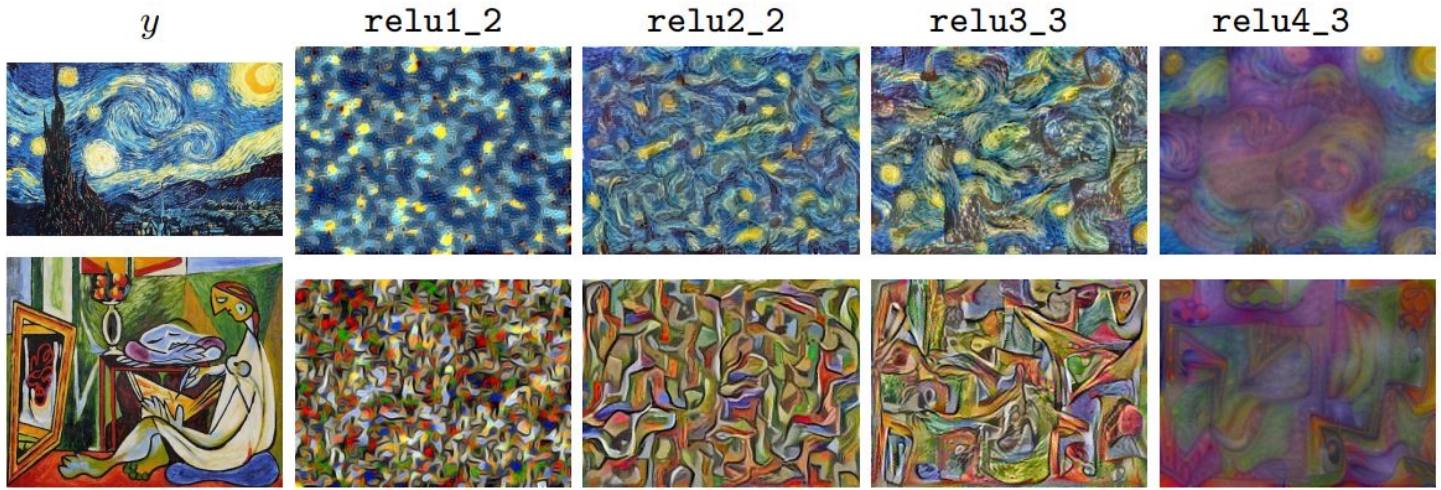


$$\mathcal{L}_{\mathrm{VQ}}(E, G, \mathcal{Z}) = \|x - \hat{x}\|^2 + \|\mathrm{sg}[E(x)] - z_{\mathbf{q}}\|_2^2 + \beta\|\mathrm{sg}[z_{\mathbf{q}}] - E(x)\|_2^2.$$

Perceptual loss - Reference from original paper




Style Target
y_s
$\ell^{\phi,\texttt{relu1_2}}_{style}$
$\ell^{\phi,\texttt{relu2_2}}_{style}$
$\ell^{\phi,\texttt{relu3_3}}_{style}$
$\ell^{\phi,\texttt{relu4_3}}_{style}$
f_W
x
Input Image
Image Transform Net
$\hat{y}$
Loss Network (VGG-16)
ϕ
y_c
Content Target
$\ell^{\phi,\texttt{relu3_3}}_{feat}$


y
relu1_2
relu2_2
relu3_3
relu4_3

GAN Loss - Lgan

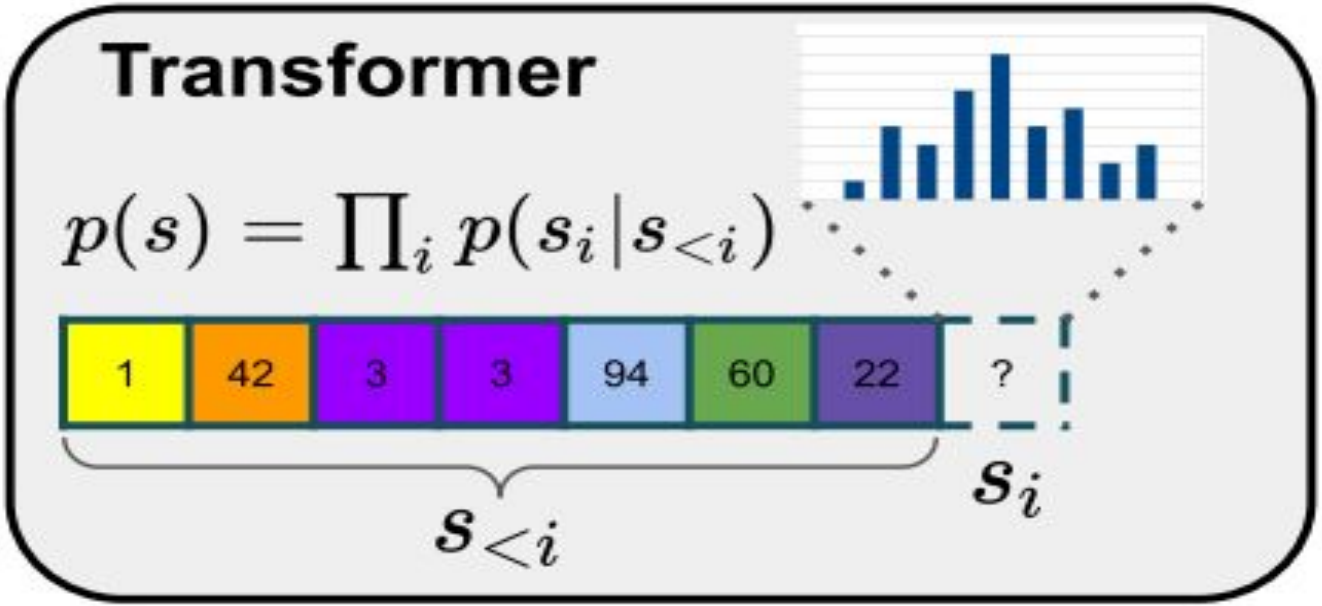
$$\mathcal{L}_{\mathrm{GAN}}(\{E, G, \mathcal{Z}\}, D) = [\log D(x) + \log(1 - D(\hat{x}))]$$

Final loss function of VQ-GAN

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

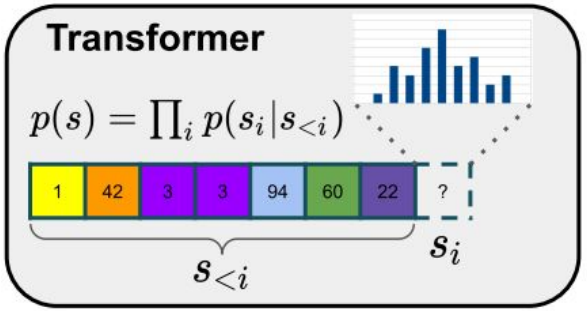
Where $\lambda = \dfrac{\nabla_{G_L}[\mathcal{L}_{\mathrm{rec}}]}{\nabla_{G_L}[\mathcal{L}_{\mathrm{GAN}}] + \delta}$

Autoregressive Transformer



Transformer Loss

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



- This sequence can be represented as a multiple of individual probabilities.

P(s)=p(s1|s<1) . p(s2|s<2) . p(s3|s<3) . .....

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

$$\mathcal{L}_{\text{Transformer}} = \mathbb{E}_{x \sim p(x)} \left[-\log p(s) \right].$$

- This will be maximized when it reaches to 1, which followed by log p(s) become 0.

Conditional Synthesis

Conditional Synthesis

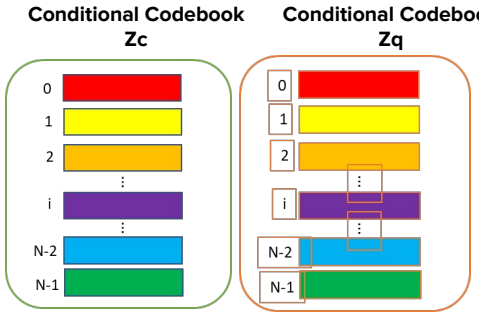
- Controlling image generation process by conditioning the input

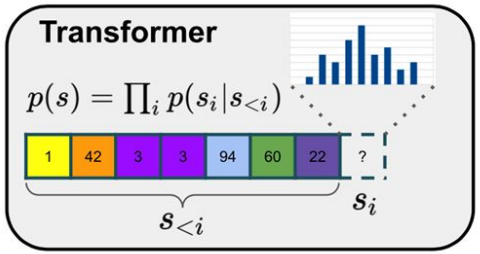
$$p(s|c) = \prod_i p(s_i | s_{<i}, c).$$

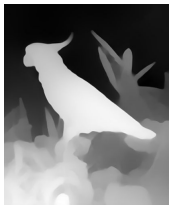
- Condition C can be
  - Class Label
  - Spatial Data – Depth image, low resolution image etc
- Image2Image [2]
  - Conditional synthesis is referenced from Pix2PixGAN

Conditional Image synthesis Process

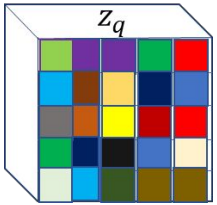
1. C is the conditional Spatial data
2. Zq = Codebook of Unconditional VQ-GAN
3. S = Index based representation of c from Zq
4. Train a new VQ-GAN on the conditional data
    a. Conditional Codebook Zc
    b. Indexed representation of c from Zc --> r
5. Prepend r with the s
6. Feed this sequence to transformer
7. Obtain synthesized output from Conditional Decoder







Conditional Image







Synthesized Output

Different Conditions - I

Cropped Image



Condition



Synthesized Variants

Depth Image



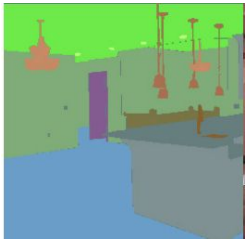
Condition



Synthesized Variants

Different Conditions - II

Semantically guided Condition

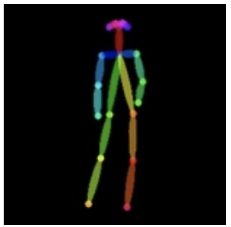


Condition



Synthesized Variants

Pose Estimated Condition



Condition

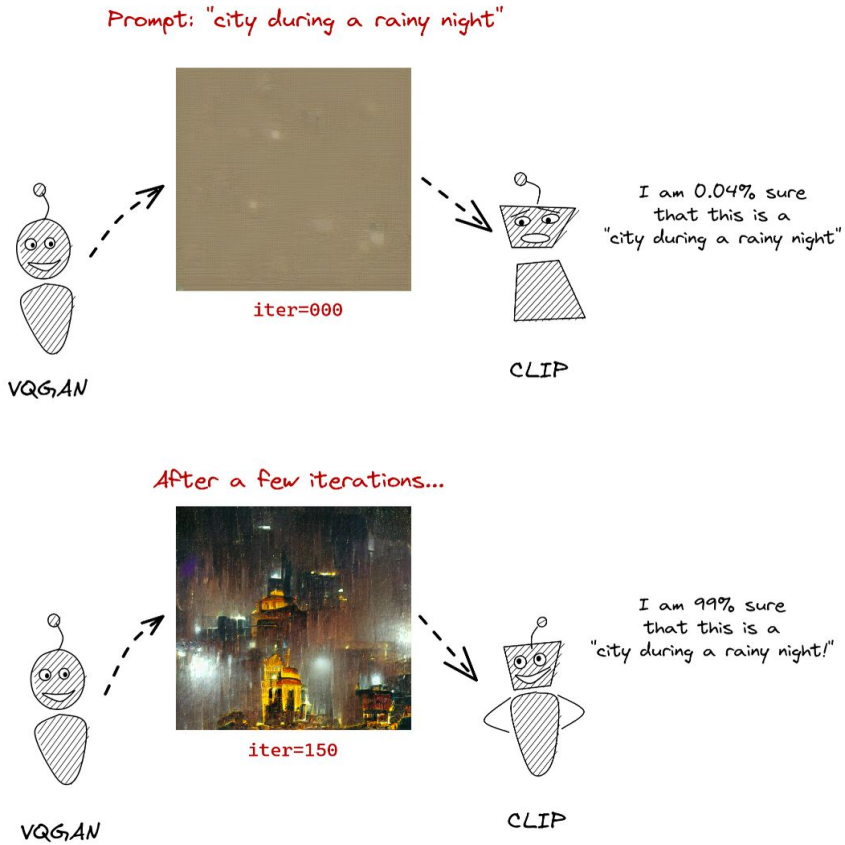


Synthesized Variants

CLIP + VQ-GAN



Text Input - “Old man with a moustache wearing a hat and smoking a cigarette art deco trending on Artstation”



Experiments and Results

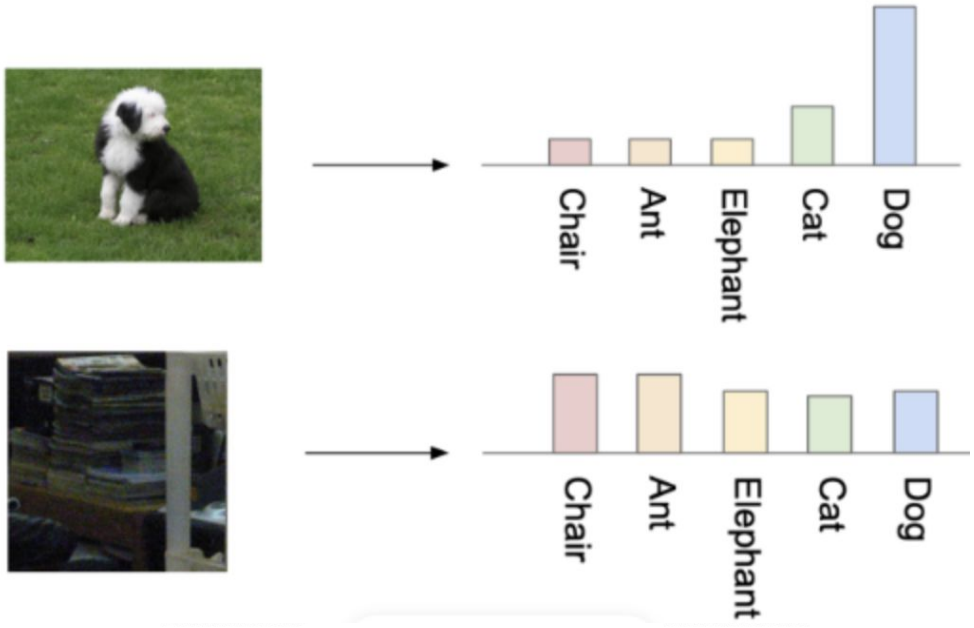
How to evaluate quality of generated images?

- Absence of a clear objective for generative models
- Need a score
  - Correlates well with human evaluations
  - Quantify quality of generated images
- Two Scores
  - FID - Fetcher Inception Distance
  - IS - Inception Score



Scores

- **FID Scores:**
  - Calculates similarity between features of original and generated image
  - Lower FID ➜ Better quality image
- **Inception score:**
  - Class prob of generated image by Inception Net [5]
  - Score seeks 2 props:
    - Image Quality
    - Image Diversity



Experiments - **1**
Comparison with Conv Model

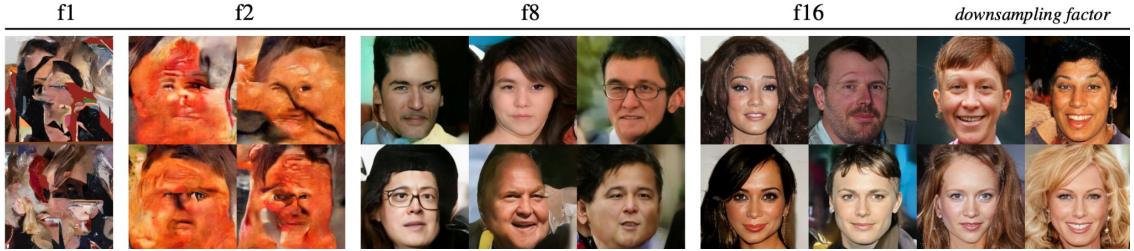
- Experimental setup
  - Num of Codebook Entries |Z| = **1024**
  - Predicted sequence length = **256**
- Comparison
  - PIXEL-SNAIL[1]
  - Trained on same codebook
  - Constrained training
- Transformers **outperform** P-SNAIL

	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

Experiments - **2**
High Resolution Images

- Sliding Window Attention ➔ High res images beyond 256x
  - Upto 1024x576
- Conditional Synthesis
  - Depth Map
  - High Resolution
  - Segmentation Map
  - Edged Images
- Unconditional Synthesis
  - LSUN-CT, FaceHQ
  - Reduction Factor f





Experiments - **3**
Class cond. Image Synthesis

- Model param
  - Codebook size = 16384
  - Image size = 256x
- Dataset: ImageNet
- VQ-GAN generates **better or comparable quality** images like SOTA networks

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

FID scores for class cond synthesis

Experiments - **4**
Uncond. Image Synthesis

- Comparison with existing generati models
  - GANs, VAEs, Flows, AR, Hybrid
  - Performance depended on dataset
- Decoding strategy
  - Beam Search best results **K=400**
  - **VQ-GAN not SOTA results**

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

FID scores for 256x images K=100

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)

FID scores for 256x images K=400

Conclusion

- Combines strength of Convolutional networks with transformers
  - Local correlations - Conv
  - Long Range dependencies - Transformers
- Code book or Quantized vectors instead of pixels
  - Long range dependencies
  - No more Quardratic Computational complexity
- Proposed hybrid approach - Outperform SOTA convolutional approaches
- BottleNeck:
  - Training additional VQ-GAN for conditional synthesis

References

[1] Xi Chen, Nikhil Mishra, Mostafa Rohaninejad, and Pieter Abbeel. Pixelsnail: An improved autoregressive generative model. In ICML, volume 80 of Proceedings of Machine Learning Research, pages 863–871. PMLR, 2018. 2, 5

[2] Isola, Phillip et al. “Image-to-Image Translation with Conditional Adversarial Networks.” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017): 5967-5976.

[3] https://machinelearningmastery.com/how-to-implement-the-frechet-inception-distance-fid-from-scratch/

[4] Radford, Alec, et al. "Language models are unsupervised multitask learners." *OpenAI blog* 1.8 (2019): 9.

[5] https://machinelearningmastery.com/how-to-implement-the-inception-score-from-scratch-for-evaluating-generated-images/

[6] https://towardsdatascience.com/bag-of-visual-words-in-a-nutshell-9ceea97ce0fb

[7] https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73