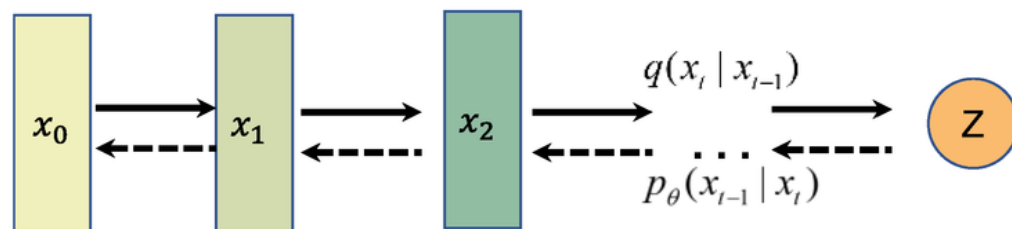
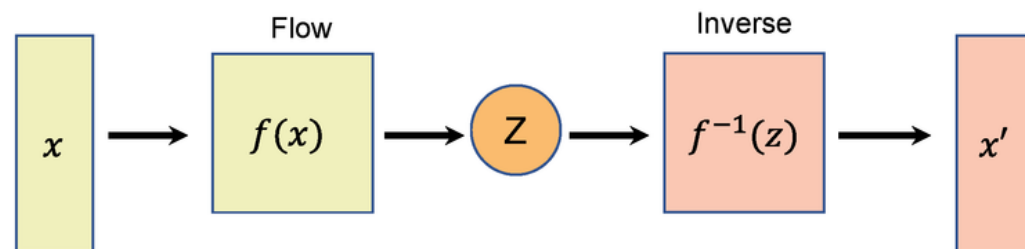
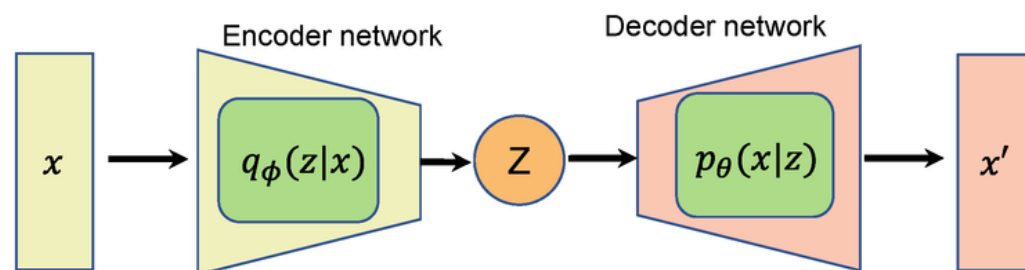
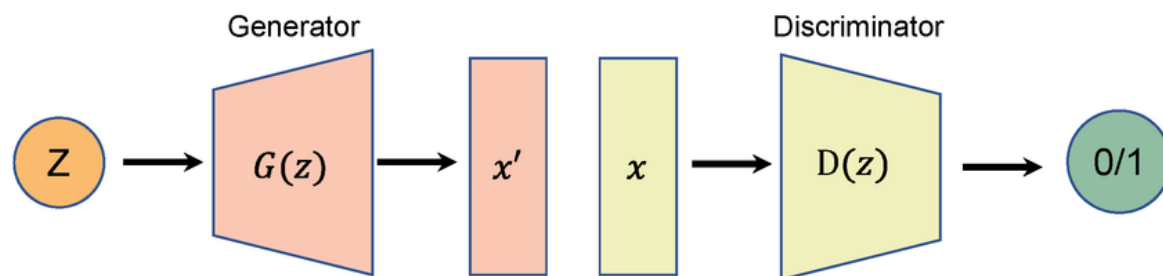
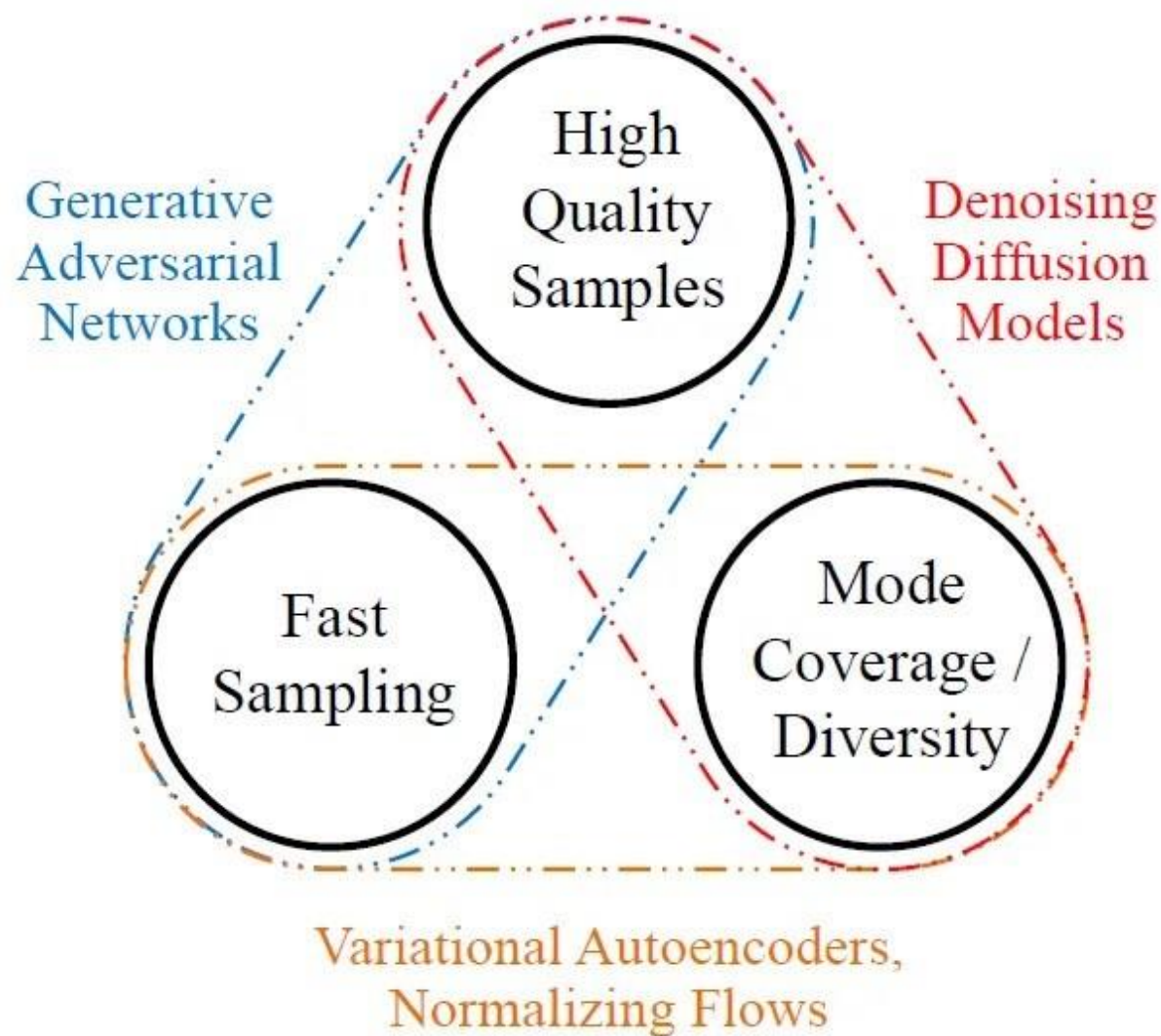


Генеративные модели Computer Vision

Угадай семейство моделей

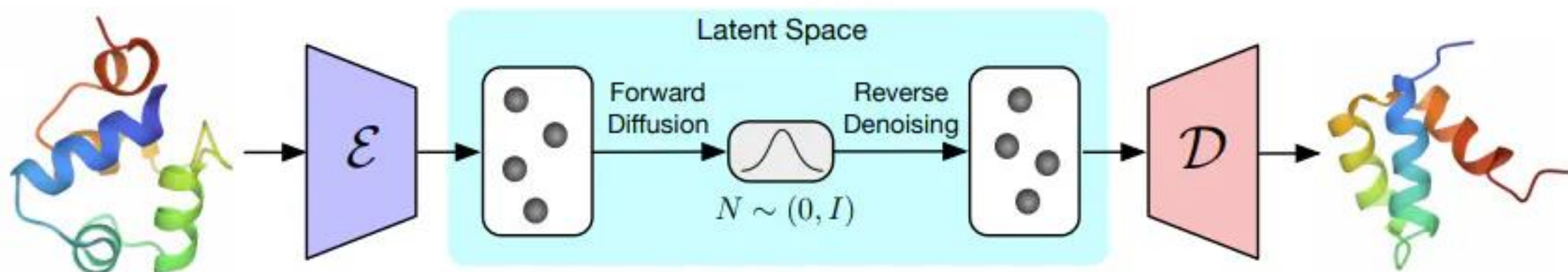


Generative learning trilemma



Source: <https://zhuanlan.zhihu.com/p/503932823>

Stable Diffusion (Diffusion + VAE + CLIP)



Добавление CLIP-эмбеддингов

Cross-attention:

- У нас есть внутренние представления картинки на текущем шаге диффузии (features)
- У нас есть CLIP-эмбеддинг текста (тоже в виде векторов)

UNet делает cross-attention:

- Для каждого «пиксельного» или patch-представления картинки
- Смотрит на все векторы из текстового эмбеддинга
- Решает, какая часть текста важна для этого участка изображения

Модель взвешивает информацию из текста и картинки, чтобы обновить шум так, чтобы он стал ближе к описанию.

Добавление CLIP-эмбеддингов

1. Queries (Q) — это патчи картинки (текущие feature-векторы в UNet на шаге диффузии).

- Каждая маленькая часть картинки “спрашивает”: «На что из текста мне обратить внимание?»

2. Keys (K) и Values (V) — это векторы слов из CLIP-эмбединга текста

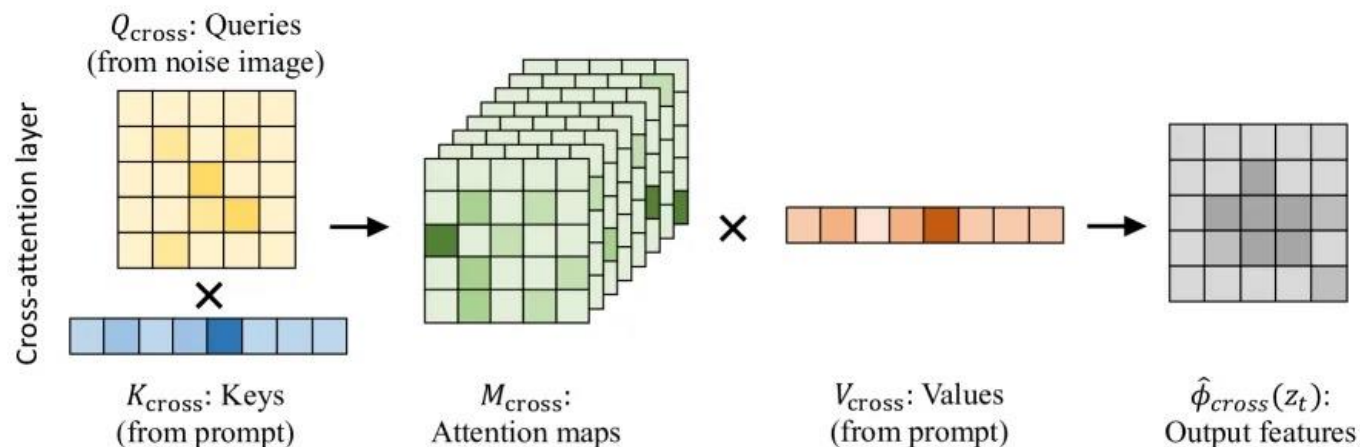
- Каждое слово текста превращается в вектор.

- Keys отвечают за «идентификацию важности слова», Values — за информацию, которую можно использовать.

3. Attention:

1. Для каждого патча Query считается вес сходства с каждым Key \rightarrow softmax \rightarrow веса внимания

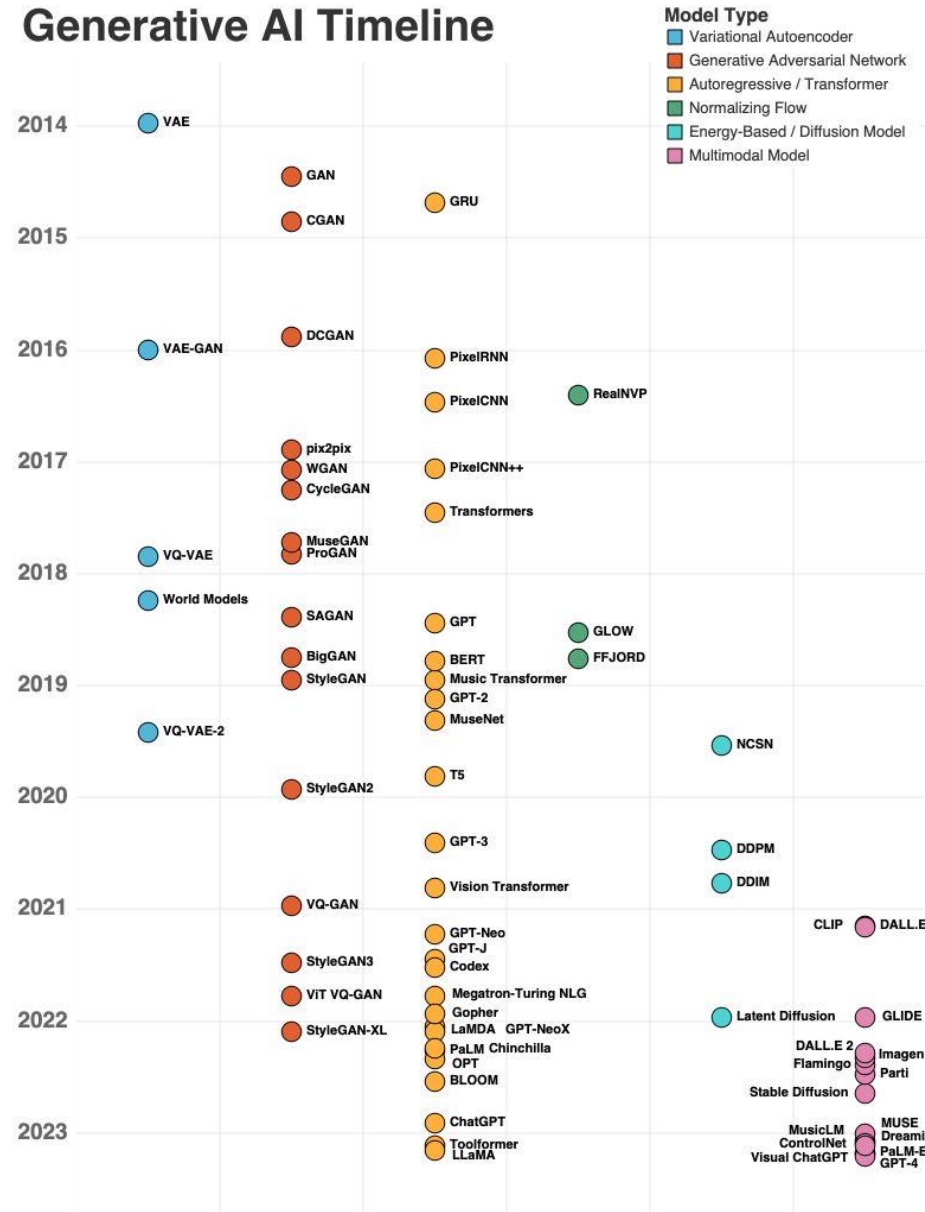
2. Query суммирует Values, взвешенные этими коэффициентами \rightarrow обновляет патч картинки



Практика по Stable Diffusion

<https://colab.research.google.com/drive/1evC2-9flfkMOchqZuYQ2fgKWWUbHqpxb?usp=sharing>

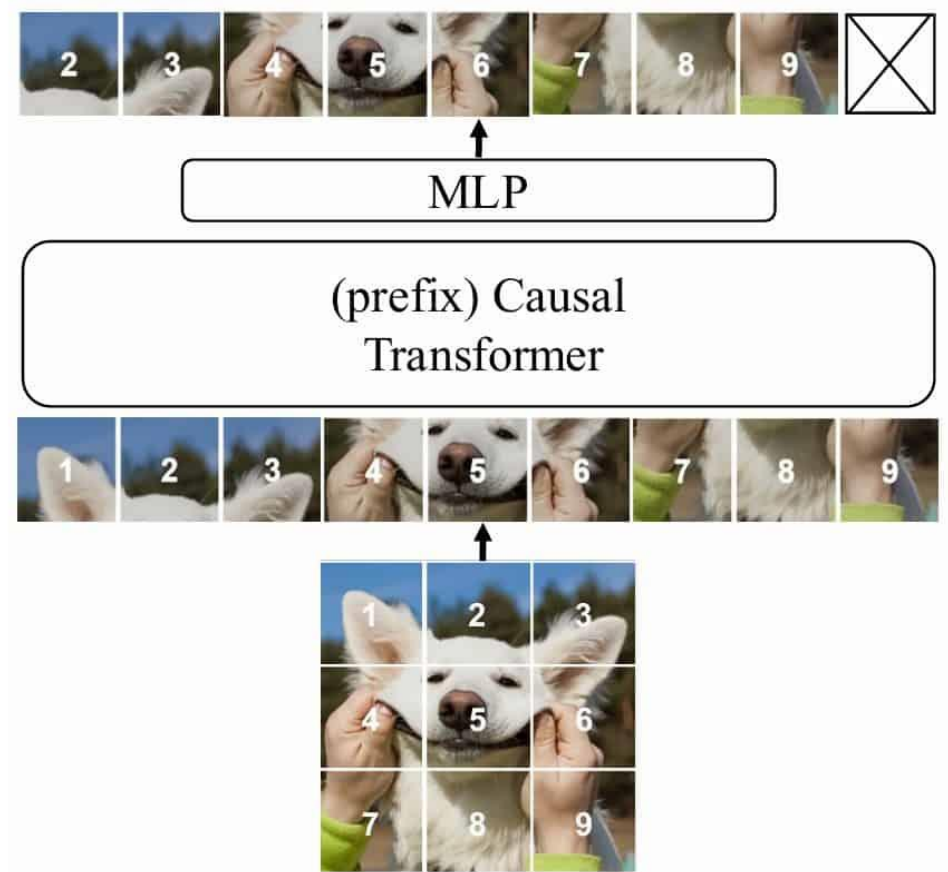
Timeline генеративных CV-моделей



Авторегрессионные модели

Идея AR-моделей

Авторегрессионные модели (как и в NLP) учатся прогнозировать кусочки изображения “токен за токеном”

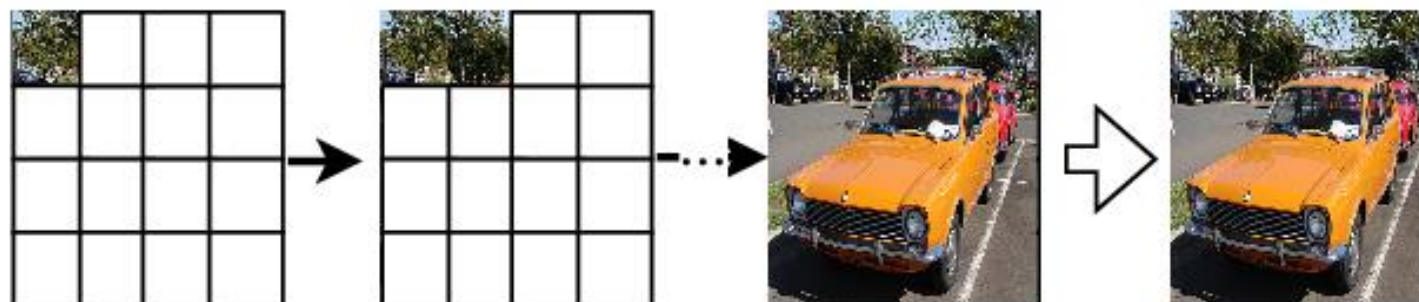


Идея AR-моделей







Токен картинки (или же patch) – это кусочек изображения, одна “клеточка”.

- Изображение разбивается на патчи, затем эти патчи векторизуются (для каждого патча модель выучит числовой вектор-embedding) и подаются в модель

Next-Patch Prediction [Taming Transformers, 2020]



Сравнение AR-моделей с Stable Diffusion

Model	Quality	Diversity	Speed
AR (modern)	 9.5	 7	 8
Diffusion (Stable Diffusion XL)	 8.5	 9	 6

- **AR:** Best visual quality and structural consistency
- **Diffusion:** More diverse outputs
- **AR:** Faster with parallel decoders

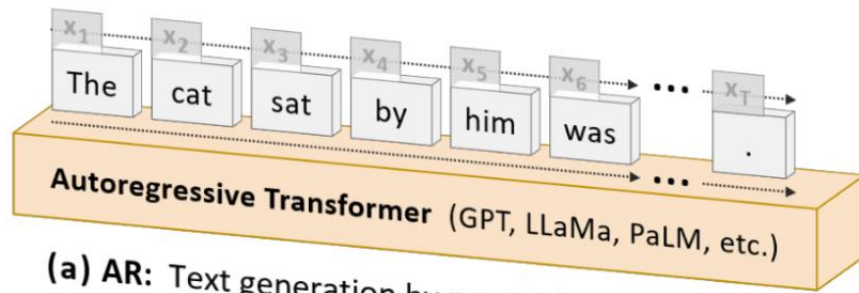
Типы архитектур

- **PixelAR / PixelCNN / PixelRNN**
 - Генерируют изображение пиксель за пикселем
 - Медленно генерируют и редко используются сейчас
- **VQ-AR (через токены)**
 - Изображение разбивают на токены
 - Потом Transformer генерирует эти токены один за другим

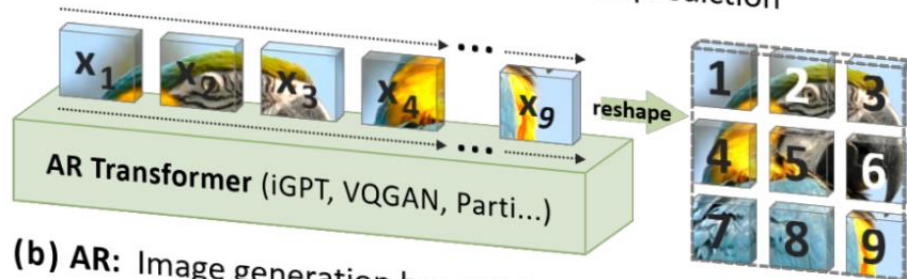
Пример: DALL·E 1, RQ-Transformer
- **Иерархический подход**
 - Сначала модель генерирует крупные “размытые” блоки
 - Потом в несколько шагов уточняет их содержимое

Coarse-to-fine иерархический подход

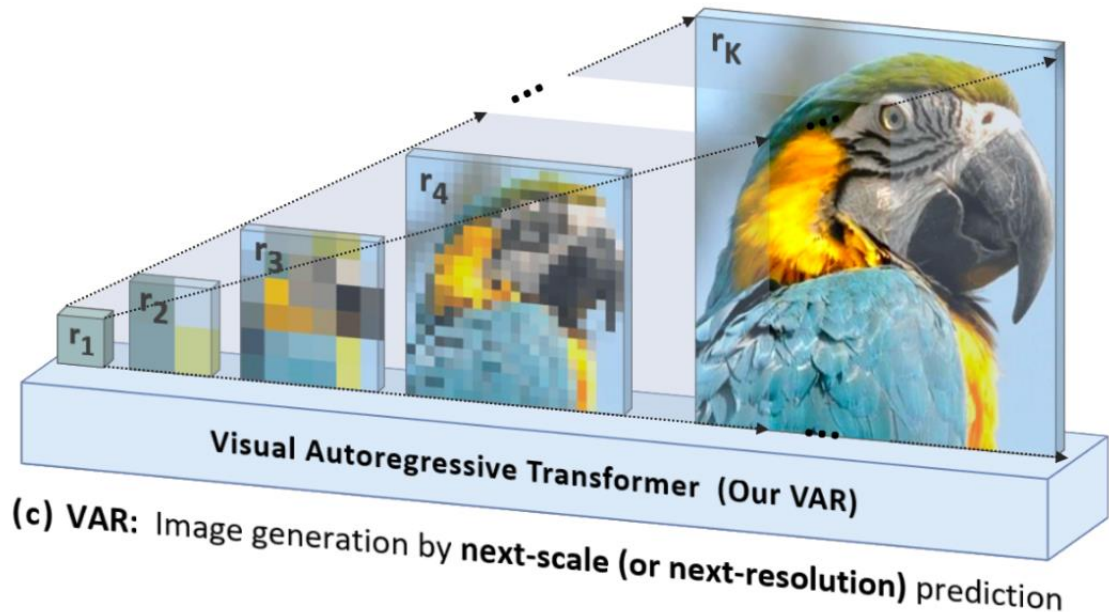
Three Different Autoregressive Generative Models



(a) AR: Text generation by **next-token** prediction



(b) AR: Image generation by **next-image-token** prediction



(c) VAR: Image generation by **next-scale (or next-resolution)** prediction

Практика по AR-моделям

<https://colab.research.google.com/drive/1EIsBxITcz3ZB0j-ZxN1gWD2dSQq6krPj?usp=sharing>