

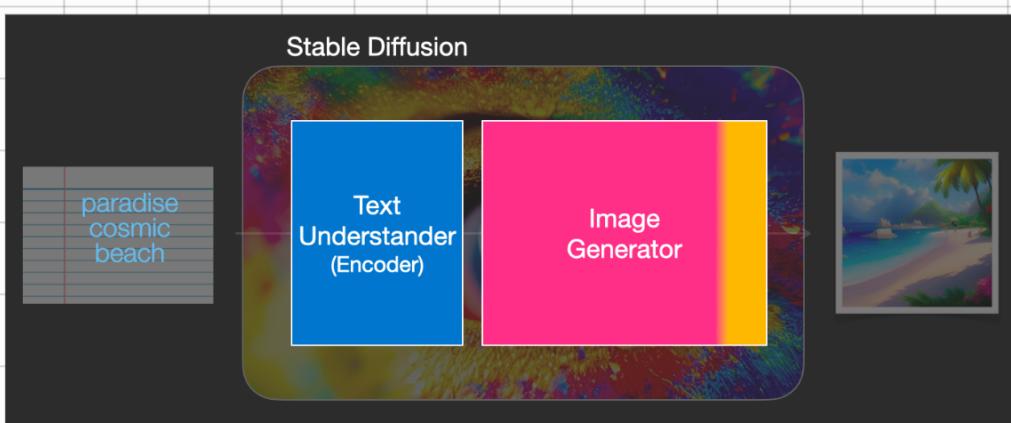
Простейшее представление работы Stable Diffusion:



Такие в качестве входных данных могут служить текст + изображение. То есть результатом отдаётся модифицированное соответствие тексту изображению:



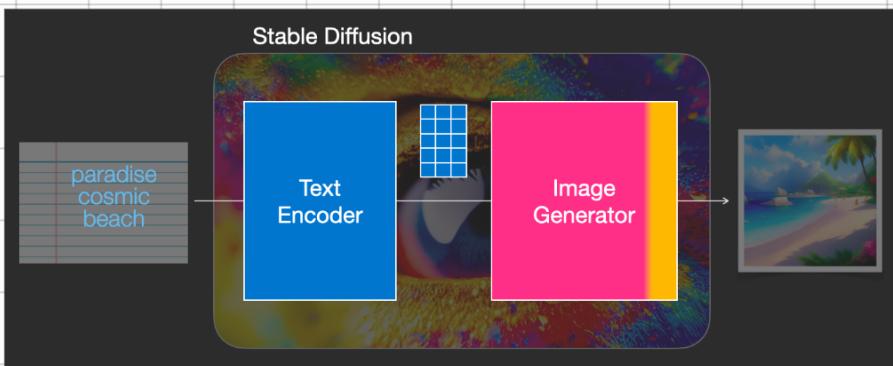
Первая составл. архитектура — текстовая интерпретирующая компонента, преобраз. текст. инф. в числ. предел.



Постепенно будем углублять в детали архитектура

Текстовый якоддр — специал. язюкова юзель на основе трансформера. Принимает входной текст и векторы списков чисел.

После данных инфодр. передается в **Image Generator**, который сам состоит из нескольких компонентов. Рассмотрим в 2 этапа.



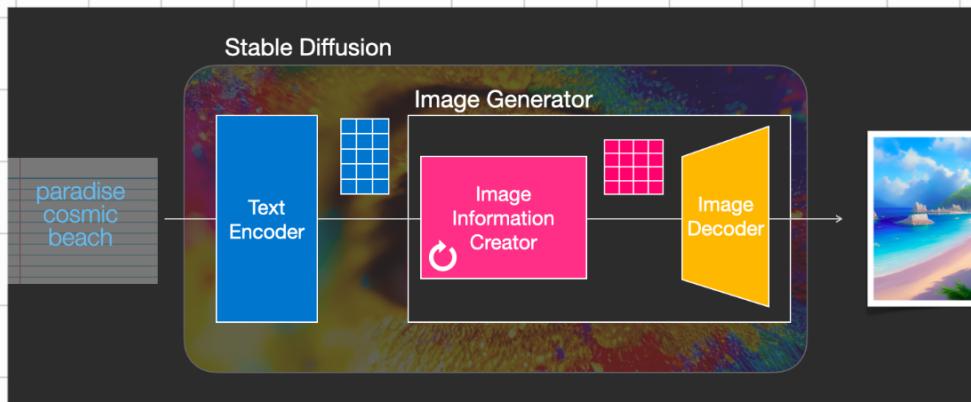
## 1. Image Information Creator

В интерфейсах и библиотеках Stable Diffusion есть соответствующий параметр **steps**, по умолчанию 50 или 100.

Работает в пространстве информации изображения.

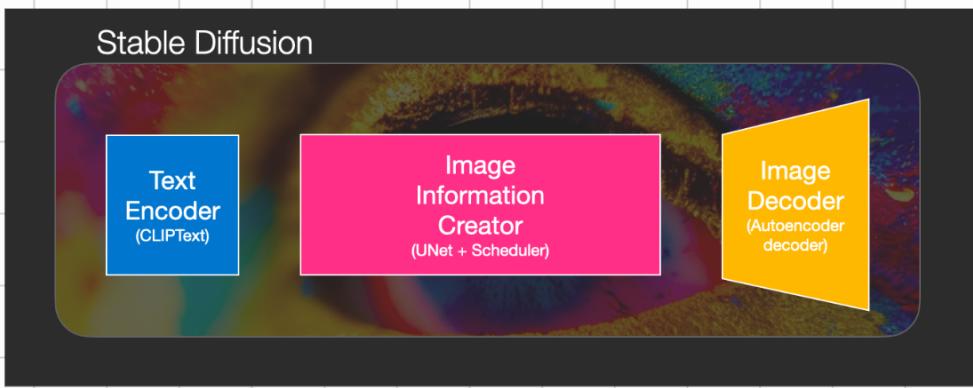
Технически состоит из UNet и алгоритма планирования.

„Диффузия“ описывает, что происходит в этой компоненте.  
Информация обрабатывается шаг за шагом, что приводит к генерации изображения (генодорри)



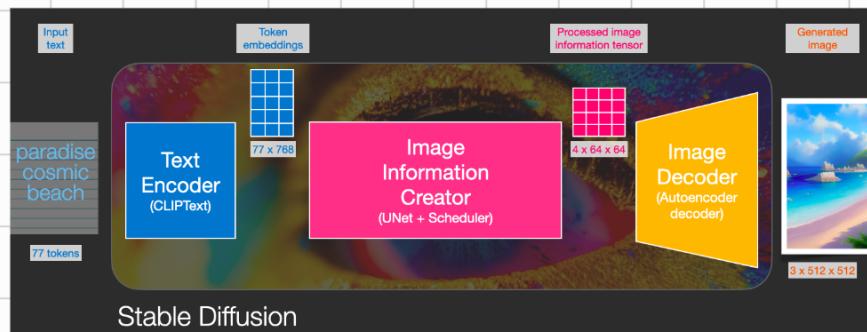
## 2. Image Decoder

Создает картинку на основе информации с предыдущей компонентой. Запускается один раз в конце процесса



### 3 основные составляющие Stable Diffusion:

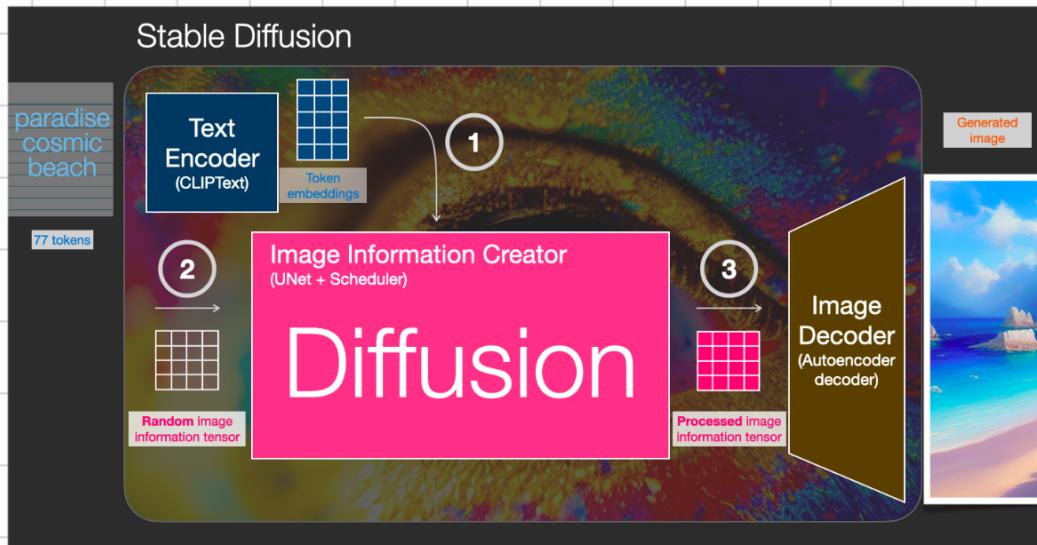
- **ClipText** - для кодирования  
 Вход: текст  
 Выход: 77 токен-эмбеддингов, каждый размер  $768 \times 1$
- **UNet + Scheduler** - для постепенной обработки/диффузии изр.  
 в 4-мерном пространстве.  
 Вход: текст. эмбеддинги и многомерный массив из шума  
 Выход: Обработанный информационный массив
- **Autoencoder Decoder** - рисует конечную картинку, используя  
 обработ. массив информации  
 Вход: массив информ.  $(1, 64, 64)$   
 Выход: регулируемое изображение  $(3, 512, 512) \rightarrow (\text{red/green/blue}, \text{width}, \text{height})$



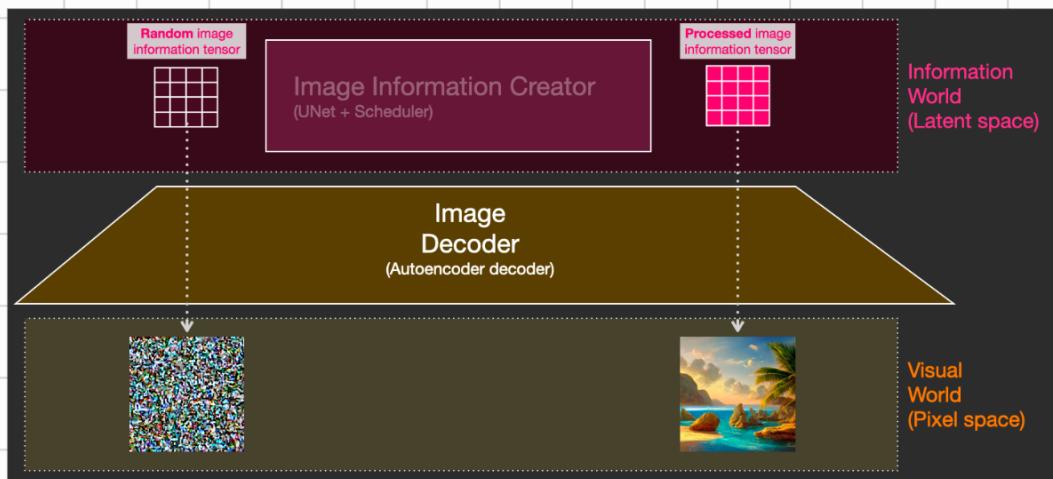
Что такое диффузия?

Диффузия - процесс создания информац. массива (в разной густоте),

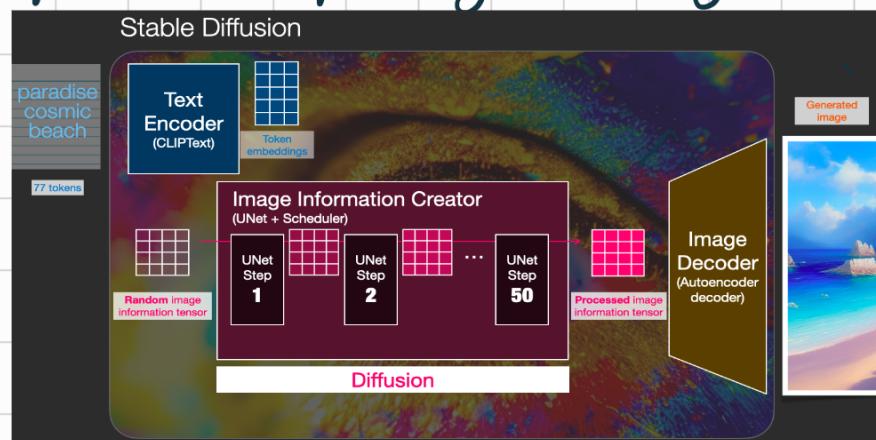
который используется для создания финальной картины сконструирована.

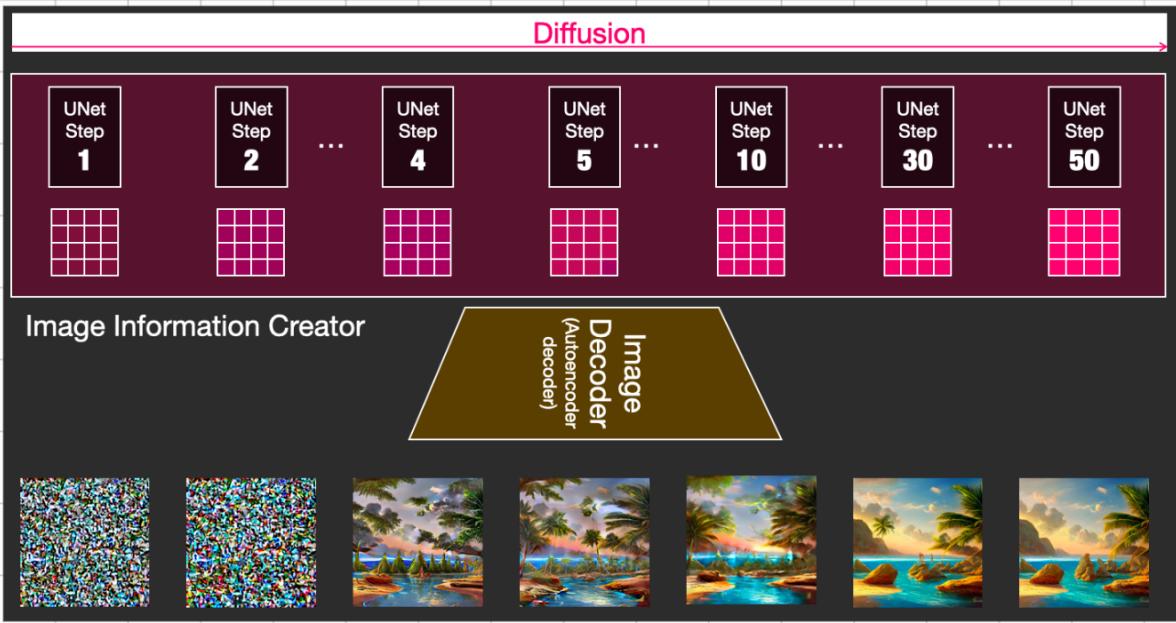


Этот процесс происходит пошагово. Каждый шаг добавляет больше релевантной информации. Следующий этап соответствует визуальному изображению



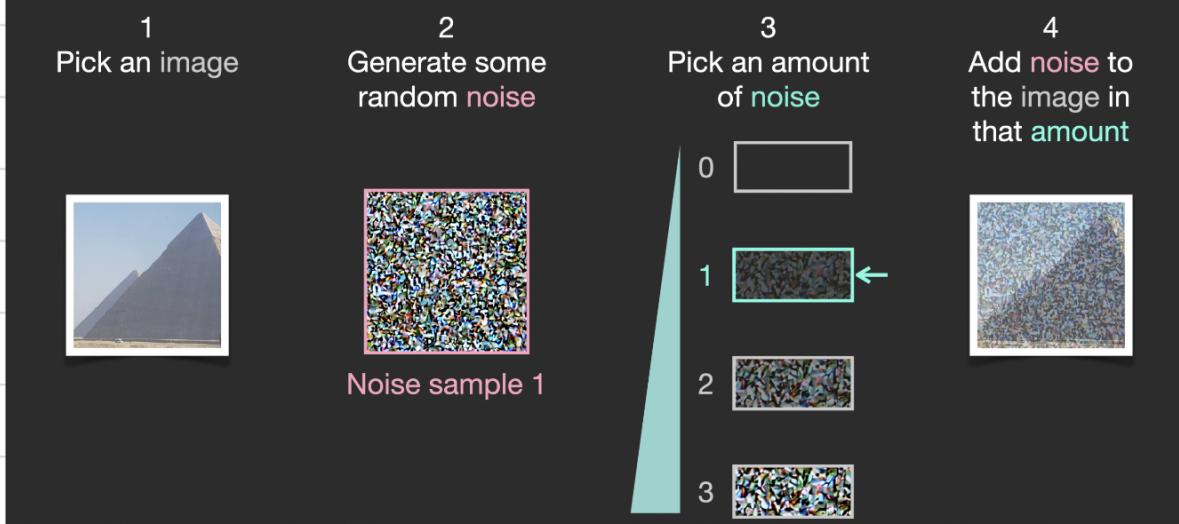
На каждом шаге входной массив преобразуется в другой массив, который лучше представляет входной текст и визуальные изображения, из которых обучалась модель.



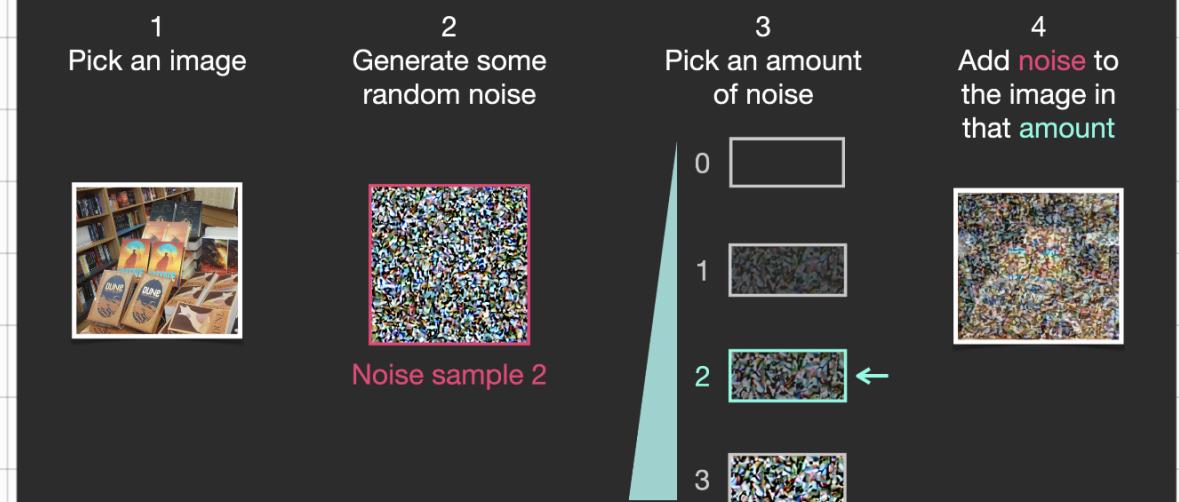


*Как падомаум гүрүүзүү?*

Training examples are created by generating **noise** and adding an **amount** of it to the images in the training dataset (forward diffusion)

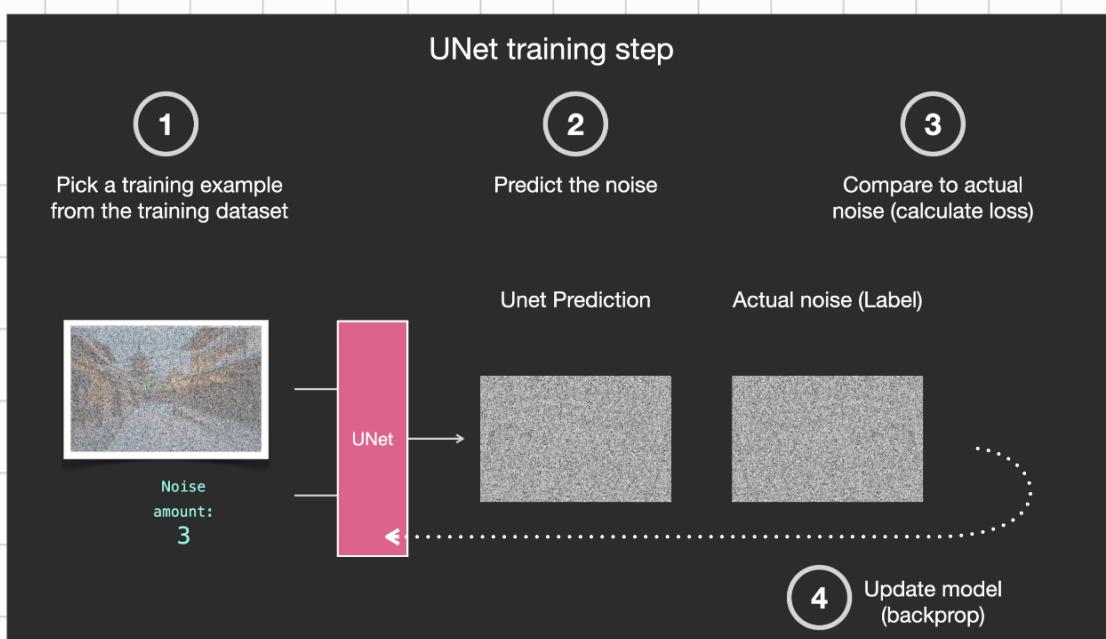


Generating a 2nd training example with a different image, **noise sample** and **noise amount** (forward diffusion)



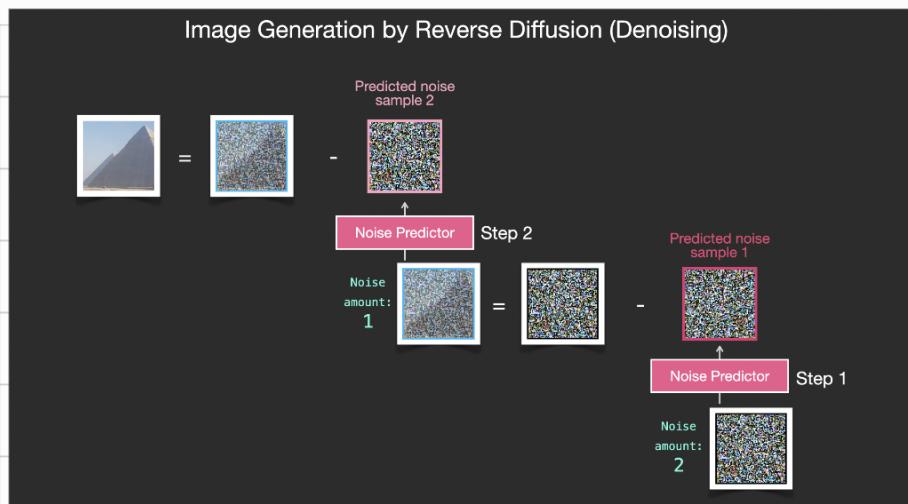


С помощью этого обучения предсказатель шума



Рисовакие картин удаление шума

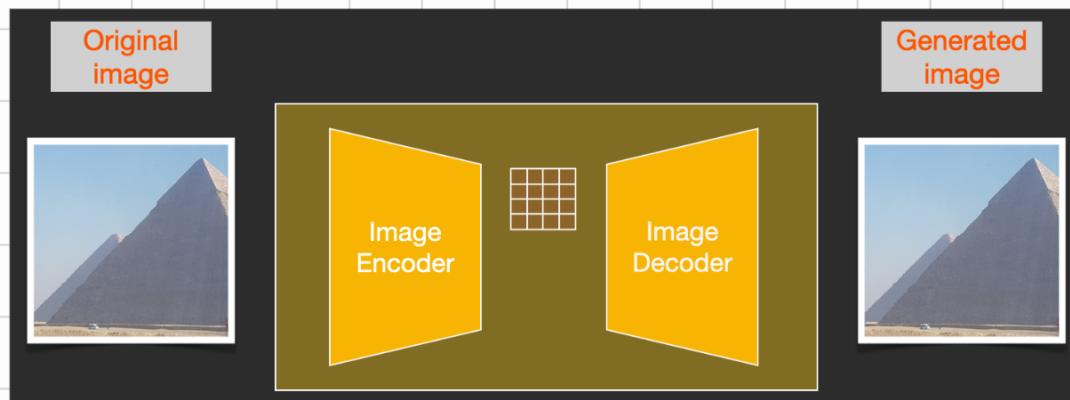
Обученный предсказатель может придать замутн. изображени. и номер шага удаления шума и способом предсказат. горушио шума. Видерка шума предсказ. таким образом, что если мы выберем его из изобраи., то получим изобраи., которое ближе к тем изобра., на которых обучалась модель



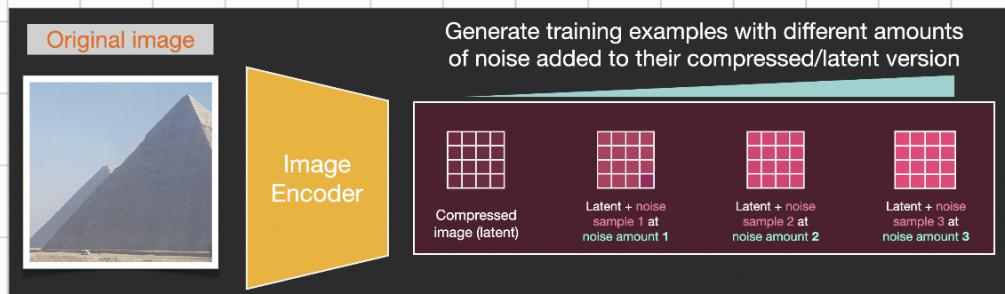
Дальше описываем генерацию изображения с учетом текст. выхода данных

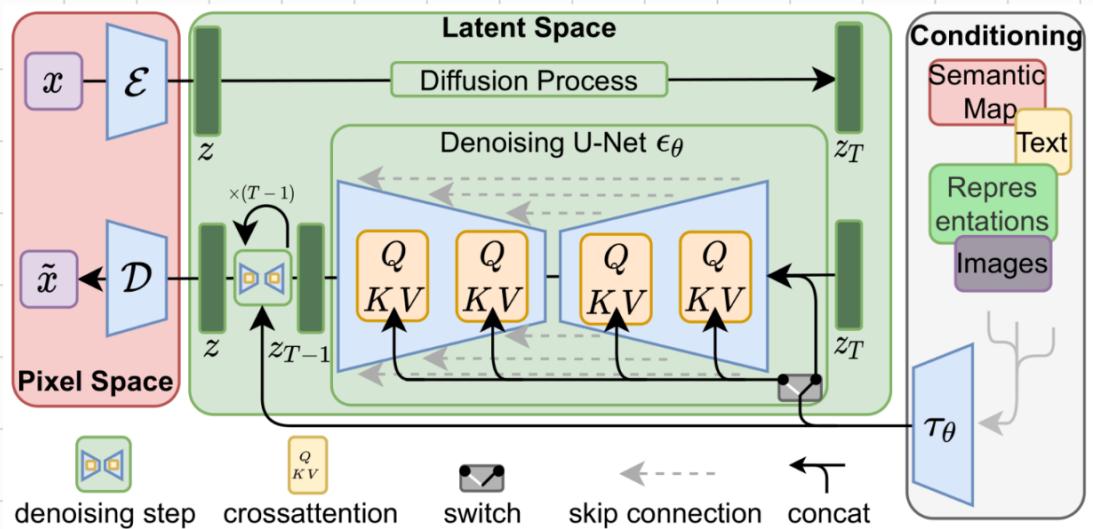
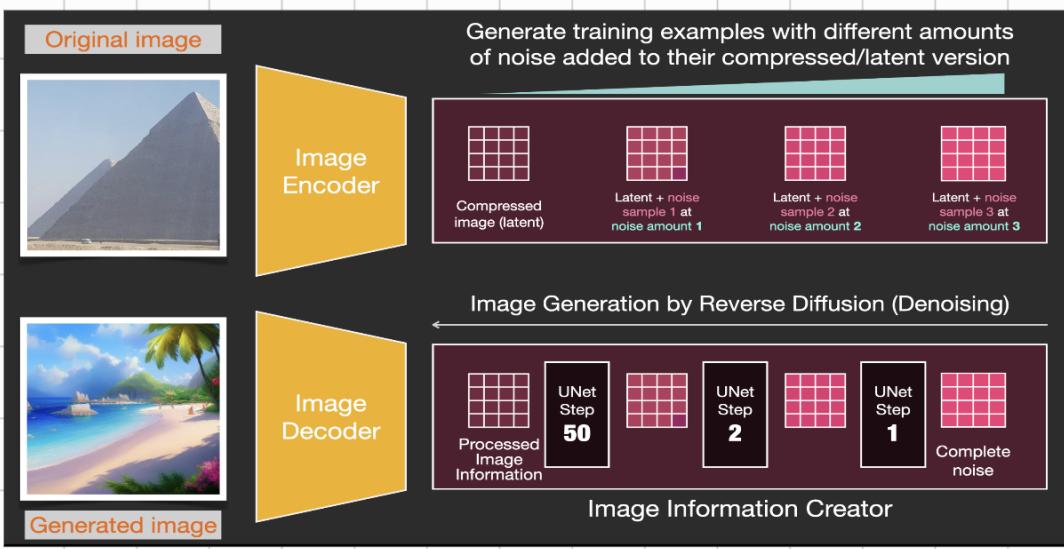
Диффузия на сжатых данных вместо пиксл. изобрз.

Автоматизир спишись изобрз. в latent. пространство, а затем восстанавливает, используя сжатую инф-ю в decoder



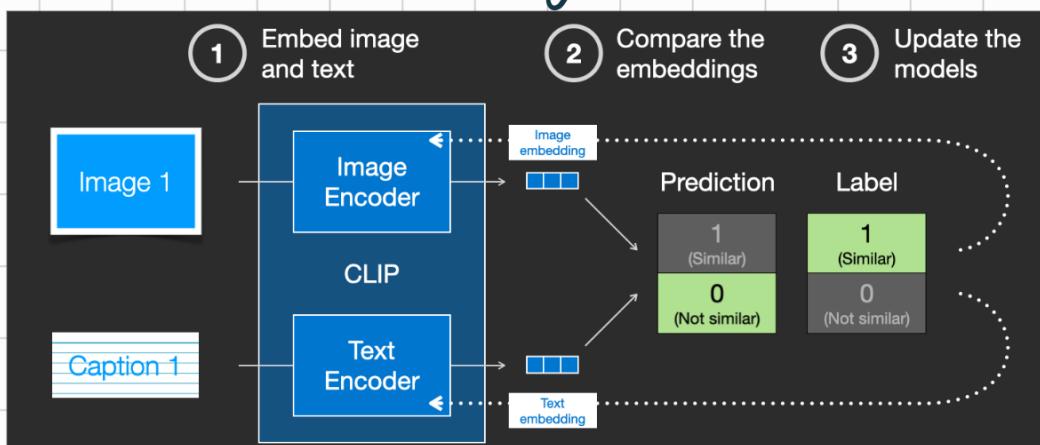
forward процесс теперь выполняется на сжатых latent. переменных + шум к ним. Предсказатель шум фактически учится предс. шум в сжатой представлении





CLIP

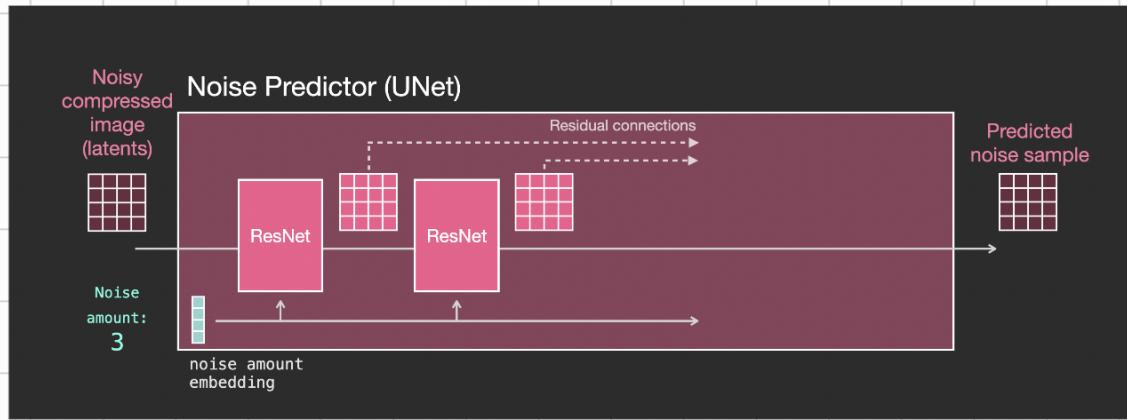
CLIP – комбинация энкодера изображений и энкодера текста. Полученное из них эмбеддинги сравниваются через cosine similarity. Цель обучения, чтобы например эмбеддинг изображения с надписью «собака» был близок



<sup>UNet</sup>

# Noise Predictor (без текста)

UNet - серия слоёв трансформирующих массив. Всего 5 слоёв прокидываются через residual connections.



# UNet Noise predictor с текстом

Ключевое изменение – добавление обработки текстового входа через attention layer после ResNet blocks

