



Факультет компьютерных наук

Искусственный
интеллект

Москва
2025

Generative Multimodal Models: Text to Image



План занятия

contrastive CLIP/CLAP → fusion → generation → MLLM

1. Почему diffusion победили в T2I
2. Diffusion basics: forward/noising → reverse/denoising
3. DDPM objective
4. Latent Diffusion → Stable Diffusion
5. Контролируемые ручки инференса: schedule, sampler/steps, guidance scale, seed, resolution
6. Как эта интуиция переносится на Text2Video



Почему диффузия?

Stable Diffusion XL → Latent **Diffusion** + U-Net, усиленный UNet и 2 text encoders

Stable Diffusion 3 / 3.5 → MMDiT (Multimodal **Diffusion** Transformer)

FLUX.1 → Rectified Flow Transformer.

Imagen 3 → latent **diffusion**

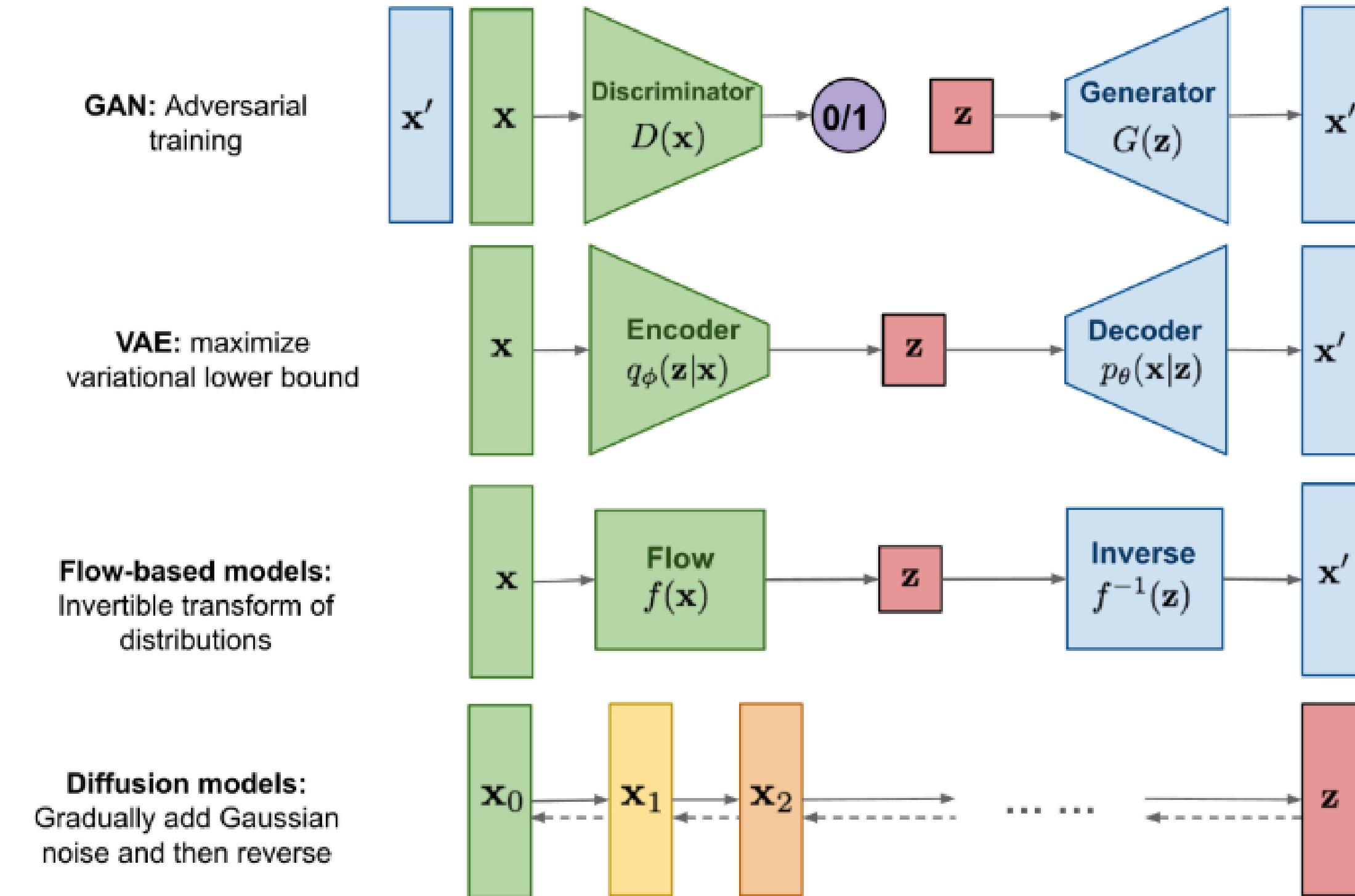
OpenAI 4o image generation → autoregressive image model

DALL·E 3 → детали базовой архитектуры не раскрываются, но DALL·E 2 используют **diffusion**-модели для декодера



Generative families: autoregressive vs VAE vs GAN vs diffusion

или почему diffusion победили по качеству



Forward process

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I})$$

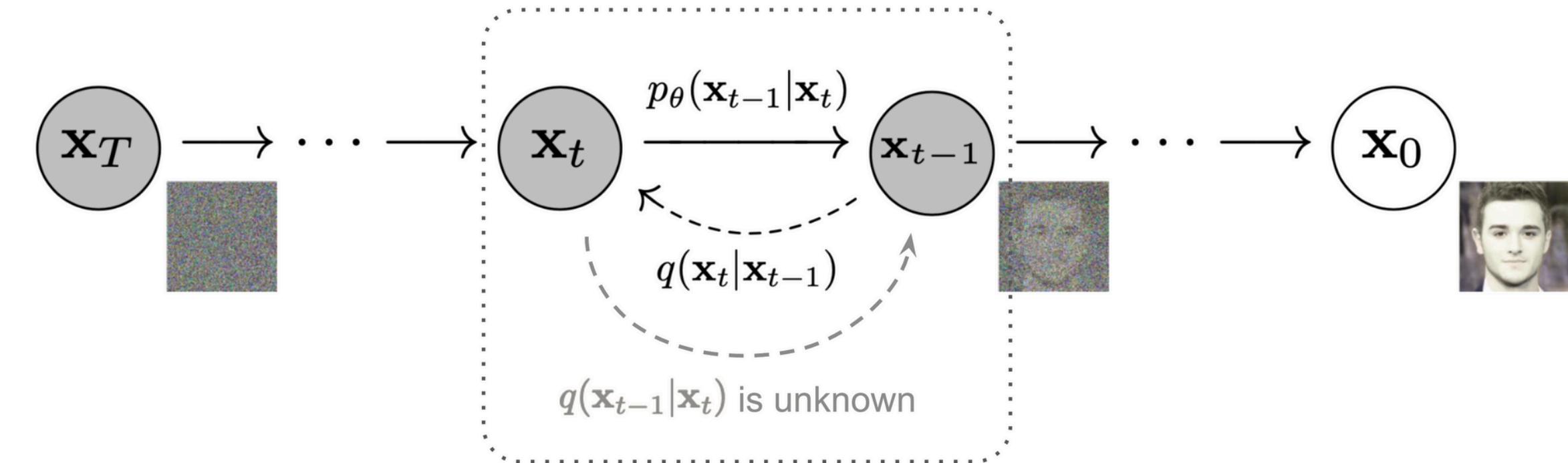
1. Мы масштабируем предыдущее состояние x_{t-1} с коэффициентом $\sqrt{1 - \beta_t}$.
2. Добавляем случайный шум $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, умноженный на β_t .
3. В результате каждое следующее изображение становится немного более зашумленным, но остаётся похожим на предыдущее.

На каждом шаге можно записать это так:

$$\mathbf{x}_t = \sqrt{1 - \beta_t} \mathbf{x}_{t-1} + \sqrt{\beta_t} \epsilon$$

Где $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ — случайный шум.

Use variational lower bound





Reverse process: учим модель снимать шум

Хотим аппроксимировать обратный процесс:

$$p_{\theta}(x_{t-1} \mid x_t) \approx \mathcal{N}(\mu_{\theta}(x_t, t), \Sigma_t)$$

и пошагово идти от $x_T \sim \mathcal{N}(0, I)$ к x_0 .

Практический параметризационный трюк (DDPM): вместо μ_{θ} учим предсказывать шум $\varepsilon_{\theta}(x_t, t)$.

Что такое зашумление (нужна для обучения и семплинга):

$$x_t = \sqrt{\bar{\alpha}_t}, x_0 + \sqrt{1 - \bar{\alpha}_t}, \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I)$$

Conditional generation (T2I): $\varepsilon_{\theta}(x_t, t, \text{text})$ - условие инжектим через cross-attention

Семплинг (идея): для $t = T \dots 1$ делаем шаг денойза и получаем всё более структурное изображение.



DDPM objective

что оптимизируем при обучении

$$\alpha_t = 1 - \beta_t, \bar{\alpha}_t = \prod_{s=1}^t \alpha_s$$

Сэмплируем случайный шаг t и шум ε

Собираем x_t по формуле зашумления

Учим модель восстанавливать ε из x_t

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t, x_0, \varepsilon} [||\varepsilon - \varepsilon_\theta(x_t, t, c)||^2]$$

c = conditioning (например, текст), одна цель \rightarrow много способов сэмплинга

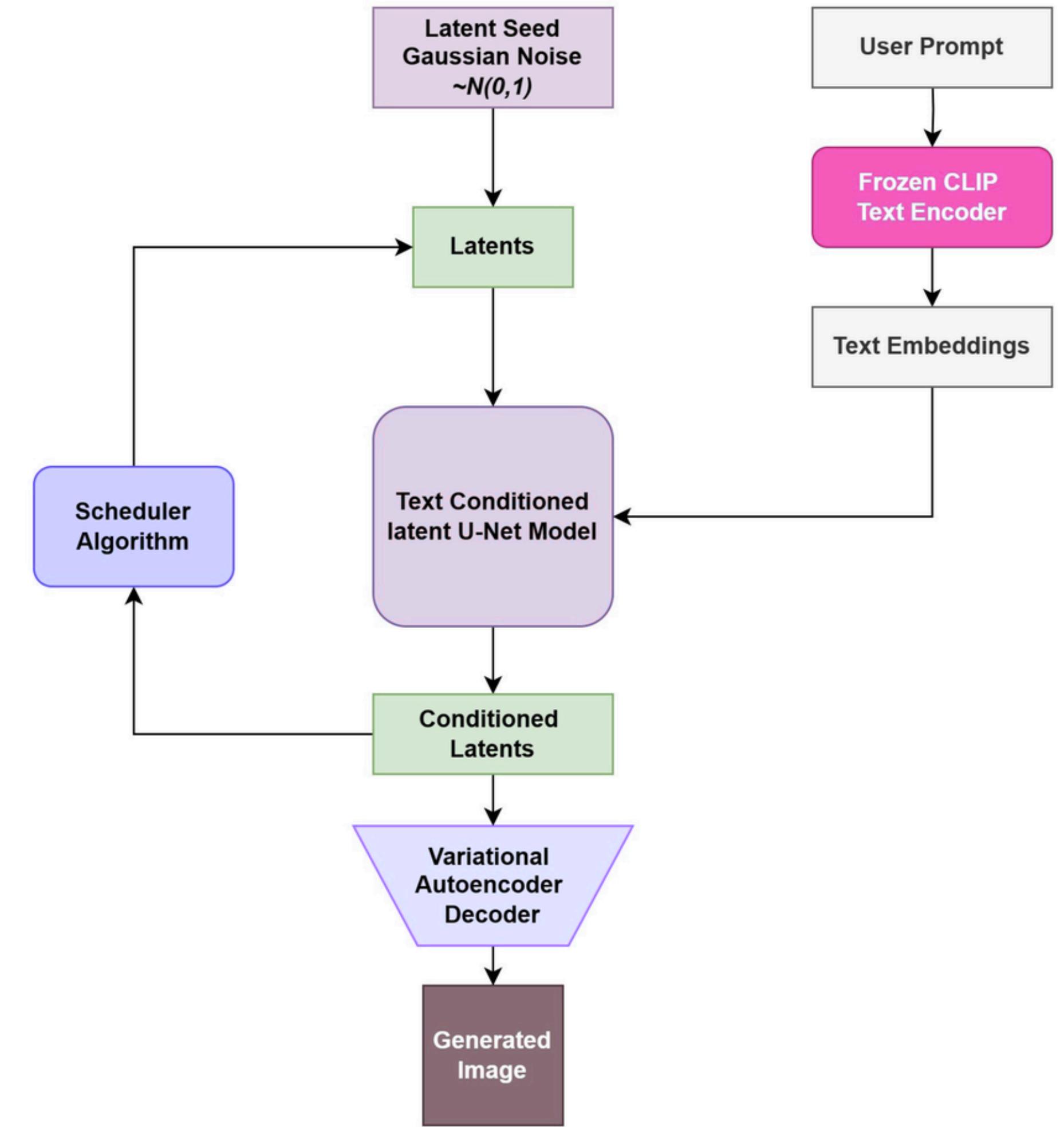
Denoising Diffusion Probabilistic Models

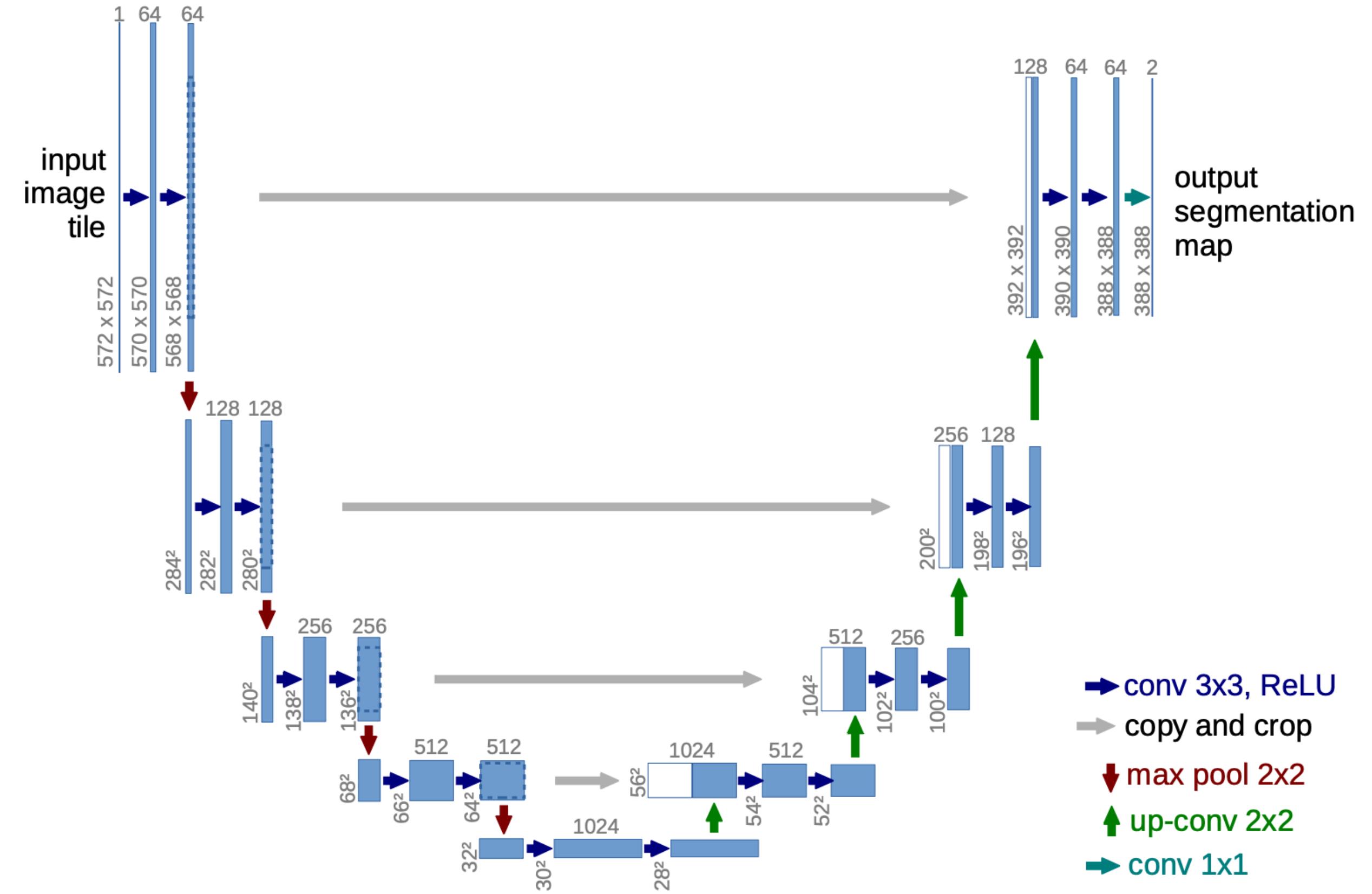


Ручки инференса

- **Noise schedule** (β_t , α_t): влияет на сложность обратного процесса и качество семплинга.
- **Sampler + число шагов**: больше шагов → обычно лучше качество, но медленнее. DDIM дает более быстрый семплинг при той же обучающей цели (trade-off скорость/качество).
- **Guidance scale (CFG)**: увеличивает следование промпту, но может снижать разнообразие/вводить артефакты при слишком больших значениях.
- **Seed и resolution/aspect ratio**

Denoising Diffusion Probabilistic Models

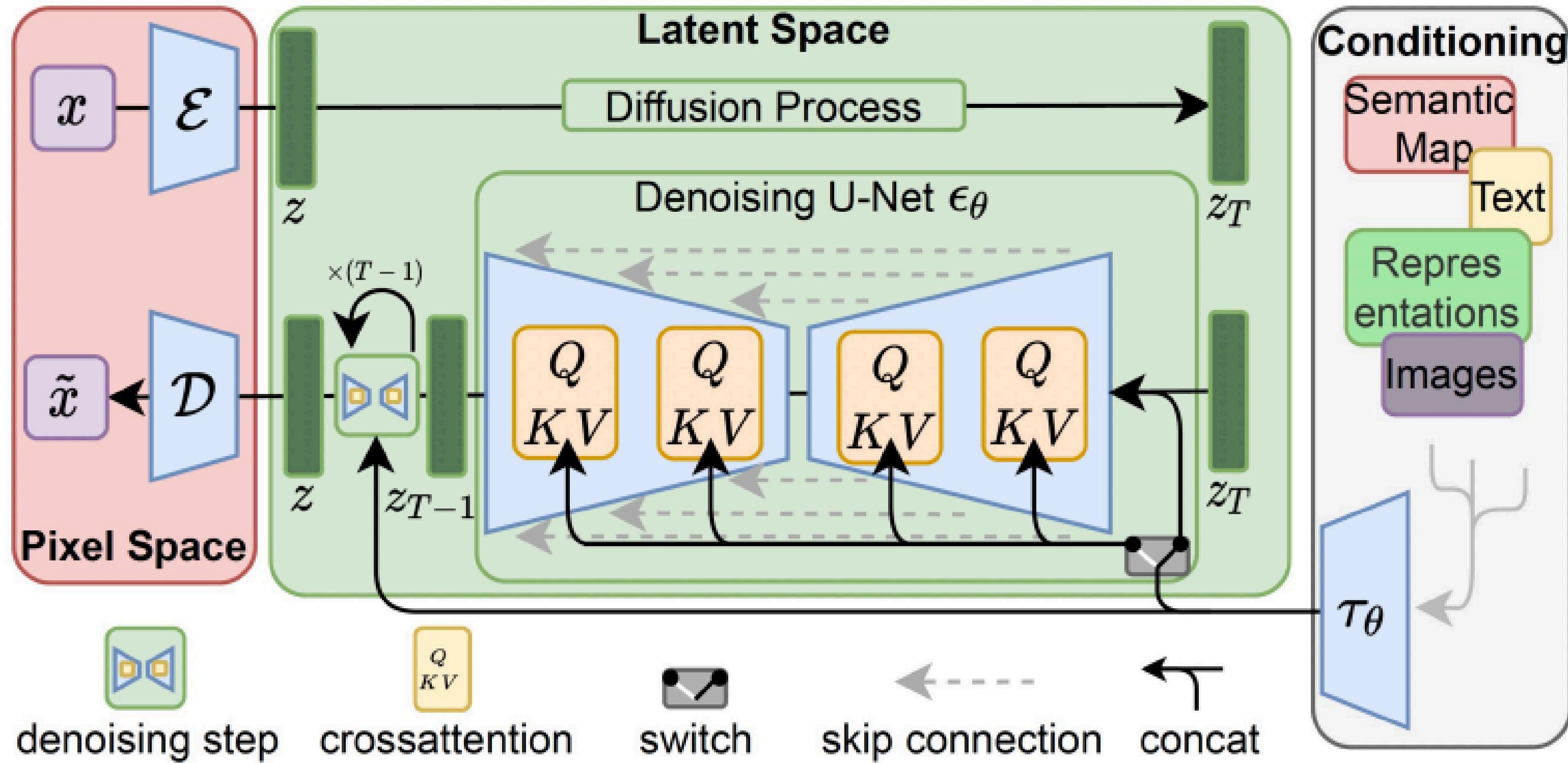




The U-net architecture

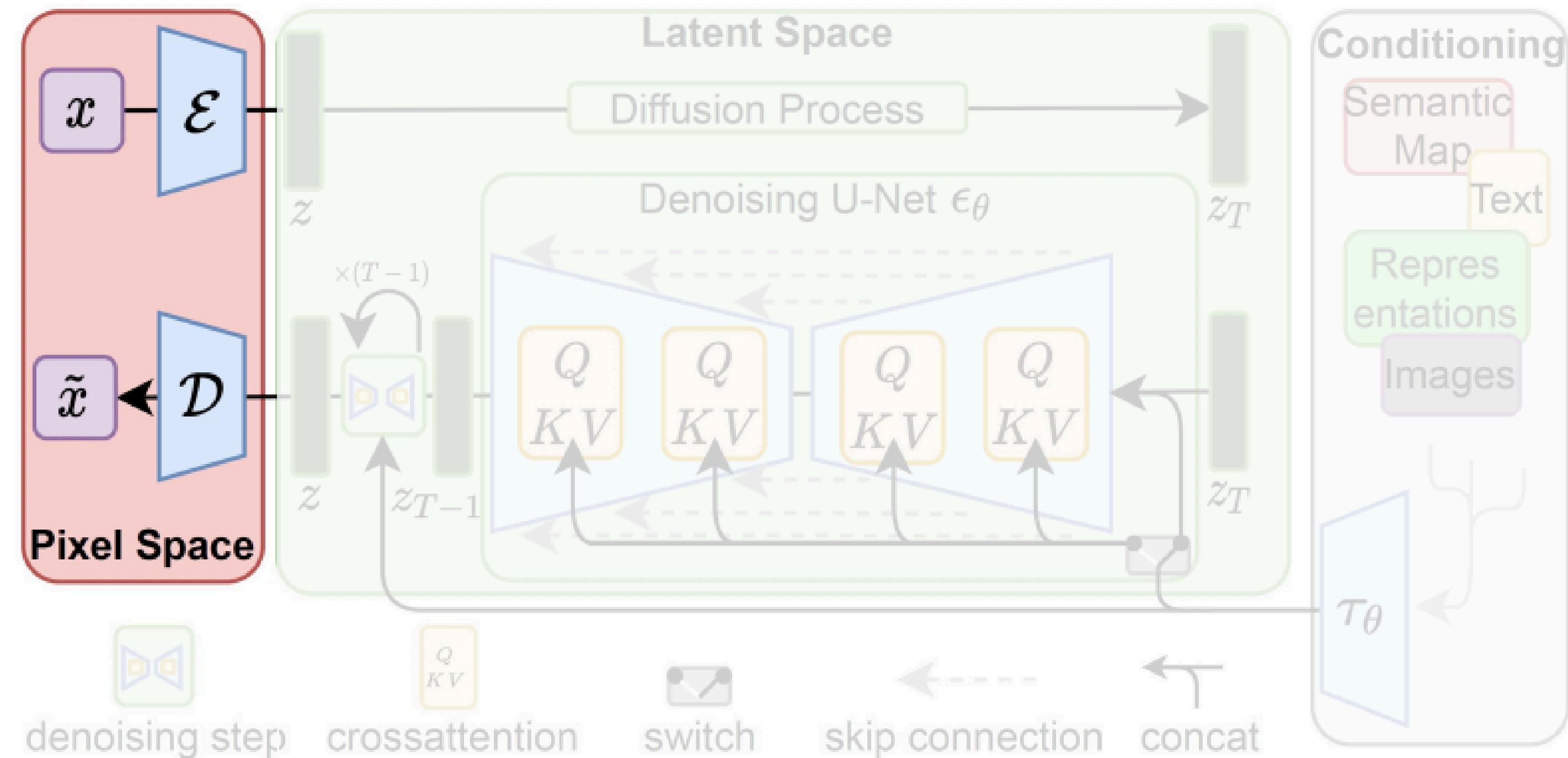


Latent Diffusion → Stable Diffusion: компоненты и dataflow



High-Resolution Image Synthesis with Latent Diffusion Models

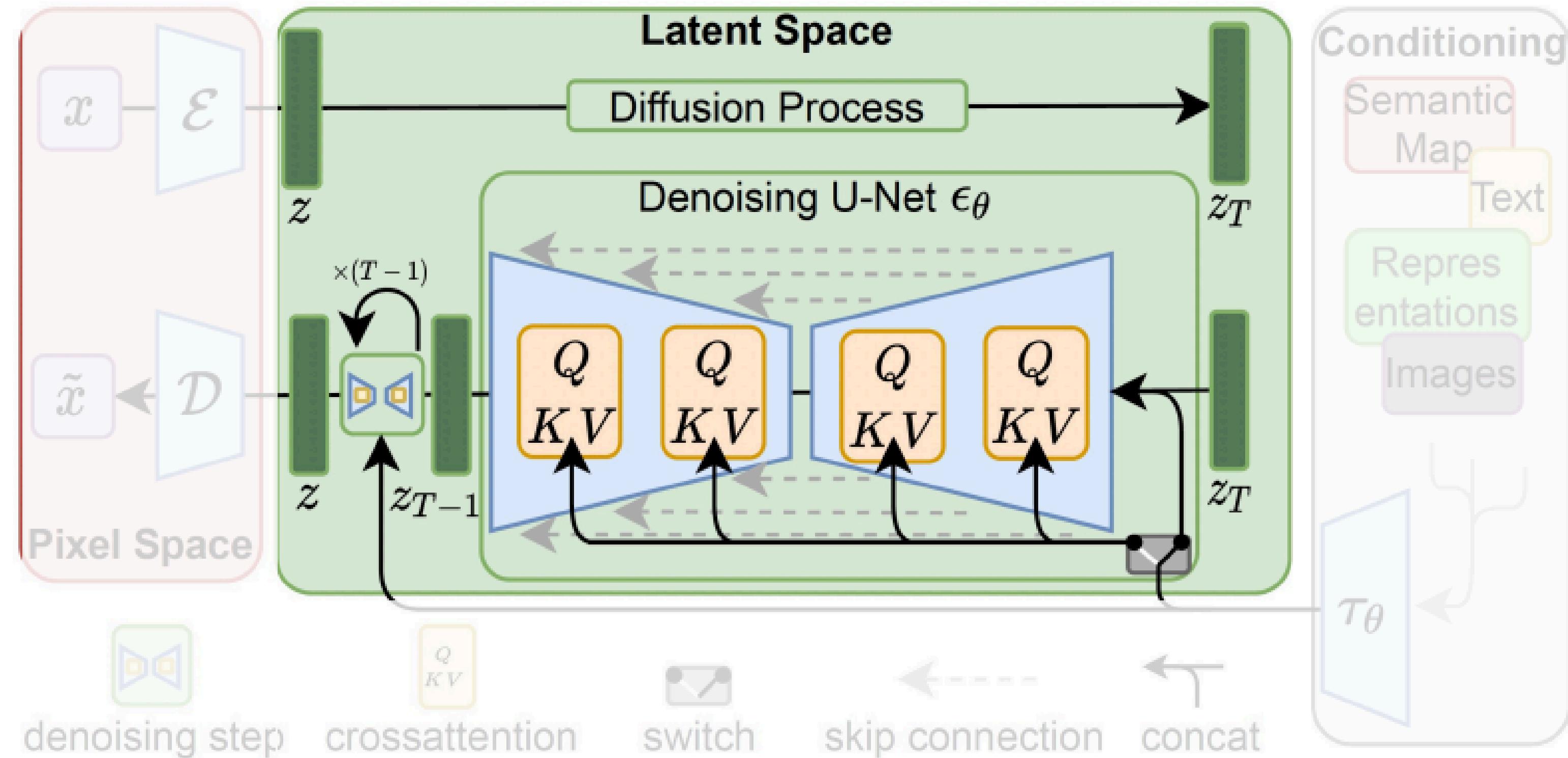
Компоненты LDM: VAE (encode/decode)



High-Resolution Image Synthesis with Latent Diffusion Models



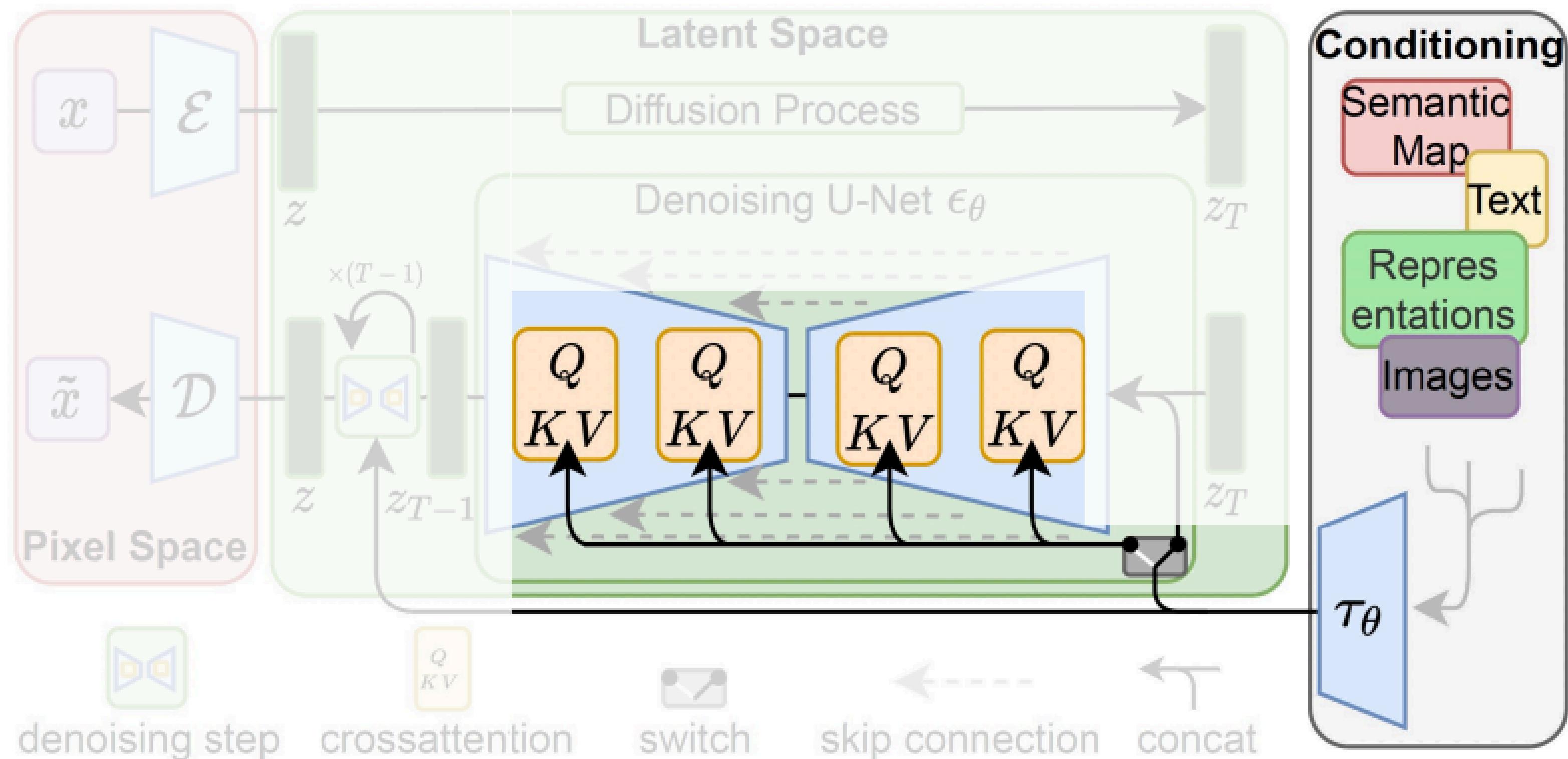
Компоненты LDM: U-Net denoiser



High-Resolution Image Synthesis with Latent Diffusion Models



Компоненты LDM: text encoder



High-Resolution Image Synthesis with Latent Diffusion Models



Версии Stable Diffusion

- SD 1.5: очень распространённая лёгкая модель (U-Net $\approx 860M$ параметров), огромная поддержка комьюнити (LoRA, кастомные чекпоинты)
- SDXL: та же общая идея latent diffusion, но крупнее: U-Net \sim в 3 раза больше, добавлен второй текстовый энкодер (плюс conditioning по crop/size)
- SD 3.0: вместо U-Net - Transformer-backbone (MMDiT, Multimodal Diffusion Transformer)
- Данные обучения: исходная Stable Diffusion обучалась на 512×512 изображениях из подмножества LAION-5B (text-image пары)

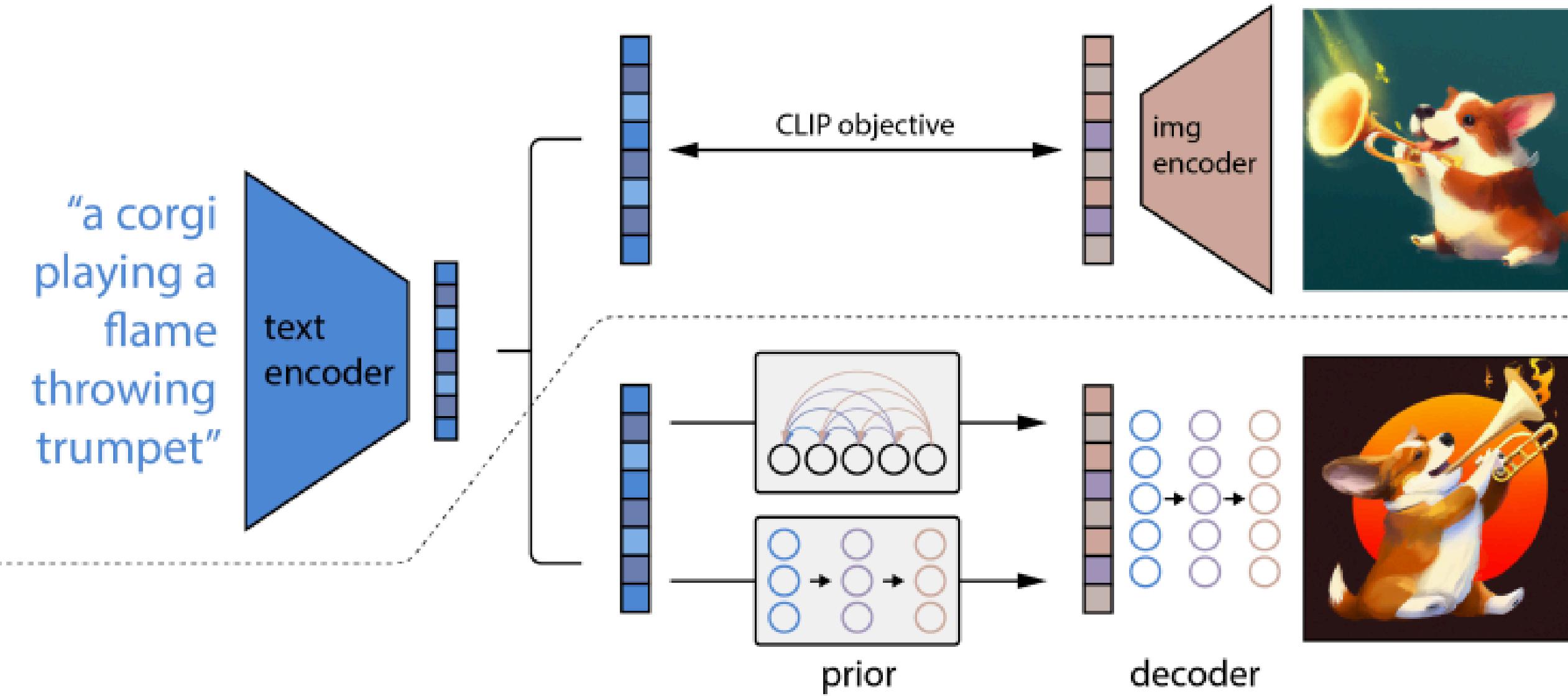


Pipeline	Supported tasks	Space
StableDiffusion	text-to-image	Hugging Face Spaces
StableDiffusionImg2Img	image-to-image	Hugging Face Spaces
StableDiffusionInpaint	inpainting	Hugging Face Spaces
StableDiffusionDepth2Img	depth-to-image	Hugging Face Spaces
StableDiffusionImageVariation	image variation	Hugging Face Spaces
StableDiffusionPipelineSafe	filtered text-to-image	Hugging Face Spaces
StableDiffusion2	text-to-image, inpainting, depth-to-image, super-resolution	Hugging Face Spaces
StableDiffusionXL	text-to-image, image-to-image	Hugging Face Spaces
StableDiffusionLatentUpscale	super-resolution	Hugging Face Spaces
StableDiffusionUpscale	super-resolution	
StableDiffusionLDM3D	text-to-rgb, text-to-depth, text-to-pano	Hugging Face Spaces
StableDiffusionUpscaleLDM3D	ldm3d super-resolution	

[Stable Diffusion pipeline](#)



UnCLIP



$$L_{\text{prior}} = \mathbb{E}_{t \sim [1, T], z_i^{(t)} \sim q_t} [\|f_\theta(z_i^{(t)}, t, y) - z_i\|^2]$$

Hierarchical Text-Conditional Image Generation with CLIP Latents



Не одним Stable Diffusion едины

- DiT/PixArt-a (diffusion-transformer)
- DeepFloyd IE (cascaded pixel diffusion)
- FLUX.1 (rectified flow transformer)
- Parti (autoregressive seq2seq)
- Imagen 3 (latent diffusion)
- Muse (masked generative transformer)



Imagen 3 - latent diffusion от Google

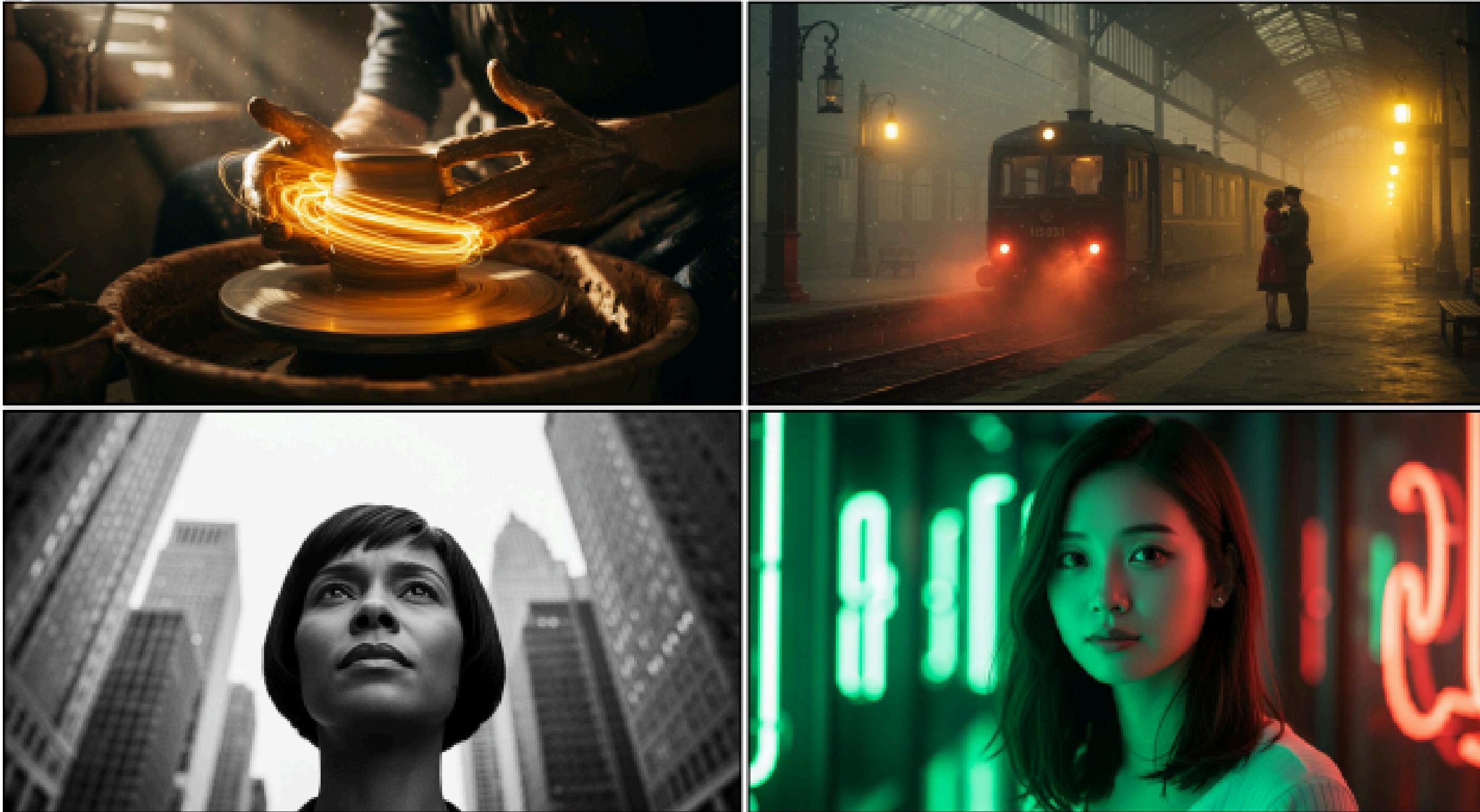


Figure 15 | Qualitative Results showcasing Imagen 3-002's capabilities.

Imagen 3



Diffusion Transformers

архитектура как у LLM, но для диффузии



Figure 1. Diffusion models with transformer backbones achieve state-of-the-art image quality. We show selected samples from two of our class-conditional DiT-XL/2 models trained on ImageNet at 512×512 and 256×256 resolution, respectively.

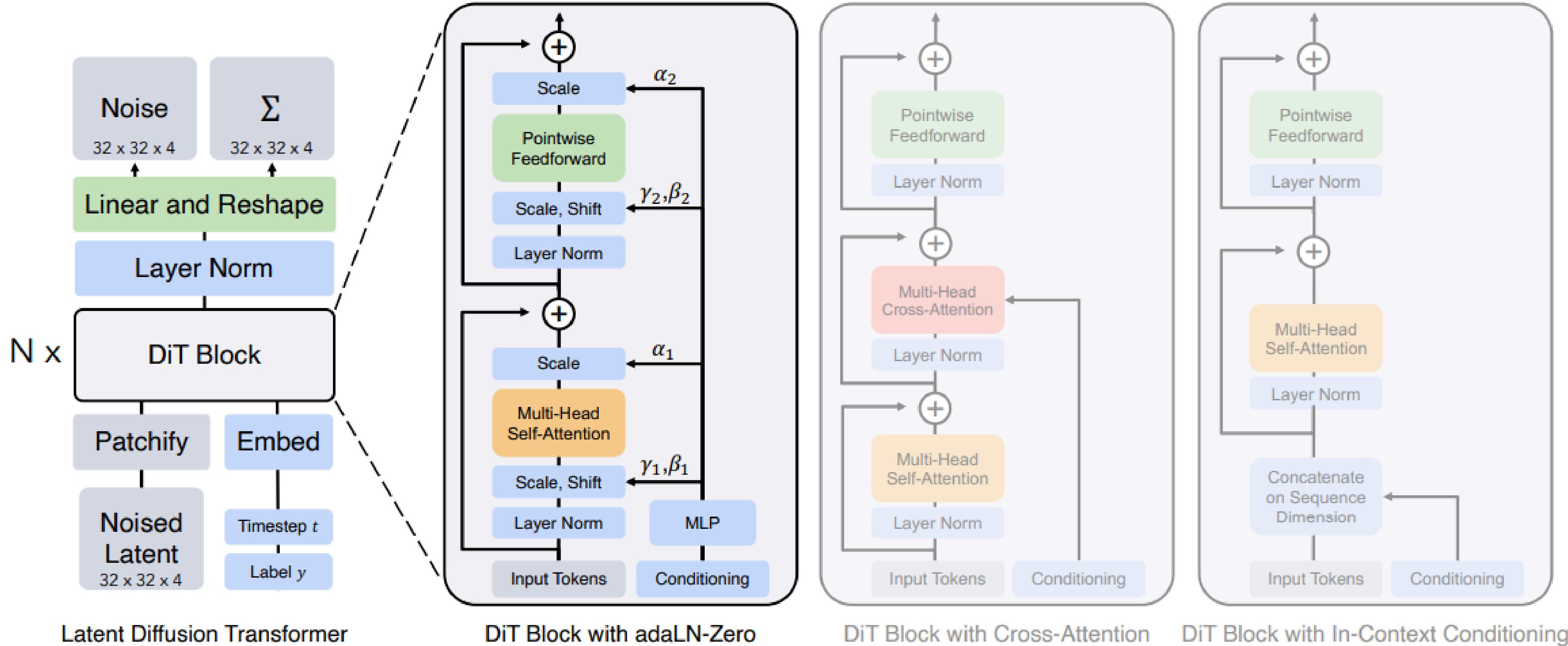


Figure 14. Uncurated 512×512 DiT-XL/2 samples.
Classifier-free guidance scale = 4.0
Class label = “arctic wolf” (270)

Scalable Diffusion Models with Transformers



DiT заменяет U-Net на Transformer в latent patches

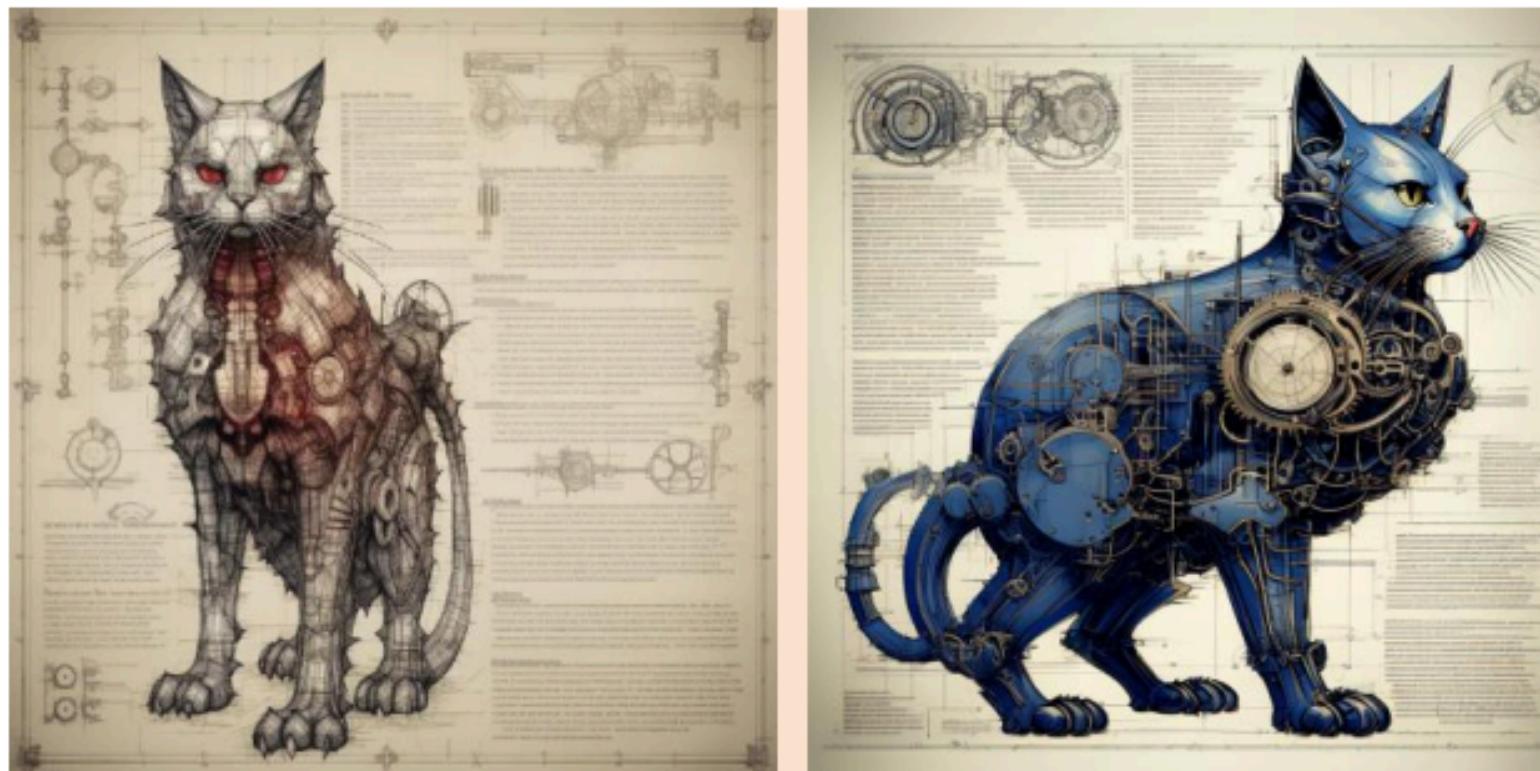


Scalable Diffusion Models with Transformers



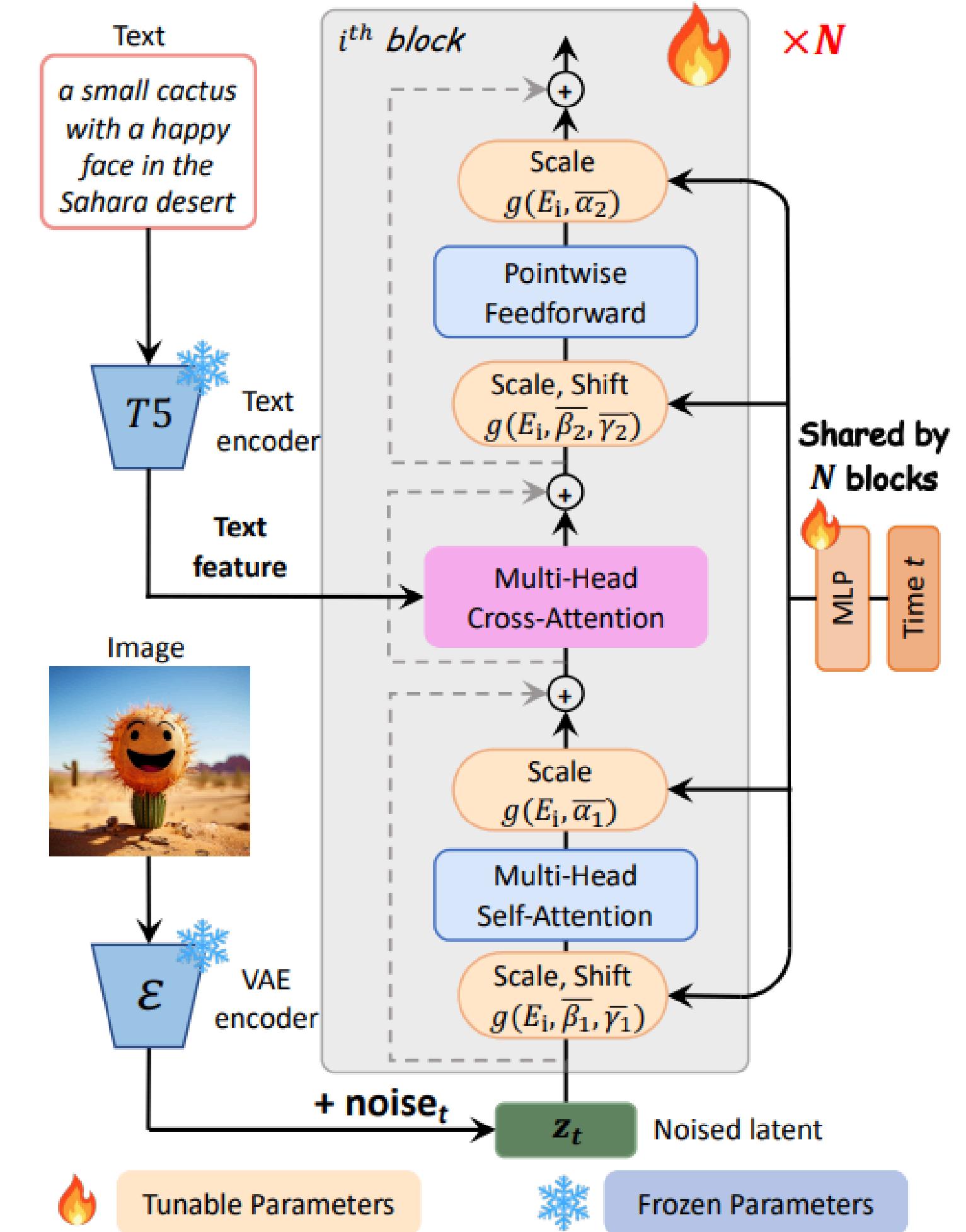
PixArt-a

- T2I diffusion-transformer, заявляет качество конкурентным SDXL/Imagen
- популярен как open-альтернатива SD-стеку

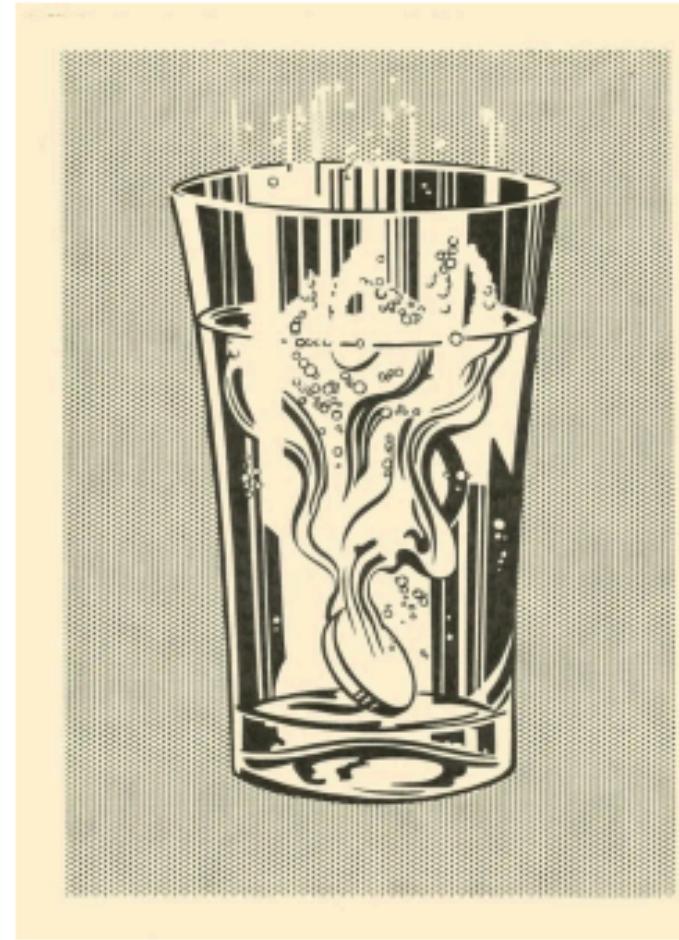


poster of a mechanical cat, technical Schematics viewed from front and side view on light white blueprint paper, illustration drafting style, illustration, typography, conceptual art, dark fantasy steampunk, cinematic, dark fantasy.

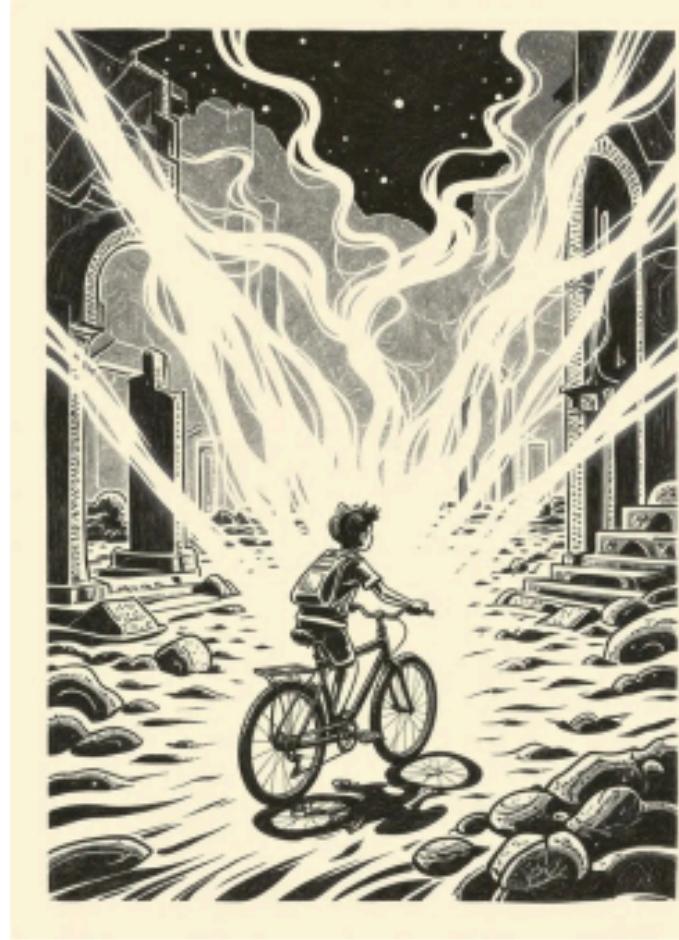
PixArt-a: Fast Training of Diffusion Transformer for Photorealistic Text-to-Image Synthesis



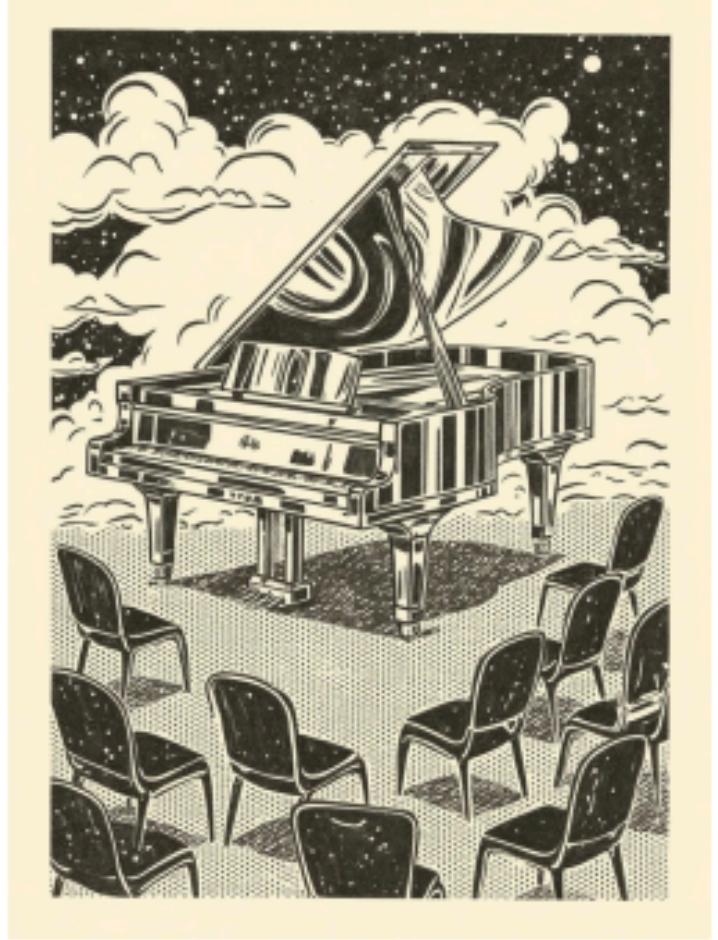
R FLUX.1 - rectified flow transformer B latent space



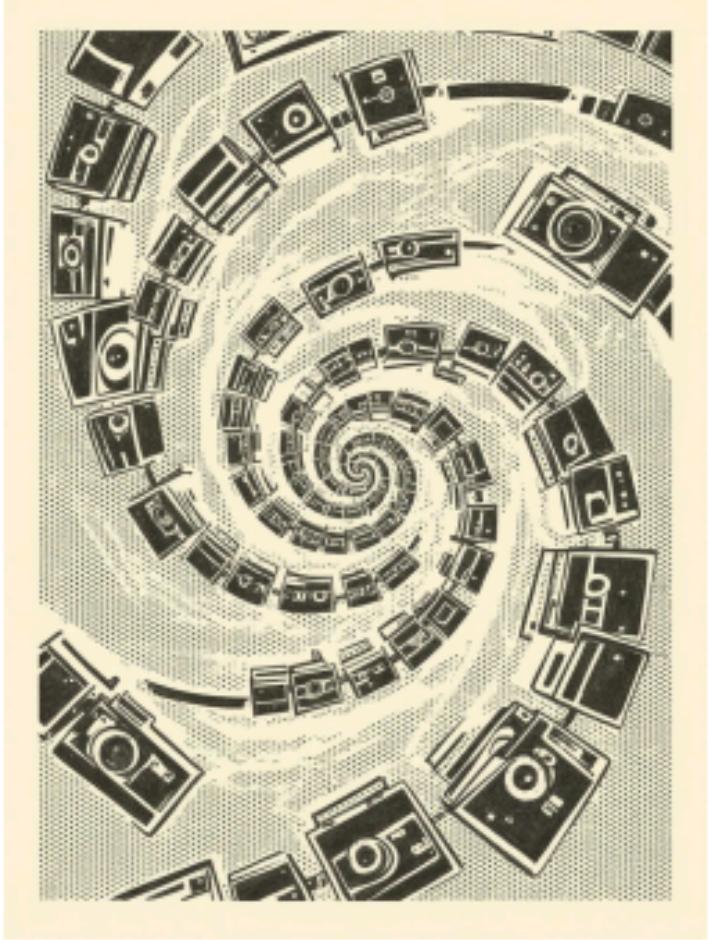
(a) Input image



(b) “*Using this style, a kid on a bicycles rolls through desert ruins, spotlights scanning ancient scrolls projected as holographic sandstorms.*”



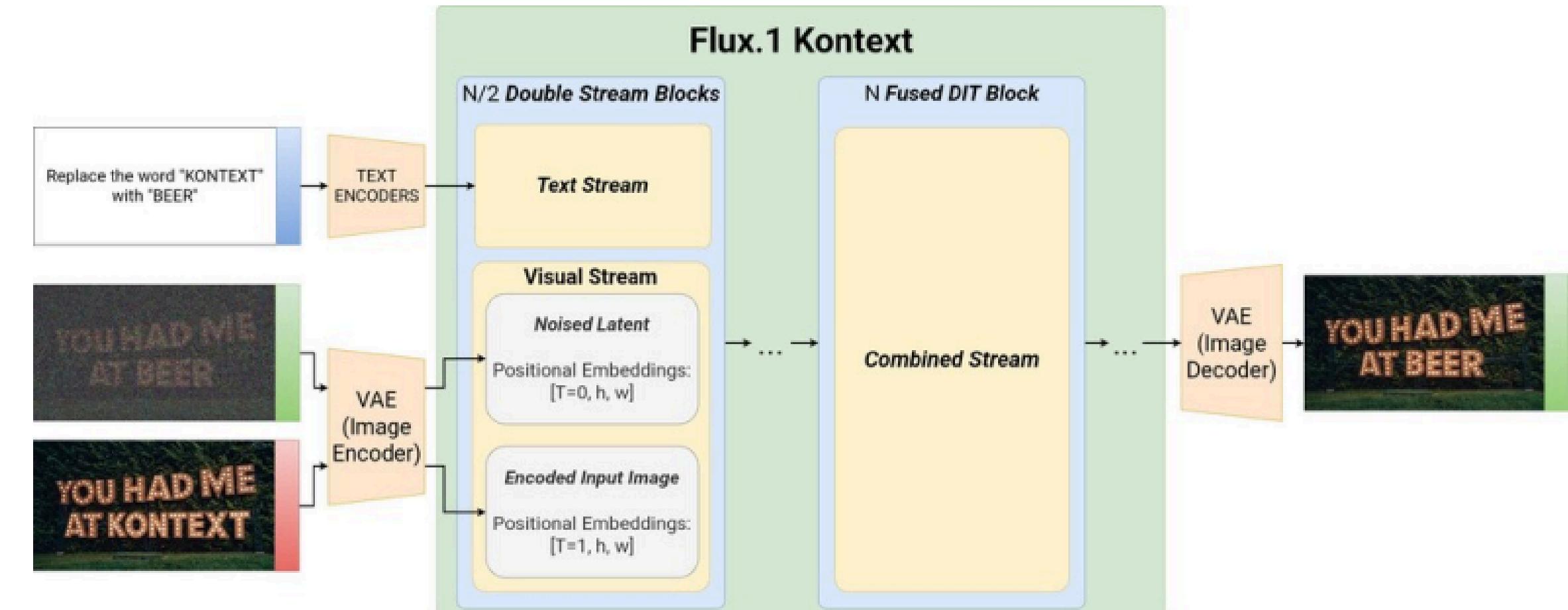
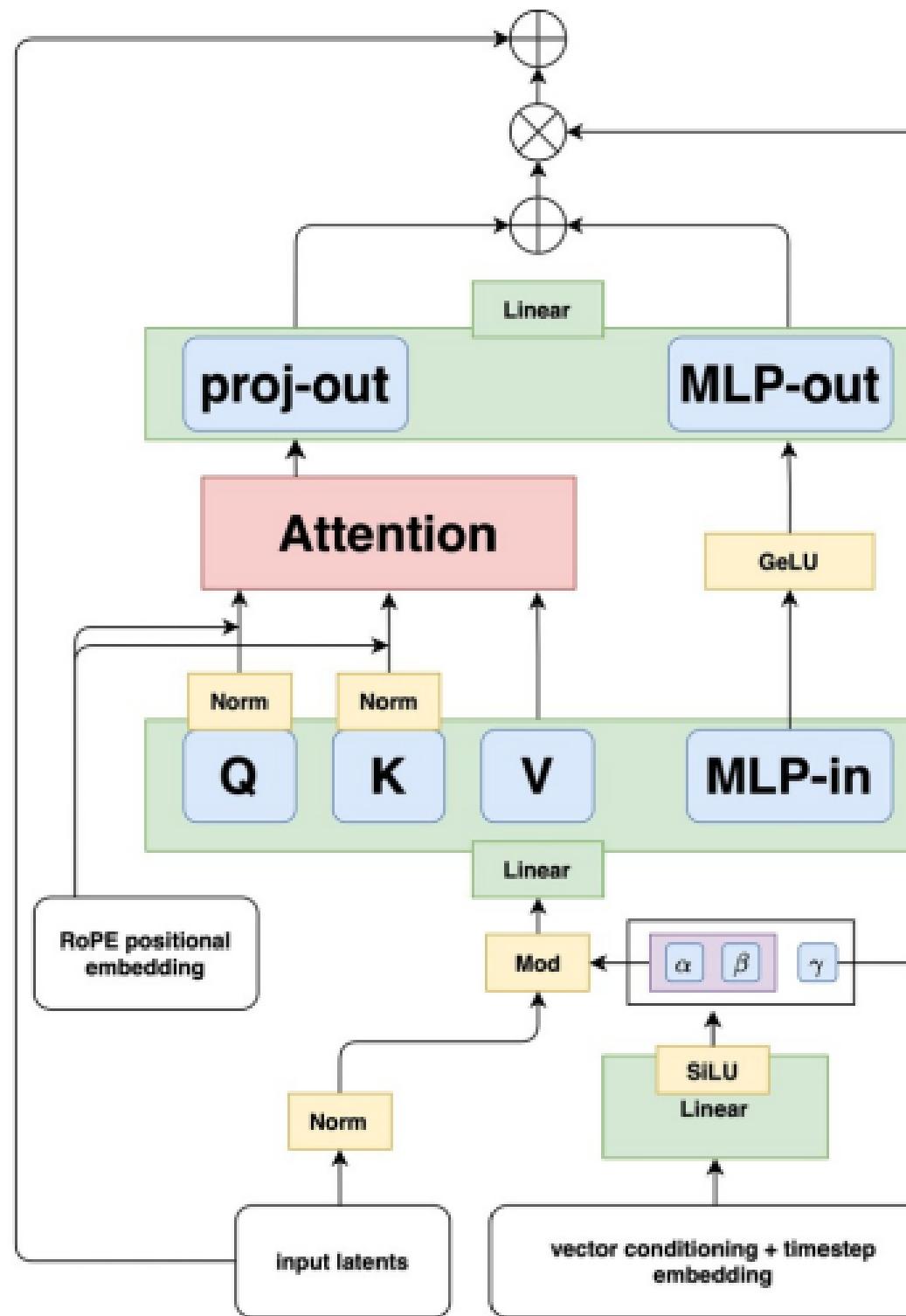
(c) “*Using this style, a grand piano made of shifting mirrors performs itself for an audience of empty velvet chairs in zero-gravity.*”



(d) “*Using this style, a spiral of vintage cameras captures its own collapse, each flash freeze-framing a different timeline.*”



Rectified Flow / Flow-Matching

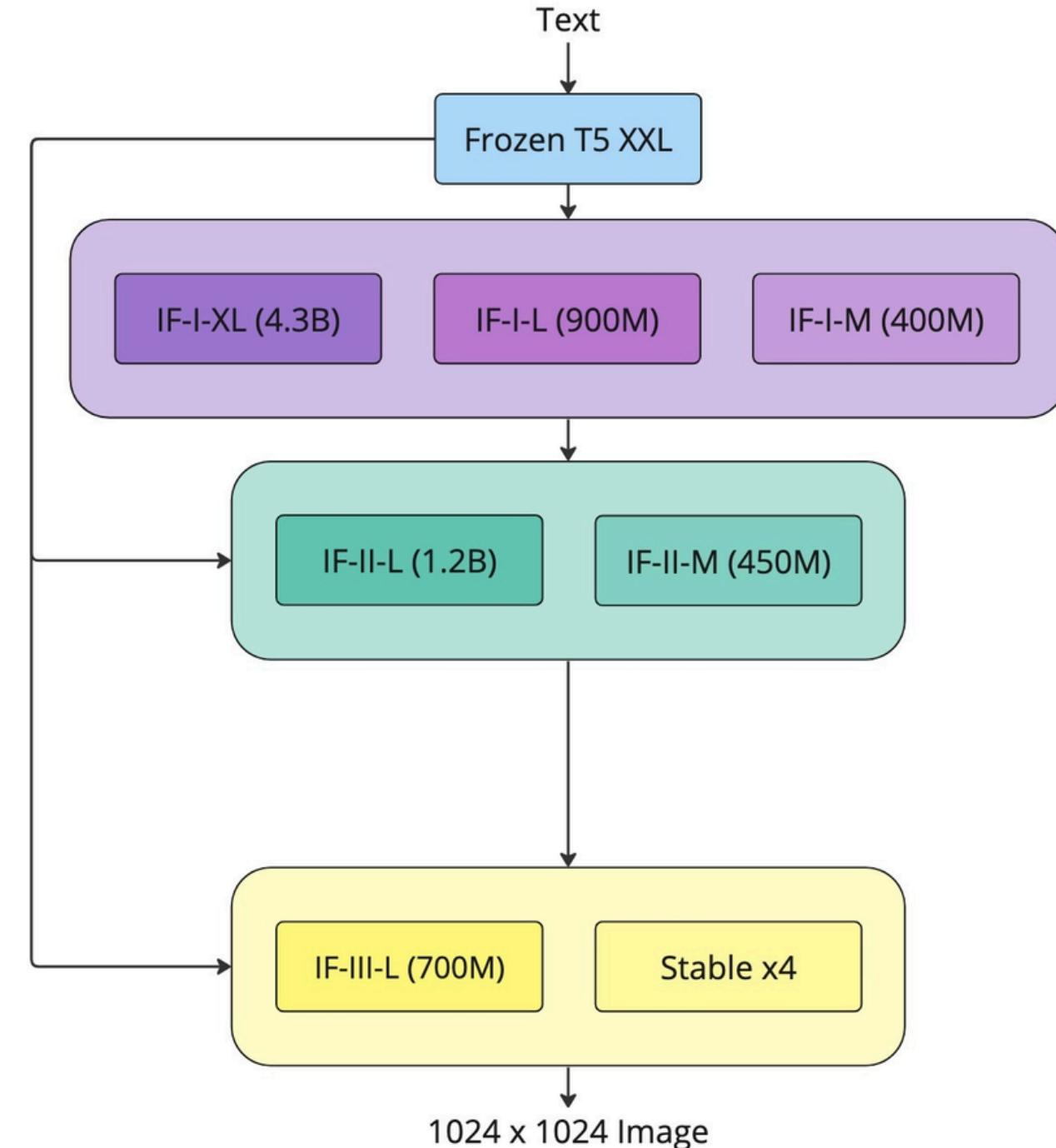


FLUX.1 Kontext: Flow Matching for In-Context Image Generation and Editing in Latent Space



DeepFloyd IF at StabilityAI

модульная каскадная pixel-diffusion

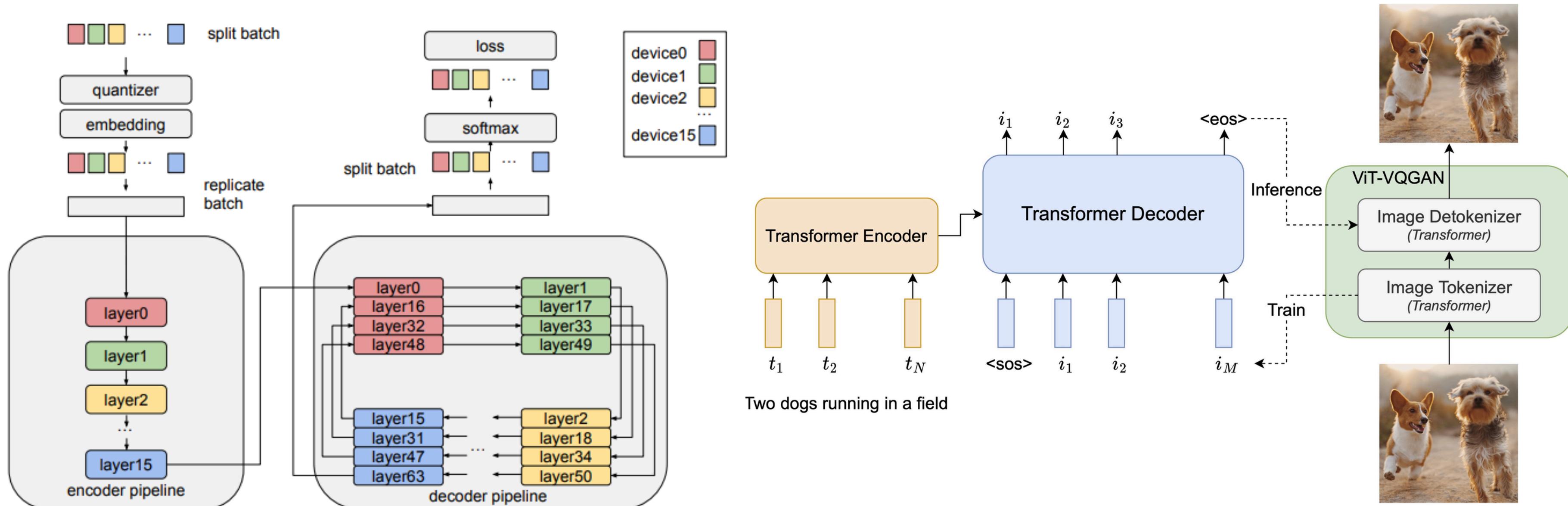


Inspired by Photorealistic Text-to-Image Diffusion Models
with Deep Language Understanding



Авторегрессионной T2I - Parti (github)

текст → последовательность *image*-токенов



Scaling Autoregressive Models for Content-Rich Text-to-Image Generation



Авторегрессионной T2I - Parti

Approach	Model Type	MS-COCO FID (↓)		LN-COCO FID (↓)	
		Zero-shot	Finetuned	Zero-shot	Finetuned
Random Train Images [10]	-		2.47		-
Retrieval Baseline	-	17.97	6.82	33.59	16.48
TReCS [46]	GAN	-	-	-	48.70
XMC-GAN [47]	GAN	-	9.33	-	14.12
DALL-E [2]	Autoregressive	~28	-	-	-
CogView [3]	Autoregressive	27.1	-	-	-
CogView2 [61]	Autoregressive	24.0	17.7	-	-
GLIDE [11]	Diffusion	12.24	-	-	-
Make-A-Scene [10]	Autoregressive	11.84	7.55	-	-
DALL-E 2 [12]	Diffusion	10.39	-	-	-
Imagen [13]	Diffusion	7.27	-	-	-
Parti	Autoregressive	7.23	3.22	15.97	8.39

Scaling Autoregressive Models for Content-Rich Text-to-Image Generation



Глобальные / наиболее заметные

Stable Diffusion XL → Latent Diffusion + U-Net, усиленный UNet и 2 text encoders

Stable Diffusion 3 / 3.5 → MMDiT (Multimodal Diffusion Transformer)

FLUX.1 → Rectified Flow Transformer.

Imagen 3 → latent diffusion

OpenAI 4o image generation → autoregressive image model (не diffusion)

DALL·E 3 → детали базовой архитектуры не раскрываются, но DALL·E 2 используют diffusion-модели для декодера



Российские флагманы

YandexART → cascaded diffusion

Kandinsky_3.0 → latent diffusion (U-Net-семейство)

Kandinsky_5.0 (image+video) → diffusion-transformer-
семейство / MM-DiT-направление (очень похоже на то)

ruDALL-E / Malevich → авторегрессионный трансформер
по последовательности токенов (текст + изображение)



Лидерборды/бенчмарки

Rapidata

LMArena Text-to-Image

Easier Painting Than Thinking

Artificial Analysis Image Arena Leaderboard

T2I-CompBench++



Text2video

- Усложнение по сравнению с T2I: нужно пространственное качество + временная согласованность.
- Типовые провалы: flicker, дрейф идентичности объекта, неустойчивые связи причинности/динамики.
- Что оцениваем? subject consistency, motion smoothness, temporal flickering, spatial relationship и др.
- T2V - это генерация распределения над траекториями пикселей, а не просто картинка + немного шума.



Text-to-Video generation: "a horse galloping on a street"



Text-to-Video generation: "a panda is playing guitar on times square"

Text2Video-Zero



Карта местности

1. Video Diffusion (end-to-end)

- Моделируем $p(x_{\{1:T\}}|text)$ через диффузию по видео-тензору/латентам.
- Пример: [Imagen Video](#) - каскад video diffusion + spatial/temporal super-resolution. (arXiv)

2. Поднимаем T2I до T2V

- Мини-адаптации/модификации inference, чтобы T2I стал T2V.
- Пример: [Make-A-Video](#), [Text2Video-Zero](#).

3. Дискретные видео-токены + Transformer

- Сначала токенизуем видео, затем генерим токены (часто masked transformer).
- Пример: [Phenaki](#).



Каскадные диффузии для видео

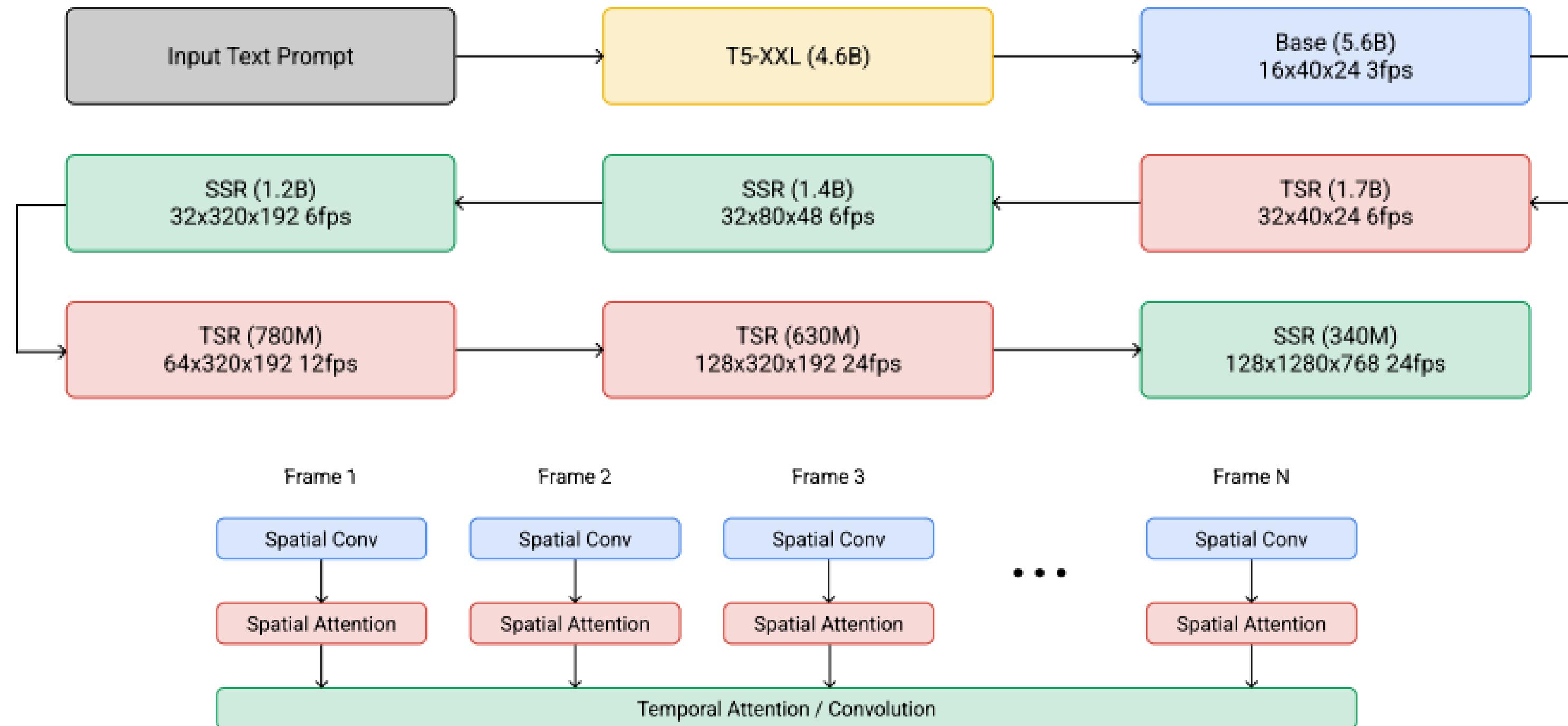


Imagen Video: High Definition Video Generation with Diffusion Models



Latent Video Diffusion: Stable Video Diffusion (SVD)



"An exploding cheese house"



"A fat rabbit wearing a purple robe walking through a fantasy landscape"

SVD: ключ - добавить temporal layers и хорошо выстроить этапы обучения
(T2I pretrain → video pretrain → HQ finetune)

Качество сильно зависит от курации видеоданных (captioning/filtering) и стадийности обучения.
Просто дообучить SD на видео почти всегда дает flicker/развал.

Imagen Video: High Definition Video Generation with Diffusion Models



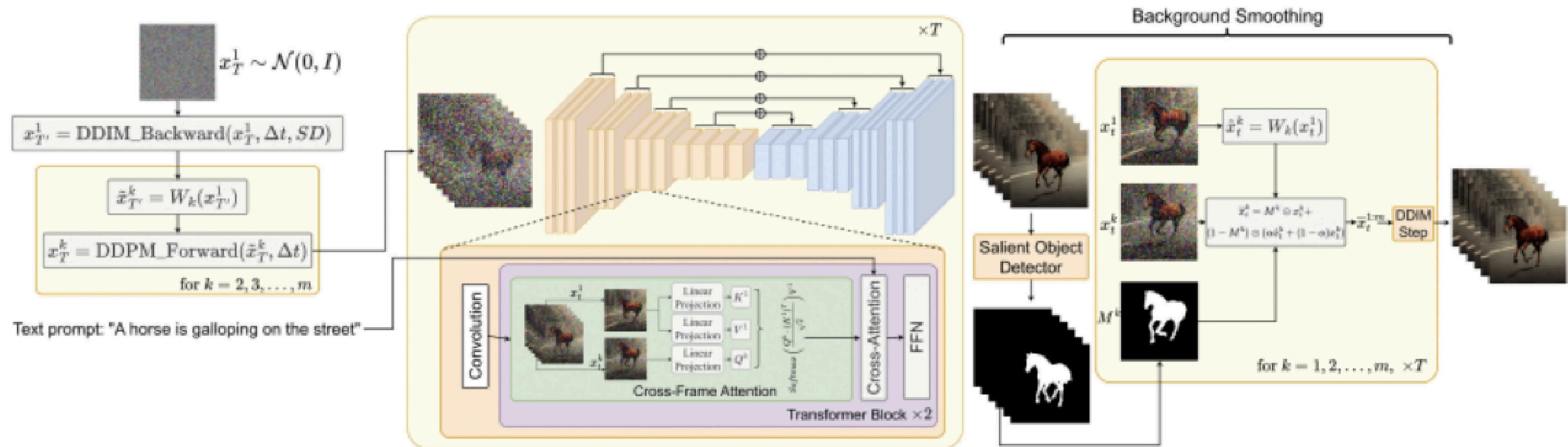
```
1 import torch
2 from einops import rearrange, repeat
3
4
5 def append_dims(x: torch.Tensor, target_dims: int) -> torch.Tensor:
6     """Appends dimensions to the end of a tensor until it has target_dims dimensions."""
7     dims_to_append = target_dims - x.ndim
8     if dims_to_append < 0:
9         raise ValueError(
10             f"input has {x.ndim} dims but target_dims is {target_dims}, which is less"
11         )
12     return x[(..., ) + (None, ) * dims_to_append]
13
14
15 class LinearPredictionGuider:
16     def __init__(
17         self,
18         max_scale: float,
19         num_frames: int,
20         min_scale: float = 1.0,
21     ):
22         self.min_scale = min_scale
23         self.max_scale = max_scale
24         self.num_frames = num_frames
25         self.scale = torch.linspace(min_scale, max_scale, num_frames).unsqueeze(0)
26
27     def __call__(self, x: torch.Tensor, sigma: float) -> torch.Tensor:
28         x_u, x_c = x.chunk(2)
29
30         x_u = rearrange(x_u, "(b t) ... -> b t ...", t=self.num_frames)
31         x_c = rearrange(x_c, "(b t) ... -> b t ...", t=self.num_frames)
32         scale = repeat(self.scale, "1 t -> b t", b=x_u.shape[0])
33         scale = append_dims(scale, x_u.ndim).to(x_u.device)
34
35         return rearrange(x_u + scale * (x_c - x_u), "b t ... -> (b t) ...")
```

Figure 15. PyTorch code for our novel *linearly increasing guidance* technique.



Training-free: превращаем T2I в T2V без дообучения

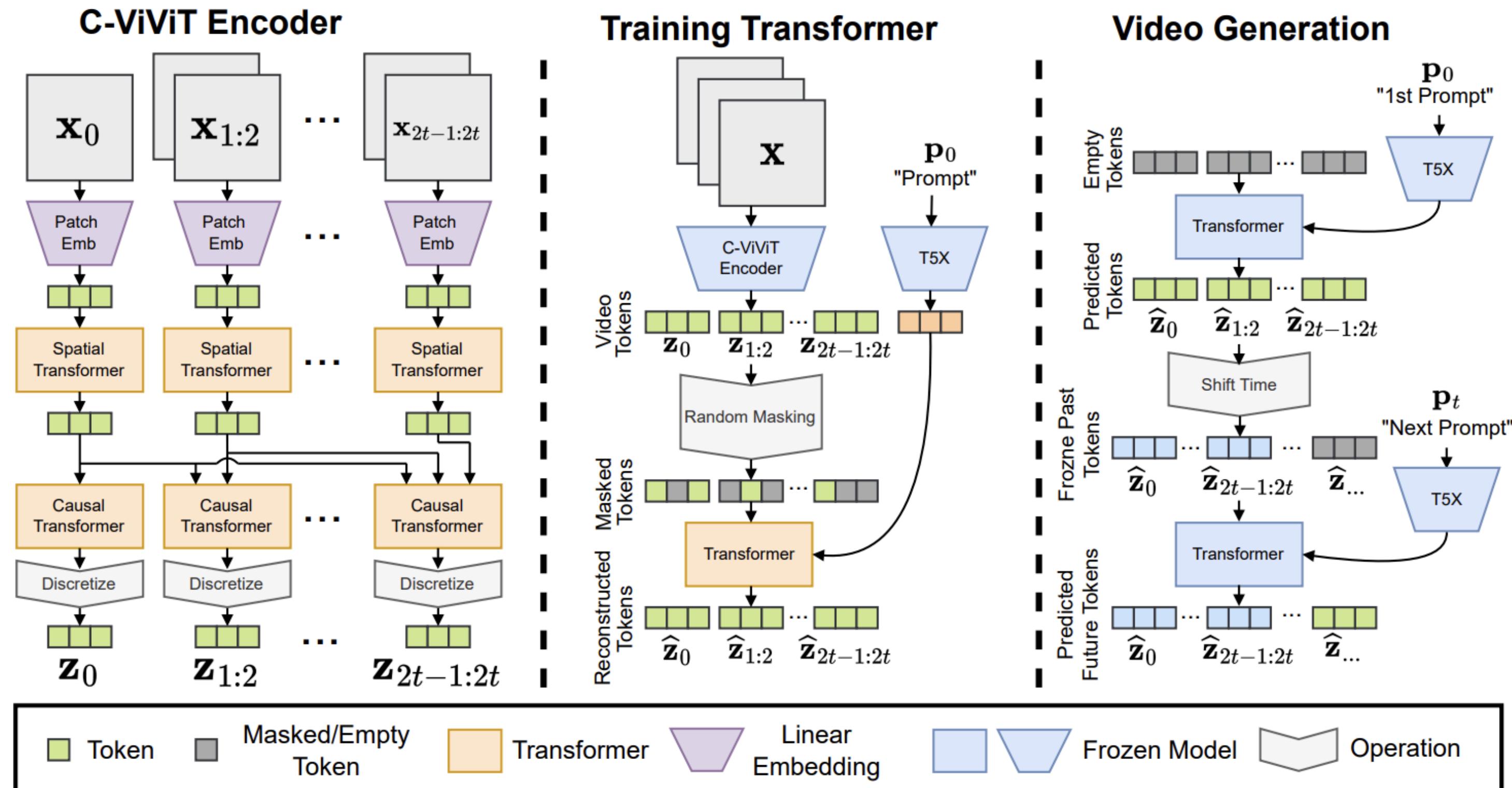
- **Make-A-Video**: учим как выглядит мир из text-image, как двигается из unlabeled video; переносим прогресс T2I в T2V.
- **Text2Video-Zero**: формулирует zero-shot / training-free T2V, используя уже обученные T2I diffusion (например, Stable Diffusion) и модификации для временной согласованности.



Text2Video-Zero: Text-to-Image Diffusion Models are Zero-Shot Video Generators



Discrete token video models



Phenaki: Variable length video generation from open domain textual descriptions



Зоопарк text-to-video

Diffusion на видео (U-Net / Latent Diffusion + временные блоки)

Video Diffusion Models (VDM, Ho et al., 2022) - расширяем image diffusion на видео

Imagen Video (Google, 2022) - каскад video diffusion моделей

Make-A-Video (Meta, 2022) - видеодиффузия

Stable Video Diffusion (Stability AI, 2023) - latent video diffusion

ModelScopeT2V (Alibaba DAMO, 2023) - эволюция от Stable Diffusion: VQGAN + text encoder + denoising UNet и добавлены spatio-temporal blocks для согласованности кадров

VideoCrafter2 (CVPR'24) - анализ связки spatial/temporal модулей и схема обучения, чтобы из низкокачественное видео вытянуть motion, а качество - из изображений

Open Diffusion Models for High-Quality Video Generation (2023) - строят T2V на базе SD 2.1, добавляя temporal attention layers в UNet + joint image/video training

Text2Video-Zero (ICCV'23) - training-free T2V

AnimateDiff (ICLR'24) - вставляют motion module в замороженный T2I diffusion и учат этот модуль на видео, потом он plug-and-play для многих SD-совместимых моделей

Lumiere (Google, 2024) - Space-Time U-Net: генерирует весь ролик целиком за один проход, а не keyframes в temporal SR



Зоопарк text-to-video

Diffusion Transformer (DiT-подобные): токены - spacetime-патчи латентов

Sora (OpenAI, 2024) - text-conditional diffusion, joint training на image+video, денойзер - это transformer по spacetime patches видео/изображений в латентном коде

CogVideoX (Zhipu AI, 2024) - diffusion transformer, используется 3D VAE для сжатия видео по пространству и времени, есть expert transformer для лучшего text-video alignment

Open-Sora (2024/2025) - Spatial-Temporal Diffusion Transformer (STDiT): декуплируют spatial и temporal attention + сильно сжимающий 3D autoencoder, open-source

Veo 3 (Google/DeepMind, tech report) - latent diffusion по video+audio латентам + transformer-based denoising



Зоопарк text-to-video

Дискретные видео-токены + Transformer (не diffusion)

Phenaki (2022-23) - видео в токены (C-ViViT tokenizer с causal attention по времени), текст в токены, генерация токенов видео через bidirectional masked transformer, потом уже детокенизация в видео.

VideoPoet (2023-24) - decoder-only transformer в стиле LLM, который умеет работать с мультимодальными входами (текст/видео/аудио/изображения) и генерировать видео.

CogVideo (ICLR, 2022) - большой Transformer для T2V, наследует pretrained T2I (CogView2) и использует multi-frame-rate training для лучшего соответствия тексту.

