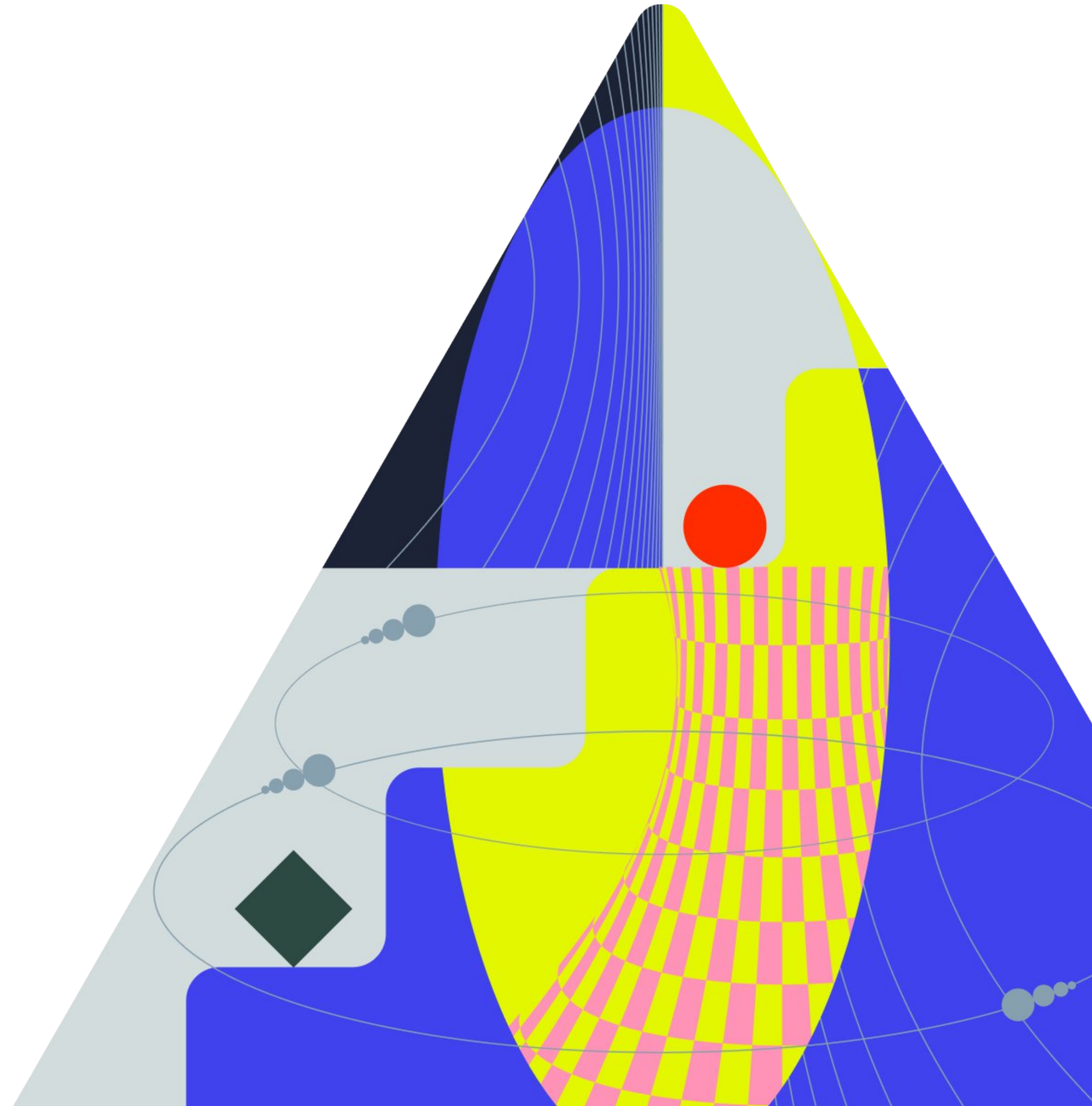
Школа анализа данных

Diffusion Model Architectures Text-to-Image Generation



Denis Kuznedelev

Yandex Research

Lecture 3 | Diffusion model architectures

01 Image Generation Metrics

02 Diffusion Model Architectures

03 Training and Inference Techniques

04 Text-to-Image Generation

Image generation metrics

- Image quality assessment is a complex and nuanced topic with many open questions and research problems
- Research papers present various image generation metrics for evaluating model performance

Good or bad?

Image generation metrics

FID
(Fréchet Inception Distance)

- Estimates how “close” generated samples are to the target data distribution
- Based on features extracted from a pre-trained Inception v3 model (Imagenet-1k)
- Assumes that Inception embeddings are multivariate normal variables
- Requires a large number of samples (10k or more) for accurate estimation of covariance

$$\mathrm{FID}(\mathcal{N}(\mu,\Sigma),\mathcal{N}(\mu',\Sigma')) = \|\mu'-\mu'\|_2^2 + \mathrm{tr}\left(\Sigma+\Sigma'-2(\Sigma\Sigma')^{1/2}\right)$$

μ, Σ ⟶ Mean and covariance (real)

μ', Σ' ⟶ Mean and covariance (generated)

Heusel M. et al., Gans trained by a two time-scale update rule converge to a local nash equilibrium, NeurIPS 2017

Image generation metrics

CLIP Score

- Estimates “prompt alignment”
- Measures average cosine similarity between the text and image embeddings

$$\text{CLIP}_{\text{score}} = \frac{(f_{\text{text}}, f_{\text{image}})}{\|f_{\text{text}}\| \, \|f_{\text{image}}\|}$$

- Depends on the specific CLIP text and vision models used

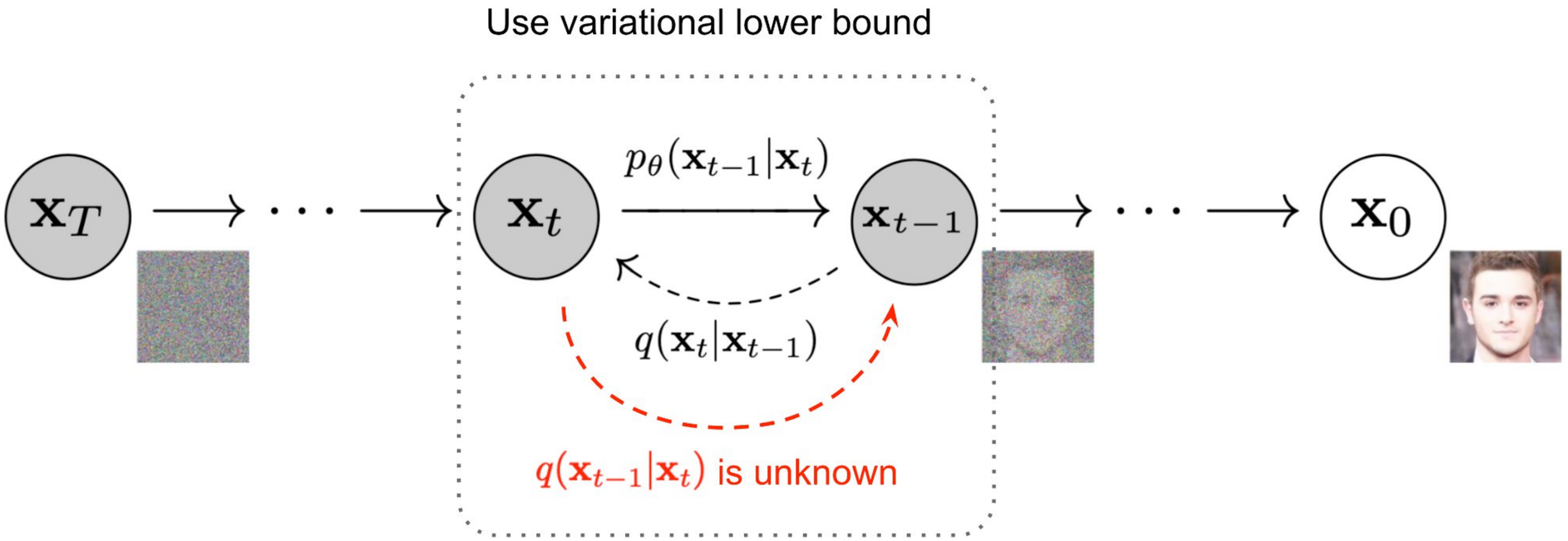
$$\text{FID}(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\mu', \Sigma')) = \|\mu' - \mu'\|_2^2 + \text{tr}\left(\Sigma + \Sigma' - 2(\Sigma\Sigma')^{1/2}\right)$$

μ, Σ ⟶ Mean and covariance (real)

μ', Σ' ⟶ Mean and covariance (generated)

Hessel J. et al. , Clipscore: A reference-free evaluation metric for image captioning, arXiv:2104.08718

Diffusion model architectures



- Diffusion models transform noise into samples from a target data distribution
- Model outputs are of the same shape (B, H, W, C) as inputs

Which NN architectures can produce outputs of the same shape as their inputs?

J. Ho et al., Denoising Diffusion Probabilistic Models, NeurIPS2020

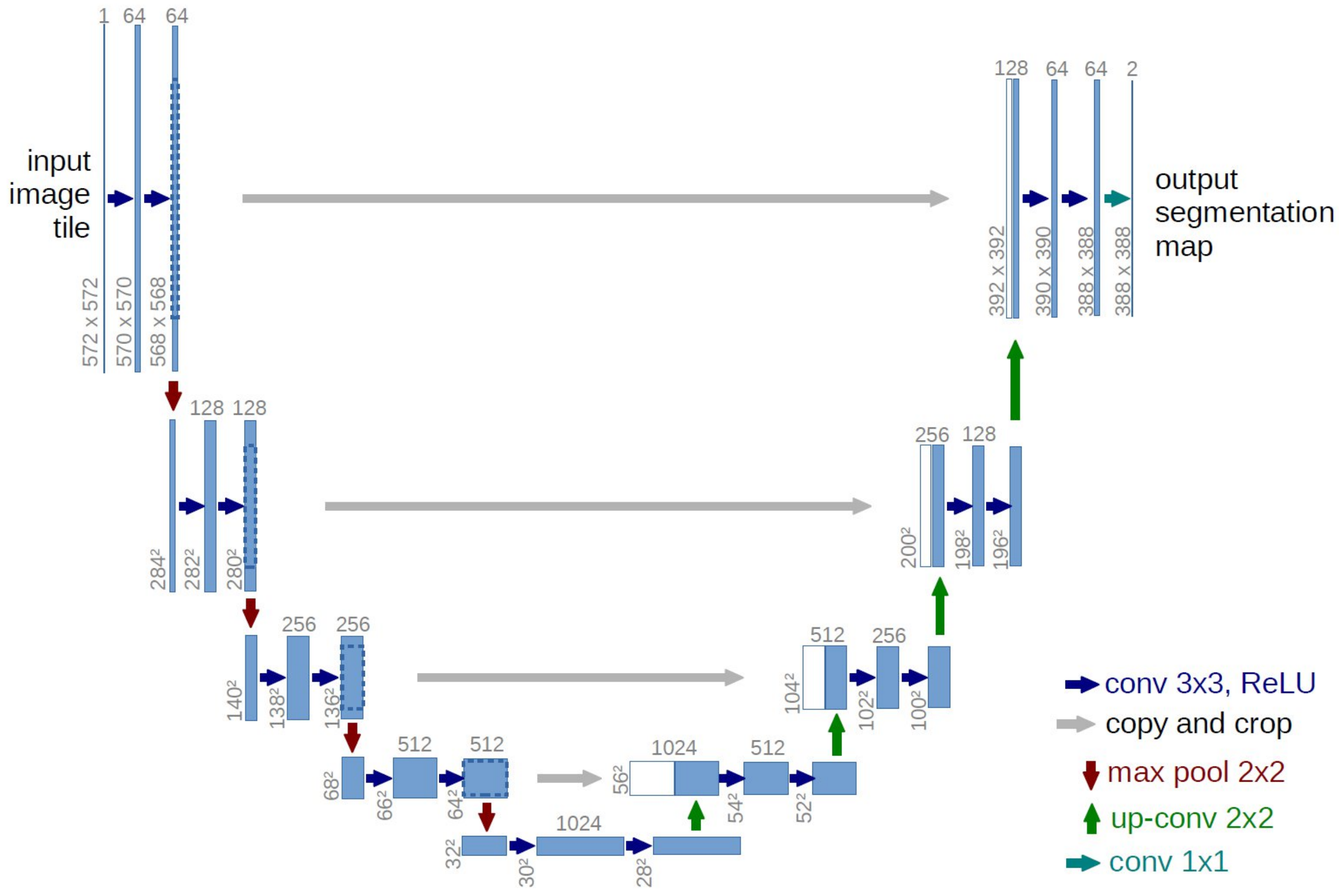
Diffusion model architectures

UNet?	Transformer?	SSM?
		

Diffusion model architectures

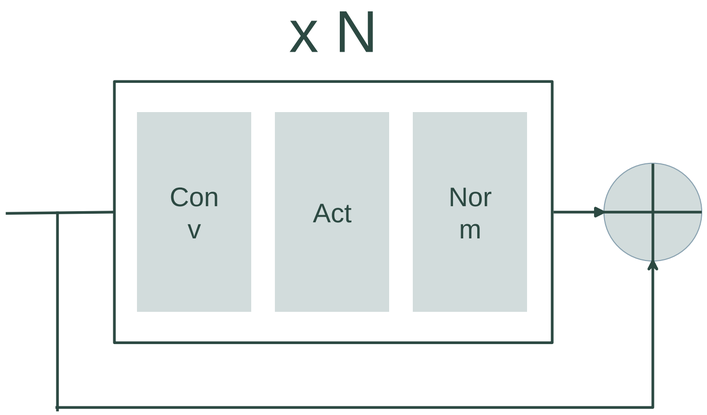
UNet

- The most popular architecture is old buddy UNet, widely used for segmentation tasks
- UNet is a stack of Residual Convolutional Blocks operating on different resolutions
- Many diffusion models also include Attention blocks
- Deeper layers have more channels and process smaller feature maps



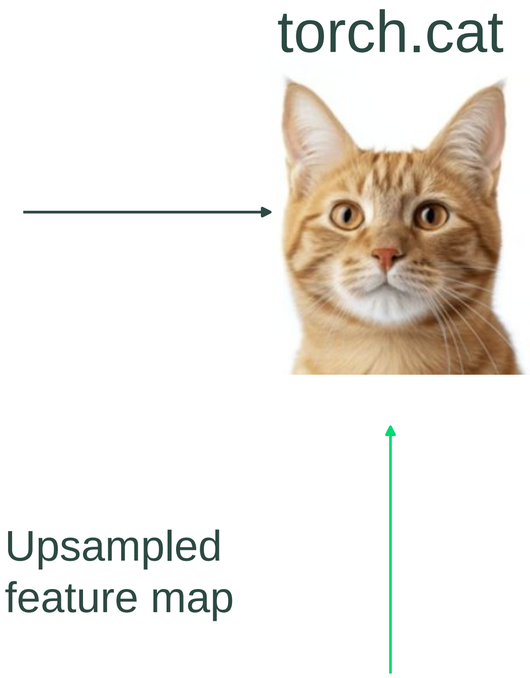
Diffusion model architectures

ResBlock



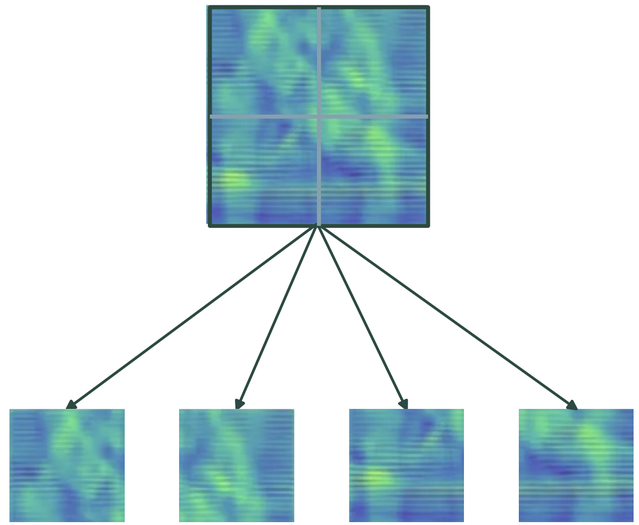
Additive Skip Connection

Concatenation in Decoder



Upsampled
feature map

Attention



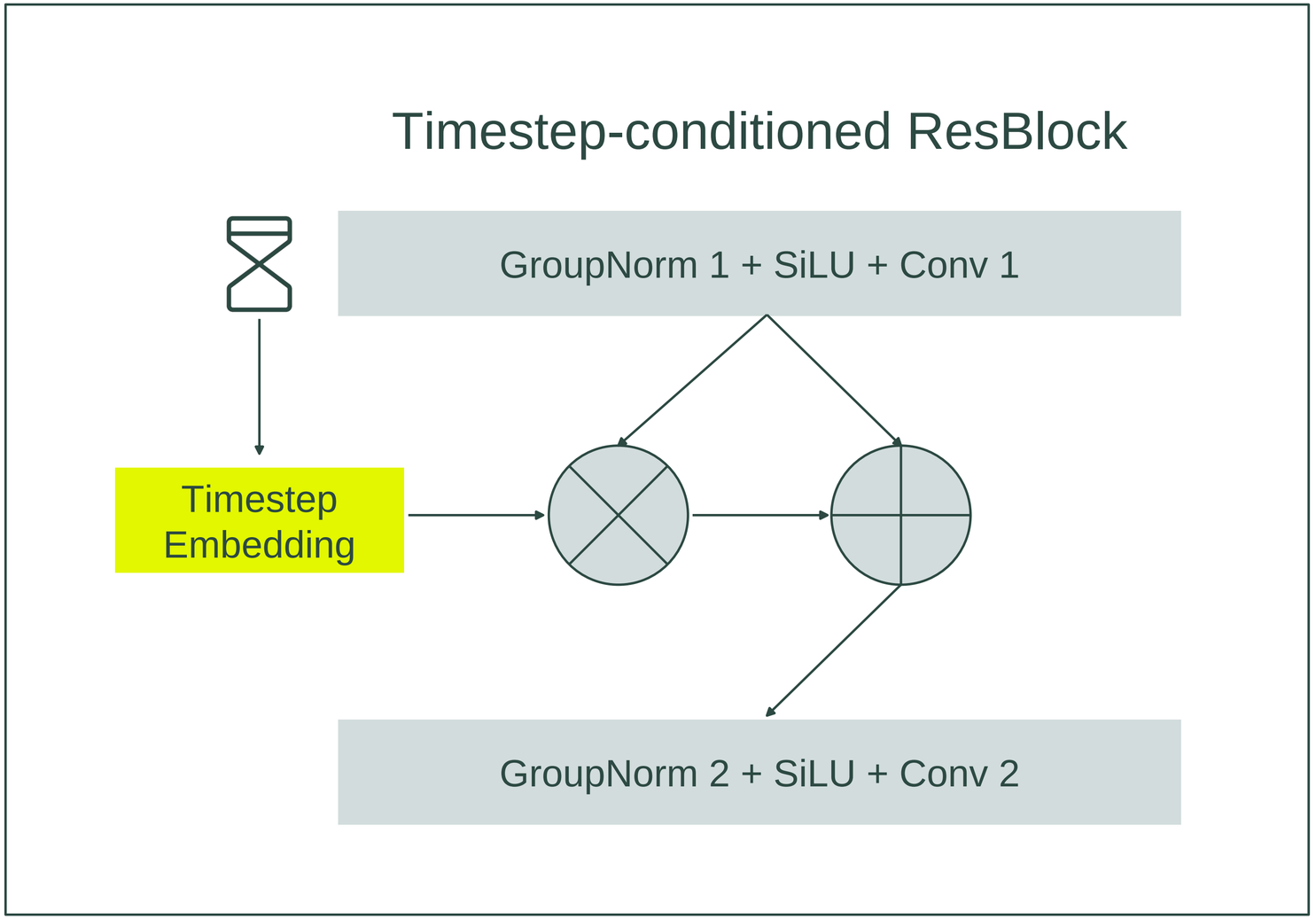
Treats pixels of a feature map as
a sequence of tokens

Diffusion model architectures

UNet

- We have to account for the diffusion timestep conditioning
- Standard option is activation modulation in residual blocks

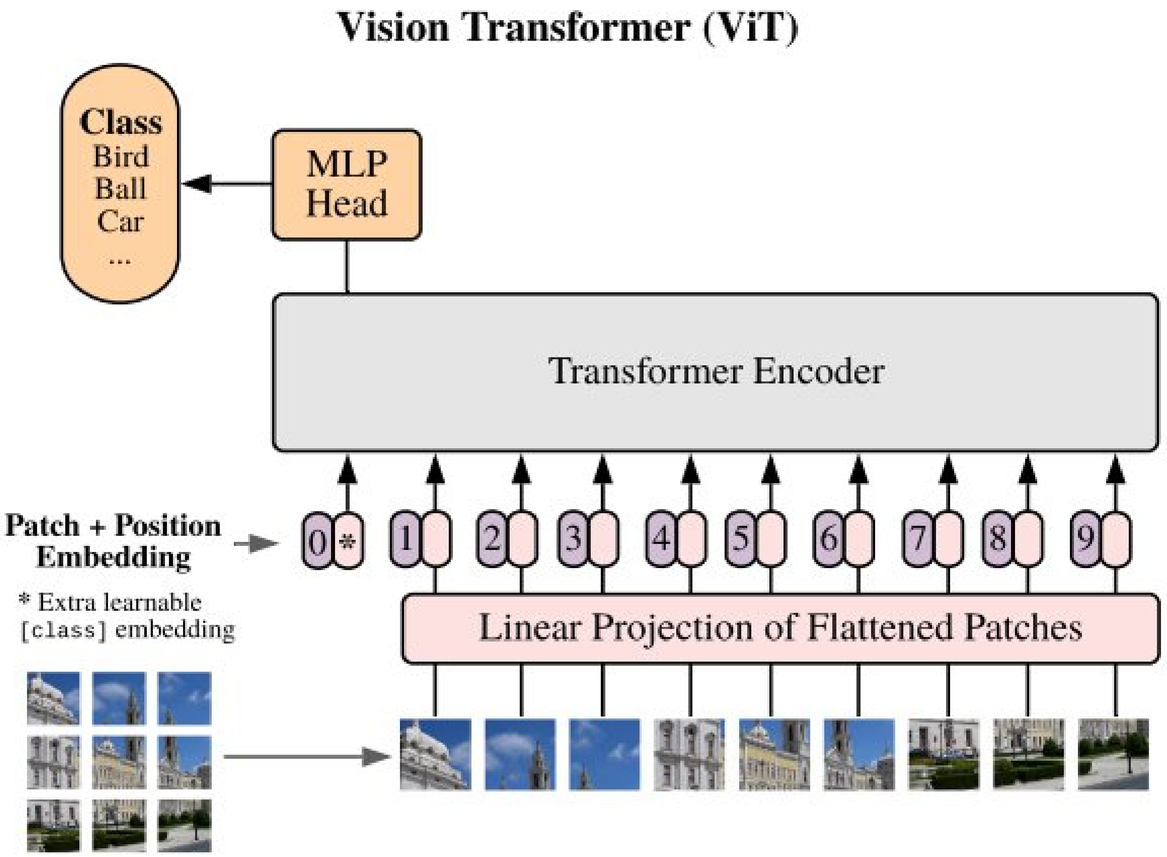
$$y = \text{scale}(t)x + \text{shift}(t)$$



Diffusion model architectures

What about transformer?

- A diffusion model architecture can be built entirely from Transformer blocks
- Vision Transformers (ViTs) divide input images into small patches, apply convolution, and process the resulting sequence of patches like language models
- Token positions are encoded using 2D positional embeddings



Dosovitskiy A., An image is worth 16x16 words: Transformers for image recognition at scale, arXiv:2010.11929

Diffusion model architectures

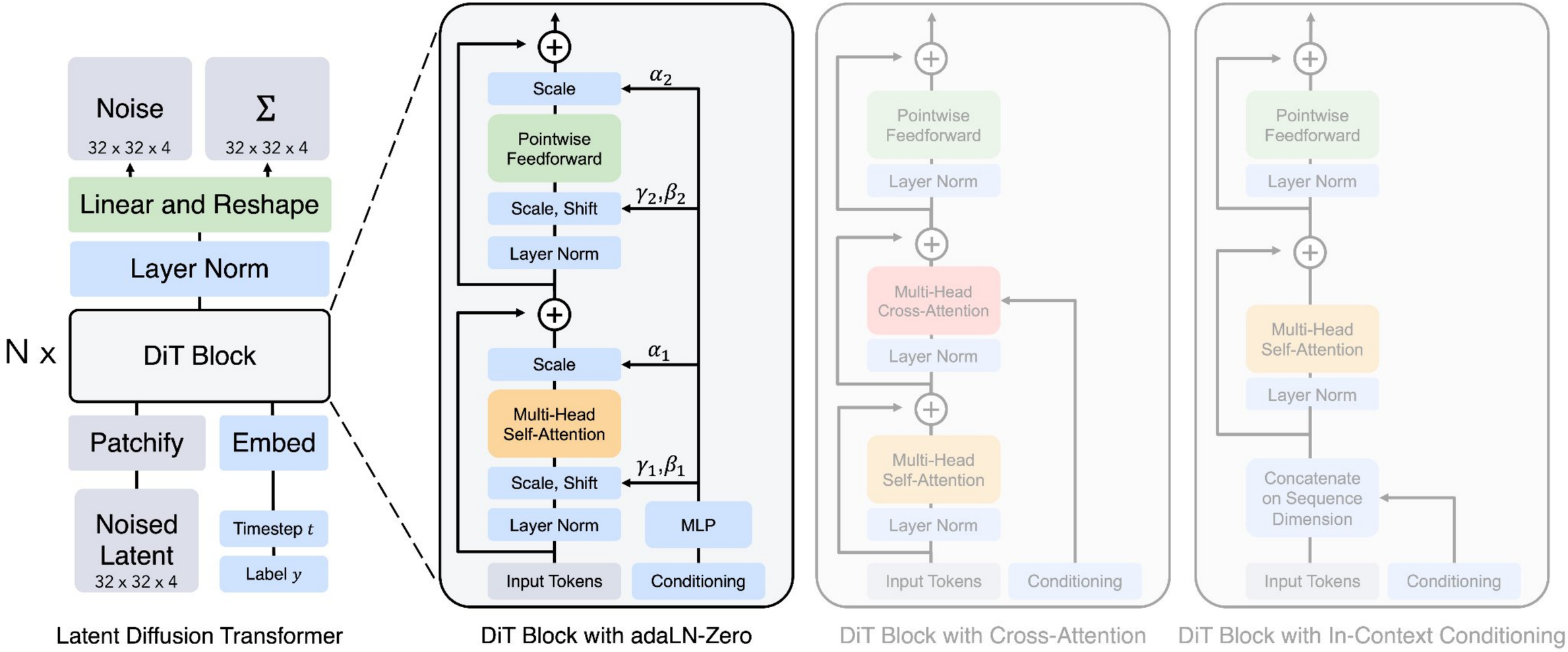
What about transformer?

- Similarly to ViT one splits image into small patches
- Timestep (and class) conditioning is performed via AdaLN
- Degrees of freedom are the model size and patch size

$$\mathrm{AdaLN}(\mathbf{x}, \mathbf{c}, t) = \mathrm{LN}(\mathrm{scale}(\mathbf{c}, t)\mathbf{x} + \mathrm{shift}(\mathbf{c}, t))$$

Peebles W., Xie S., Scalable diffusion models with transformers, ICCV 2023

Diffusion model architectures



Latent Diffusion Transformer | DiT Block with adaLN-Zero | DiT Block with Cross-Attention | DiT Block with In-Context Conditioning

Diffusion model architectures

- Computational complexity scales linearly with the model size
- The number of patches scales as follows:

$$\mathcal{O}\left(\frac{1}{p^2}\right)$$

for patch size p

- Smaller patches require more computational resource

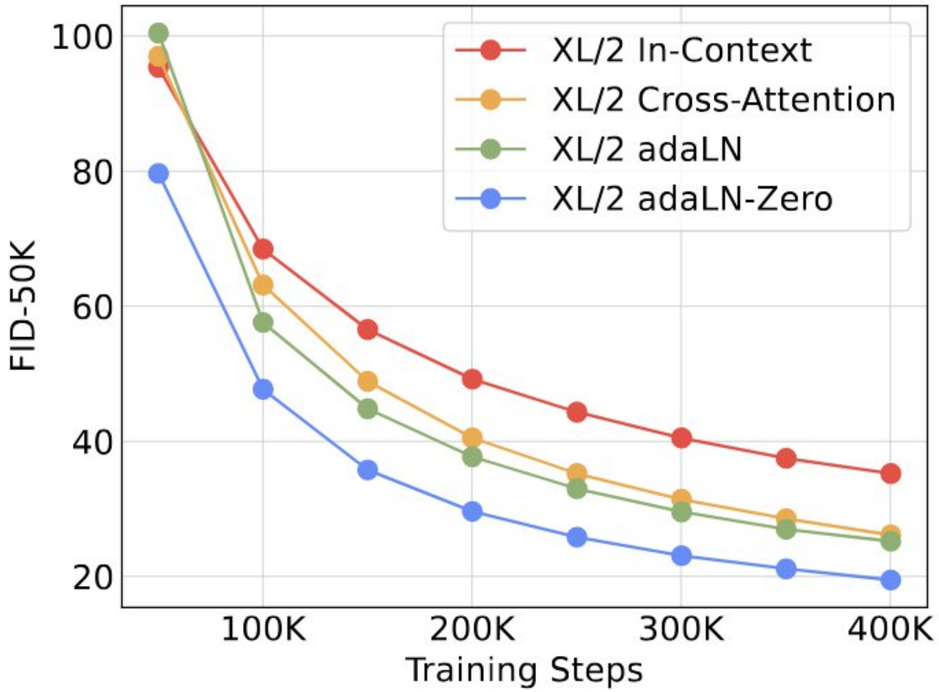
Model	Layers N	Hidden size d	Heads	Gflops (I=32, p=4)
DiT-S	12	384	6	1.4
DiT-B	12	768	12	5.6
DiT-L	24	1024	16	19.7
DiT-XL	28	1152	16	29.1

Model configurations tested in our study

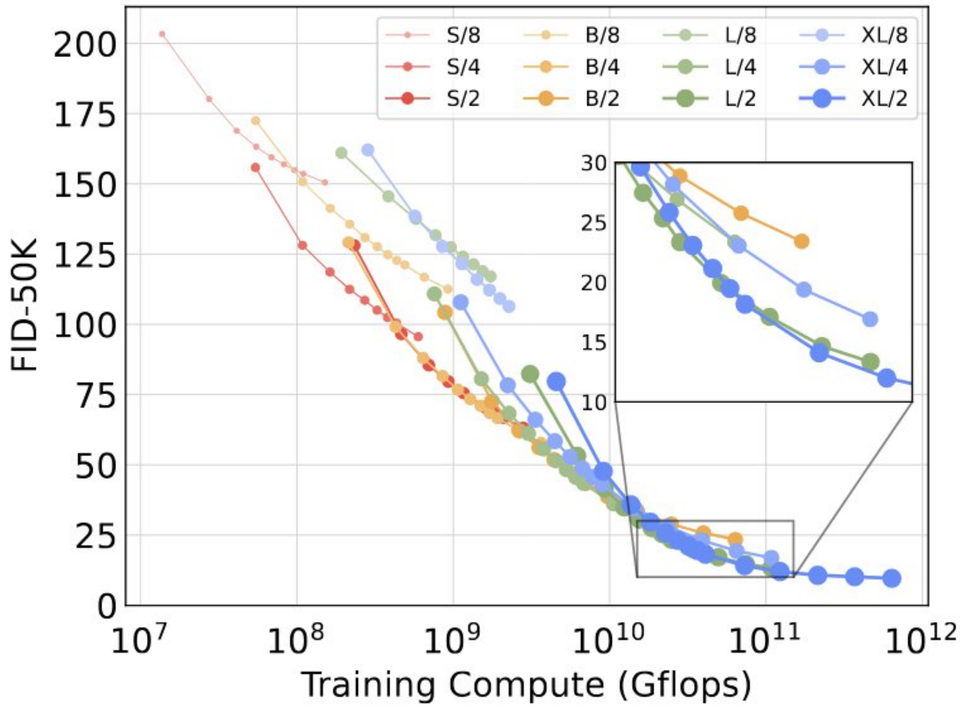
Peebles W., Xie S., Scalable diffusion models with transformers, ICCV 2023

Diffusion model architectures

AdaLN is great

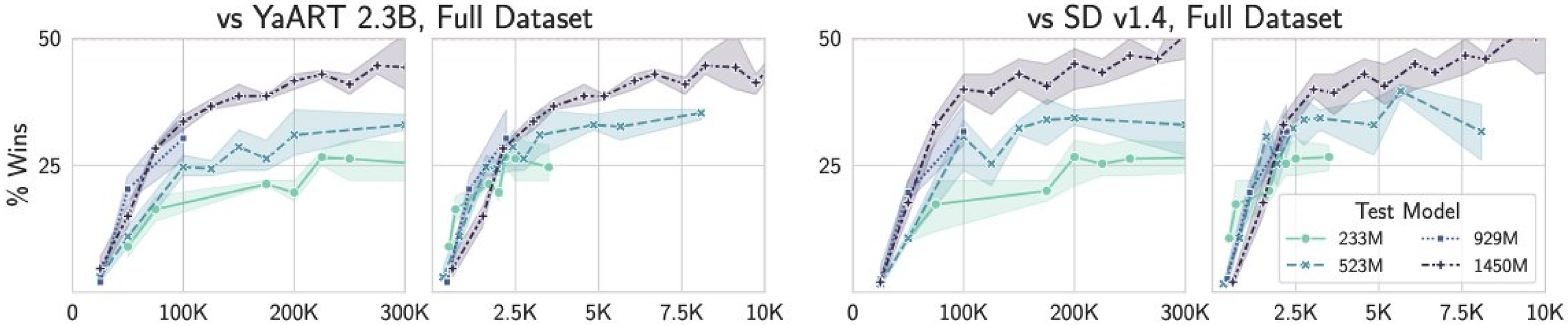


More compute — better output



Diffusion model architectures

Scaling helps UNet, too



Diffusion model architectures

- The goal is to generate high-resolution images (512+ pixels)
- Inference cost scales with both model size and image resolution
- Computational complexity s — (image size):
  — Convolutions — $\mathcal{O}\left(s^2\right)$
  — Self-Attention — $\mathcal{O}\left(s^4\right)$

$$\#\text{pixels} = \mathcal{O}\left(s^2\right)$$

- Processing large feature maps with complex models is computationally demanding:

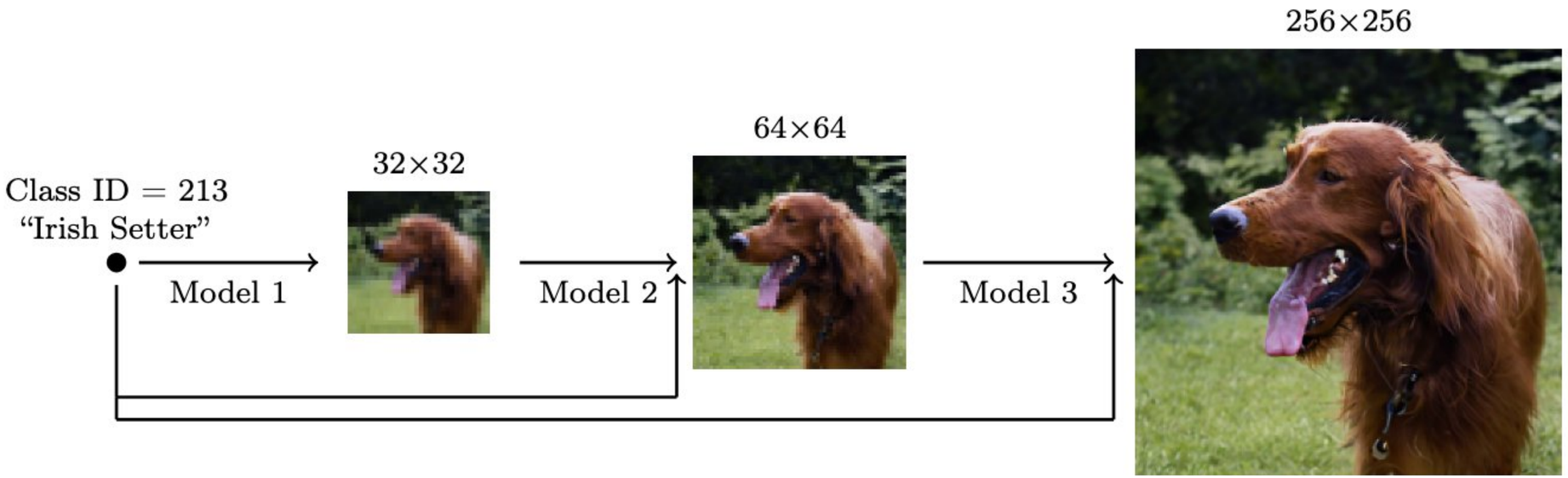
What can we do?

Progressive generation,
from small scale to large scale

Map the problem to some other space,
where the generation is cheaper

Diffusion model architectures

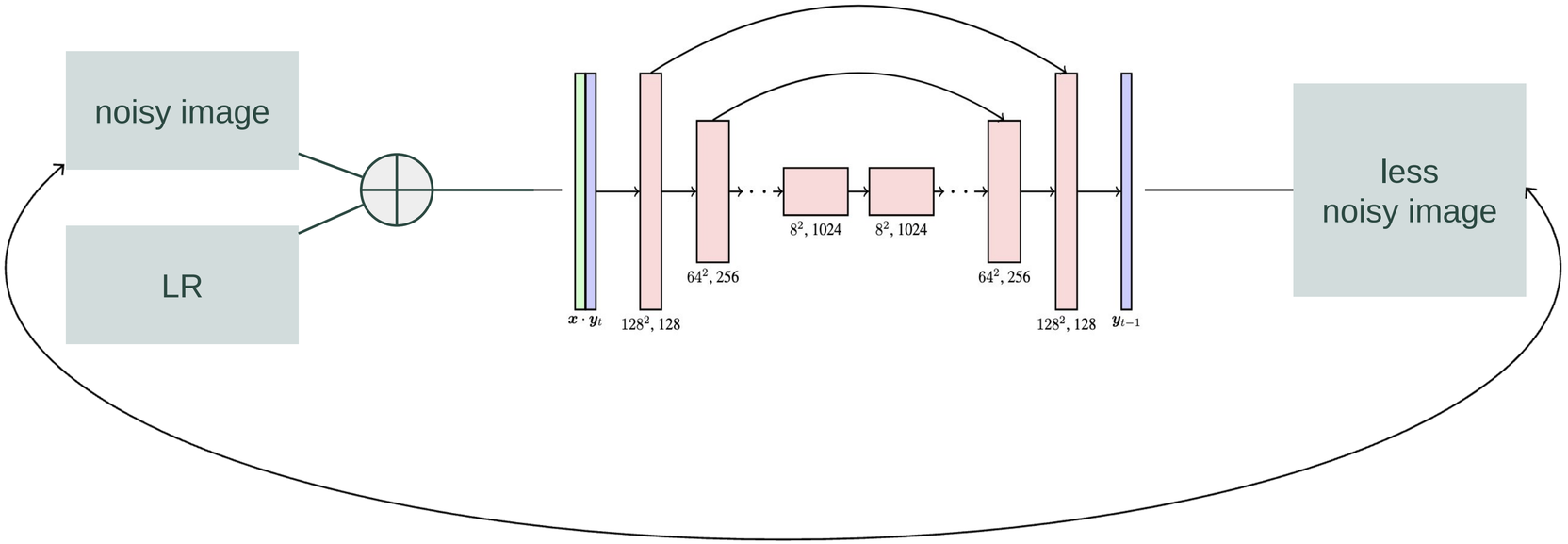
Cascaded Diffusion



- Generates a low-resolution image and then upscales it using one or more super-resolution models
- Each super-resolution model is typically a diffusion model

Ho J. et al. , Cascaded diffusion models for high fidelity image generation, JMLR 2022

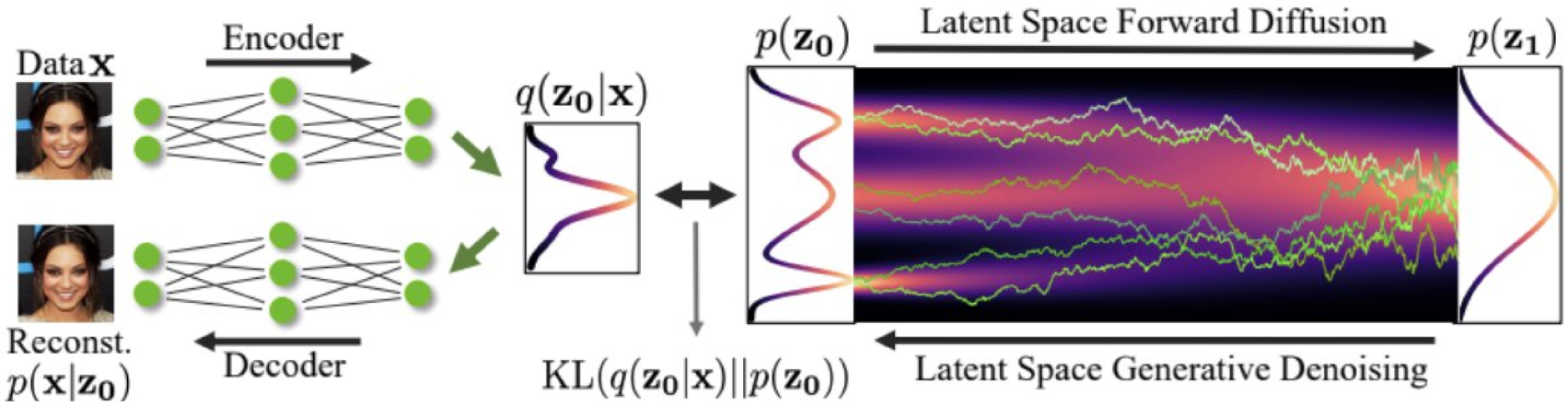
Diffusion model architectures



- Diffusion SR model: a standard UNet architecture adapted for pixel-space diffusion
- Key difference: the model incorporates low-resolution image conditioning by concatenating it with the input noise
- SR models usually have fewer parameters and work well with less sampling steps:
  — Base model ~1B parameters, 20–50 sampling steps
  — SR model ~300-500M parameters, 10–20 sampling steps

Diffusion model architectures

Latent Diffusion

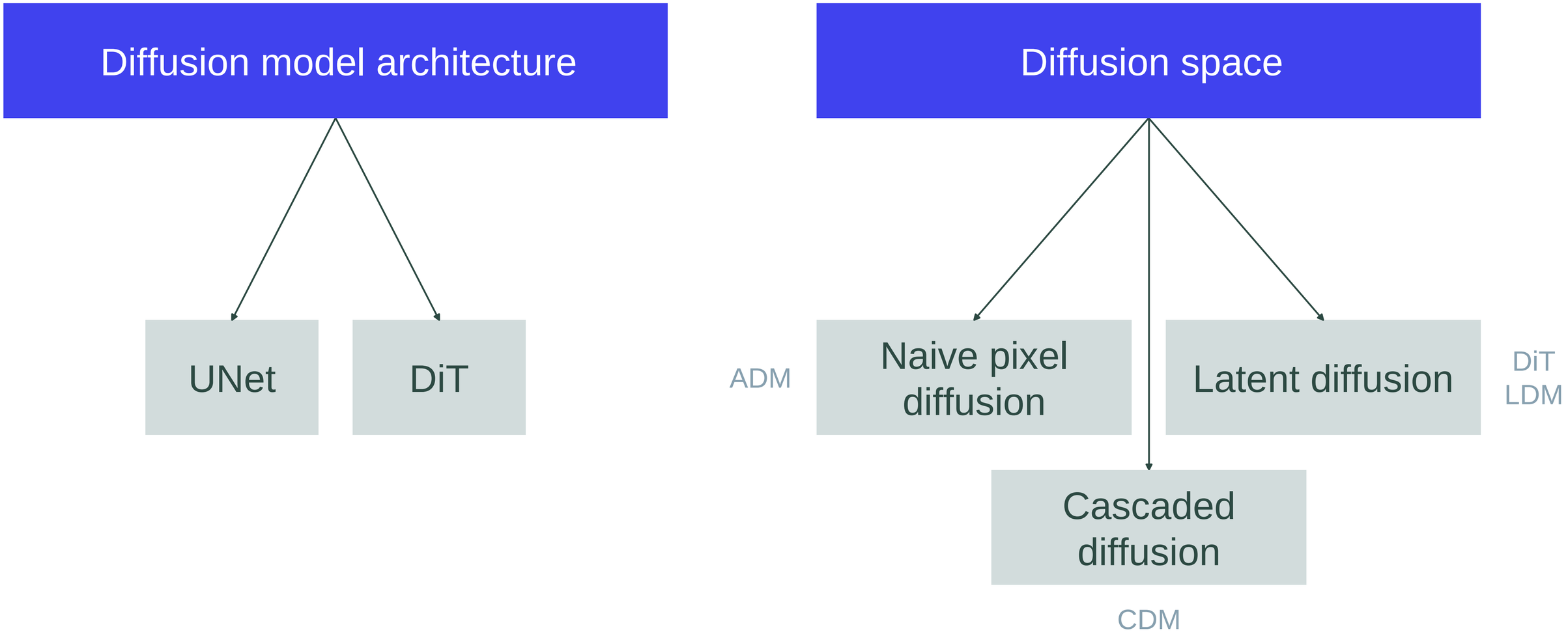


- Encodes original data into a space with smaller spatial dimensions using AE, VAE, VQ-VAE
- Performs the diffusion process in the latent space
- Decodes the final sample back from that space

Typical downsampling factor: 8x

Ho J. et al. , Cascaded diffusion models for high fidelity image generation, JMLR 2022

Diffusion model architectures



Inference techniques

Classifier guidance

Consider a case of conditional diffusion:

$$p(x \mid y)$$

y — is some input (class, text, whatever…)

To amplify the condition's influence, consider using a “sharpened” condition:

$$p_\gamma(x \mid y) \propto p(x) \cdot p(y \mid x)^\gamma. \quad \gamma \text{ guidance scale}$$

Train a separate classifier model in conjunction with the diffusion model
Using the classifier's gradients to guide the generation process

$$\nabla_x \log p_\gamma(x \mid y) = \nabla_x \log p(x) + \gamma \nabla_x \log p(y \mid x).$$

Inference techniques

Classifier-free guidance

However, there is a neat trick that allows us to use a single model: classifier-free guidance

$$p(y \mid x) = \frac{p(x \mid y) \cdot p(y)}{p(x)}$$

$$\implies \log p(y \mid x) = \log p(x \mid y) + \log p(y) - \log p(x)$$

Bayes' rule

$$\implies \nabla_x \log p(y \mid x) = \nabla_x \log p(x \mid y) - \nabla_x \log p(x).$$

Let's consider a "sharpened distribution":

$$\nabla_x \log p_\gamma(x \mid y) = \nabla_x \log p(x) + \gamma \left(\nabla_x \log p(x \mid y) - \nabla_x \log p(x) \right),$$

$$\nabla_x \log p_\gamma(x \mid y) = (1 - \gamma) \nabla_x \log p(x) + \gamma \nabla_x \log p(x \mid y).$$

We only need to perform a conditional and unconditional forward pass using the original model!

Inference techniques

Classifier-free guidance

Algorithm 2 Conditional sampling with classifier-free guidance

Require: w: guidance strength
Require: $\mathbf{c}$: conditioning information for conditional sampling
Require: $\lambda_1, \ldots, \lambda_T$: increasing log SNR sequence with $\lambda_1 = \lambda_{\min}, \lambda_T = \lambda_{\max}$

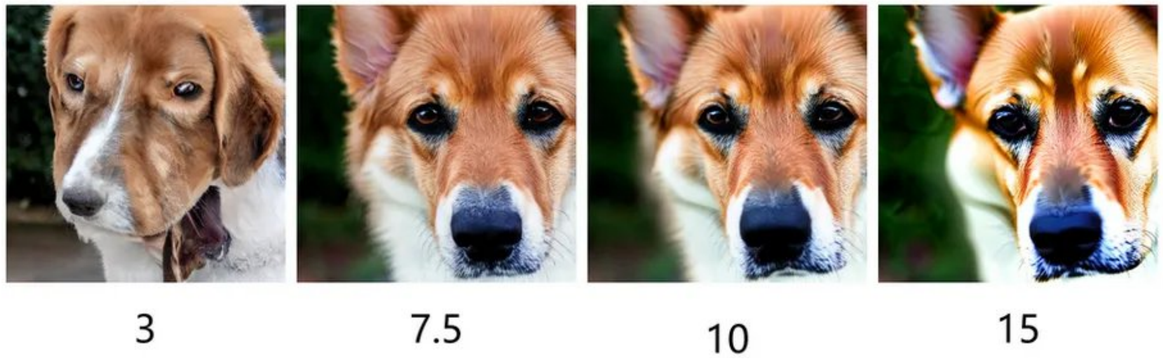
$\mathbf{z}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = 1, \ldots, T$ **do**
  ▷ Form the classifier-free guided score at log SNR λ_t
  $\tilde{\boldsymbol{\epsilon}}_t = (1 + w)\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, \mathbf{c}) - w\boldsymbol{\epsilon}_\theta(\mathbf{z}_t)$
  ▷ Sampling step (could be replaced by another sampler, e.g. DDIM)
  $\tilde{\mathbf{x}}_t = (\mathbf{z}_t - \sigma_{\lambda_t}\tilde{\boldsymbol{\epsilon}}_t)/\alpha_{\lambda_t}$
  $\mathbf{z}_{t+1} \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_{\lambda_{t+1}|\lambda_t}(\mathbf{z}_t, \tilde{\mathbf{x}}_t), (\tilde{\sigma}^2_{\lambda_{t+1}|\lambda_t})^{1-v}(\sigma^2_{\lambda_t|\lambda_{t+1}})^v)$ if $t < T$ else $\mathbf{z}_{t+1} = \tilde{\mathbf{x}}_t$
end for
return $\mathbf{z}_{T+1}$

Drawback: 2 forward passes per sampling step

Inference techniques

Classifier-free guidance

- Larger CFG scale value improves prompt alignment and fidelity
- However, excessively high CFG scales can lead to over-saturated images
- CFG scale reduces generational diversity
- Optimal CFG scale has to be determined manually for each model, sampler, and number of inference steps

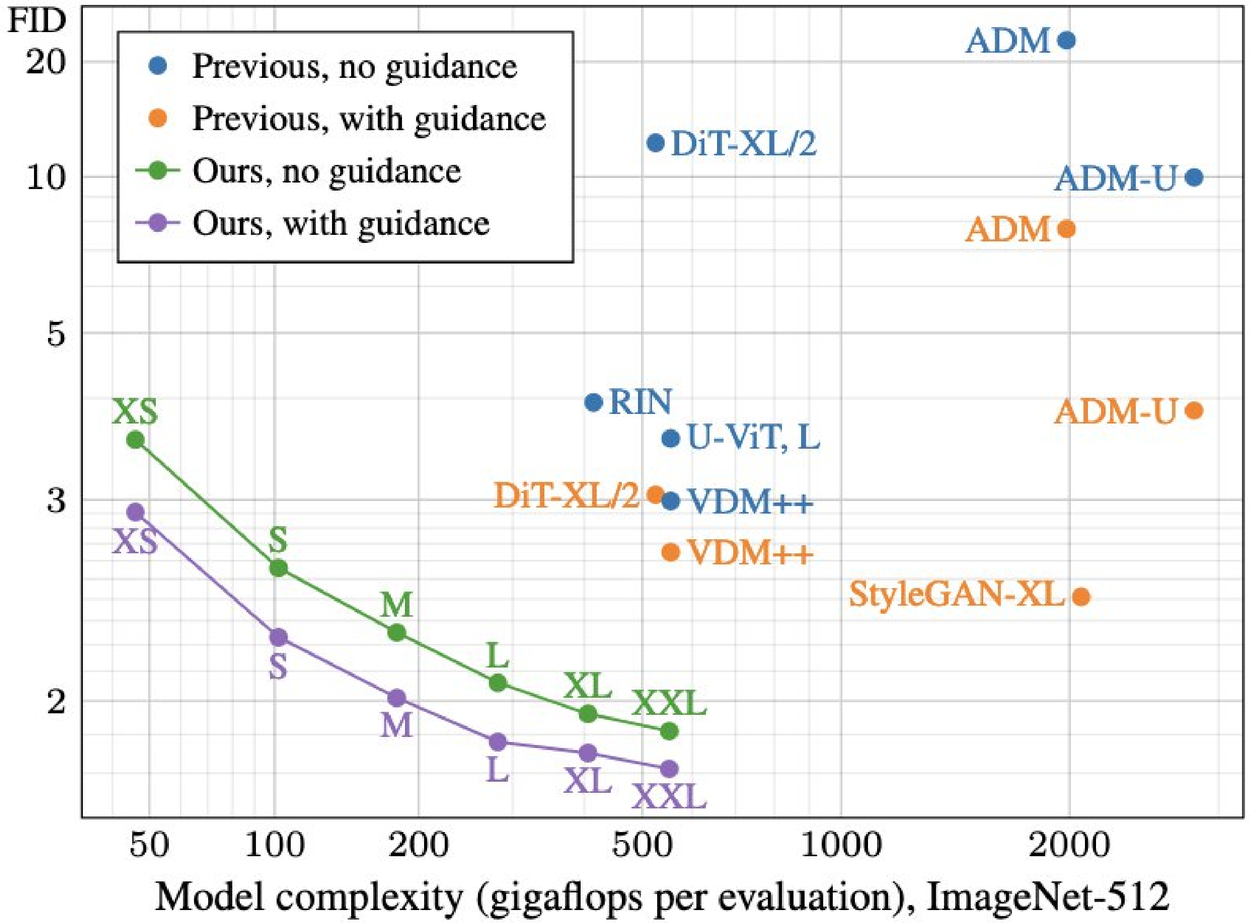


Dieleman, Guidance: a cheat code for diffusion models, Blogpost 2022

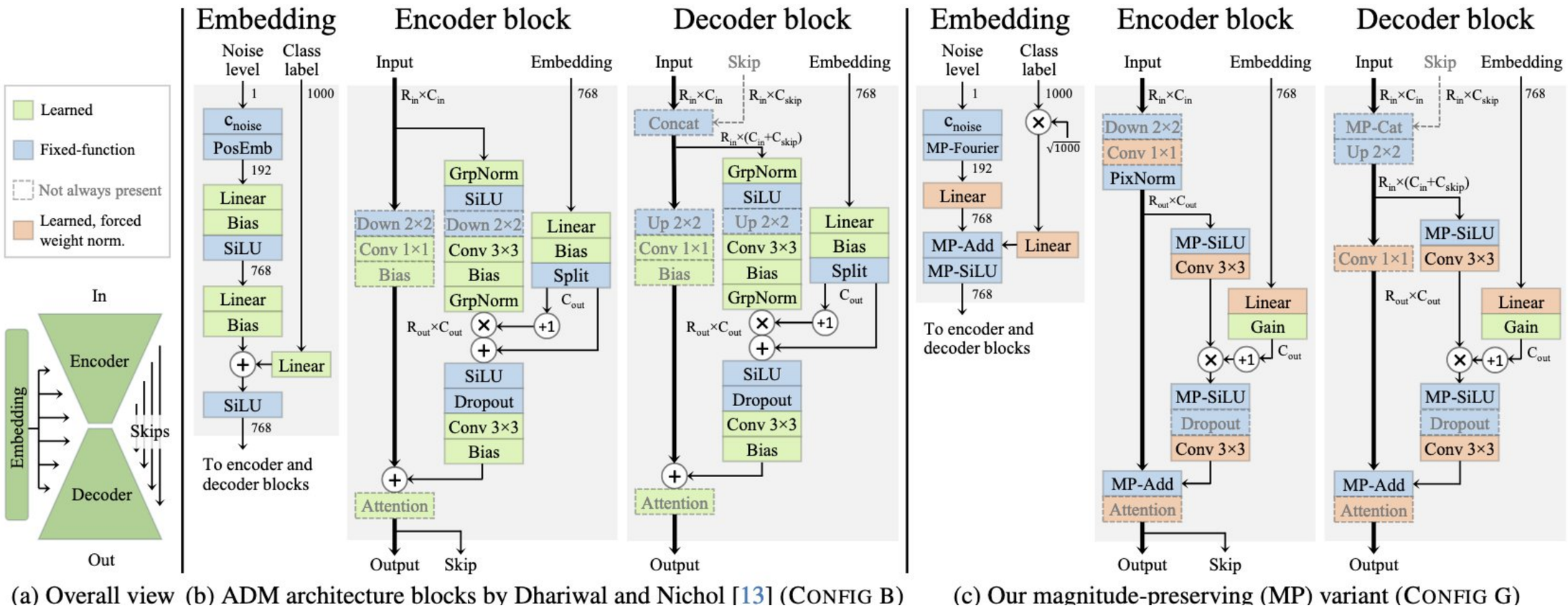
Training techniques

Analyzing and Improving the Training Dynamics of Diffusion Models

Examines the pitfalls of current diffusion training pipelines and architectures and proposes recipes for faster convergence and improved quality.



Training techniques



(a) Overall view (b) ADM architecture blocks by Dhariwal and Nichol [13] (CONFIG B) (c) Our magnitude-preserving (MP) variant (CONFIG G)

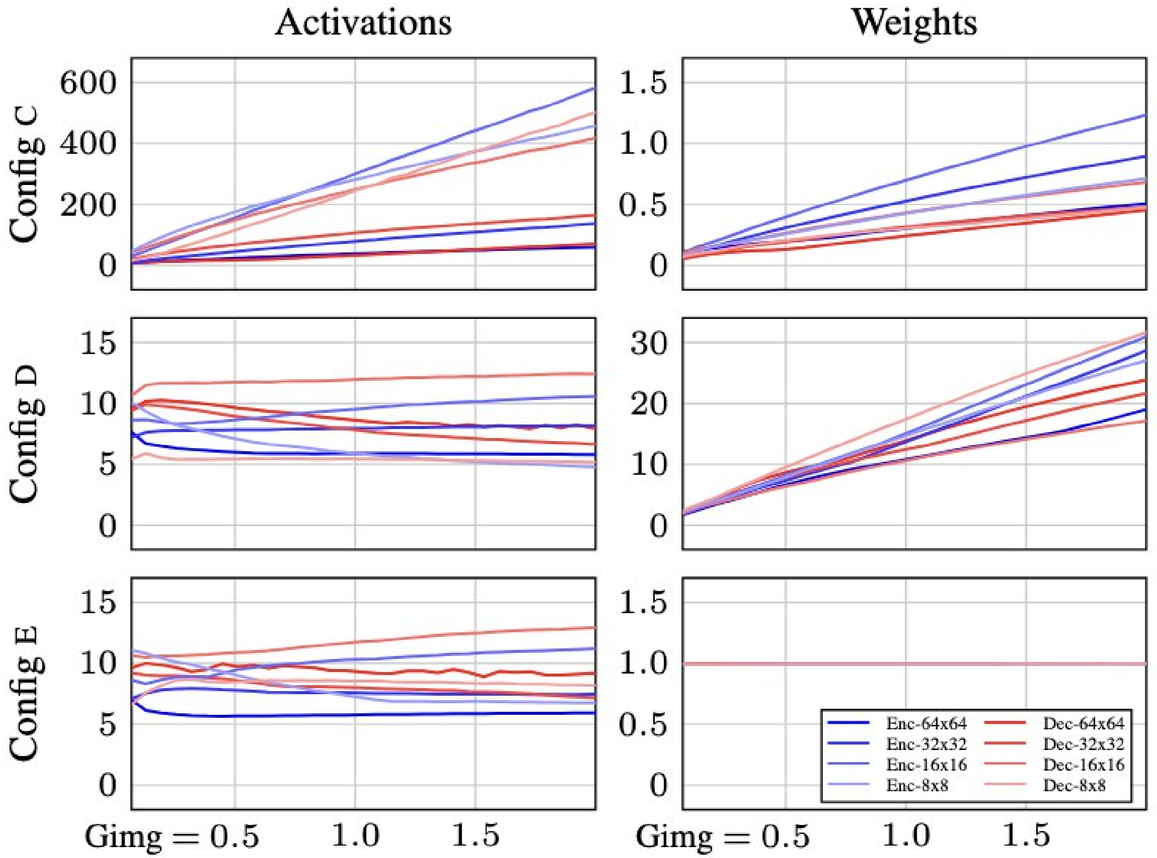
- Many studies adopt the UNet model from [Diffusion Models Beat GANs on Image Synthesis]
- However, it has some serious flaws.

Training techniques

- The authors noted that activations and weights can grow rapidly during training
- To address this issue, they proposed applying forced weight normalization:

$$\hat{\mathbf{w}}_i = \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|_2 + \epsilon}.$$

- In addition, SiLU activations and skip connection are scaled to preserve the magnitude
- These hacks boost quality dramatically



Training techniques

Training configurations, ImageNet-512	FID ↓	Mparams	Gflops
A EDM baseline	8.00	295.9	110.4
B + Minor improvements	7.24	291.8	100.4
C + Architectural streamlining	6.96	277.8	100.3
D + Magnitude-preserving learned layers	3.75	277.8	101.2
E + Control effective learning rate	3.02	277.8	101.2
F + Remove group normalizations	2.71	280.2	102.1
G + Magnitude-preserving fixed-function layers	**2.56**	280.2	102.2

Table 1. Effect of our changes evaluated on ImageNet-512. We report Fréchet inception distance (FID, lower is better) [20] without guidance, computed between 50,000 randomly generated images and the entire training set. Each number represents the minimum of three independent evaluations using the same model.

Enforcing magnitude preservation also allows us to eliminate group normalization.

Training (and inference) techniques

Exponential Moving Average is a DL technique to calculate a weighted average of current and past data points. This helps improve generalization and reduce parameter fluctuations

$$\mathrm{EMA}_t = \begin{cases} p_1 & \text{if } t = 1 \\ \alpha p_t + (1-\alpha)\mathrm{EMA}_{t-1} & \text{else} \end{cases}$$

Common values in academic works:

$$\alpha = 0.999,\ 0.9999$$

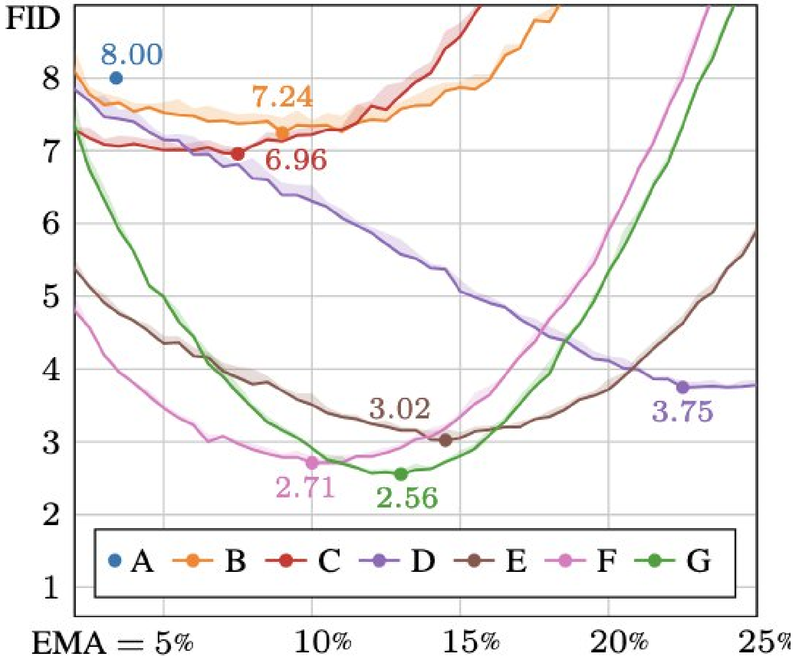
$$\mathrm{EMA}_t = \alpha \sum_{i=1}^{t} (1-\alpha)^{i-1} p_{t-i+1}$$

$$= \frac{1}{\frac{1}{\alpha}} \sum_{i=1}^{t} (1-\alpha)^{i-1} p_{t-i+1}$$

Common values in academic works:

$$t_{\text{decay}} = 10^3,\ 10^4$$

Training (and inference) techniques



(a) FID vs. EMA for each training config

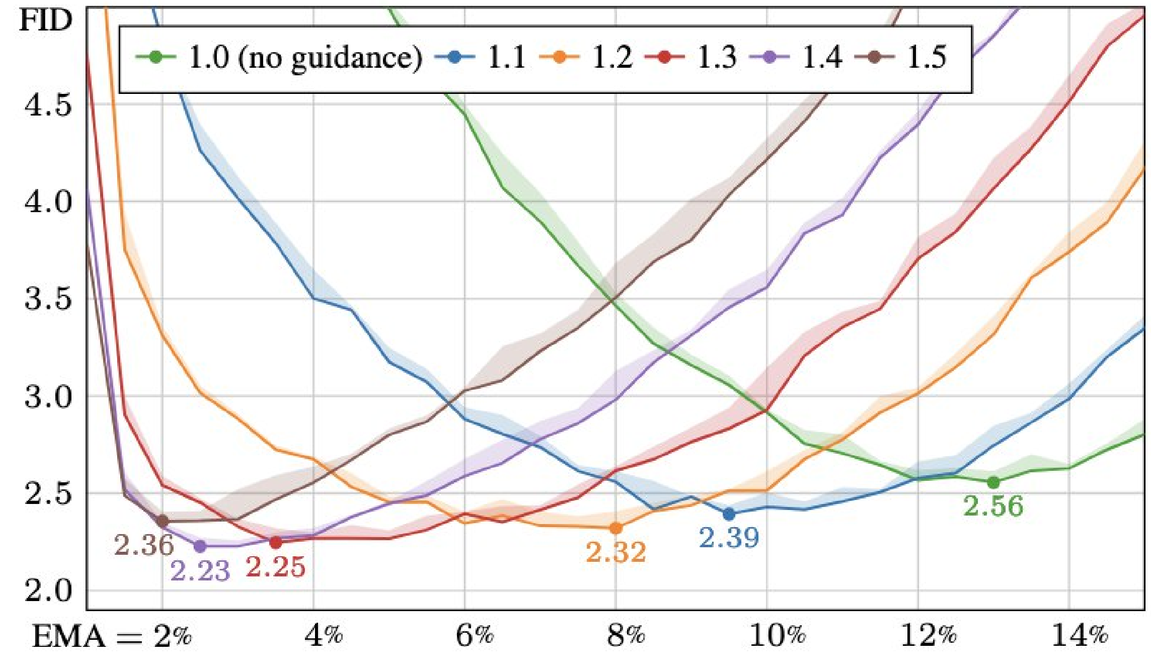


Figure 6. Interaction between EMA length and guidance strength using EDM2-S on ImageNet-512.

- EMA length has a dramatic impact on performance
- For best performance, set the EMA together with the CFG scale

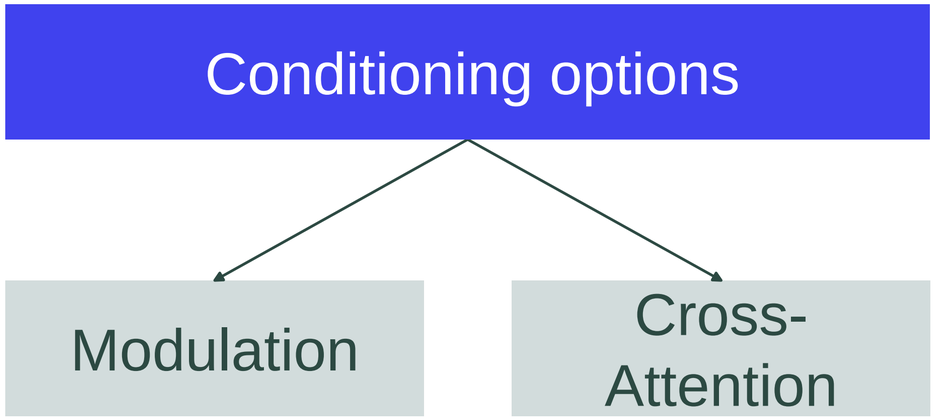
Training techniques

ImageNet-512		FID ↓		Model size		
		no CFG	w/CFG	Mparams	Gflops	NFE
ADM	[13]	23.24	7.72	559	1983	250
DiT-XL/2	[58]	12.03	3.04	675	525	250
ADM-U	[13]	9.96	3.85	730	2813	250
RIN	[31]	3.95	–	320	415	1000
U-ViT, L	[28]	3.54	3.02	2455	555*	256
VDM++	[39]	2.99	2.65	2455	555*	256
StyleGAN-XL	[73]	–	2.41	168*	2067*	1
EDM2-XS		3.53	2.91	125	46	63
EDM2-S		2.56	2.23	280	102	63
EDM2-M		2.25	2.01	498	181	63
EDM2-L		2.06	1.88	777	282	63
EDM2-XL		1.96	1.85	1119	406	63
EDM2-XXL		**1.91**	**1.81**	1523	552	63

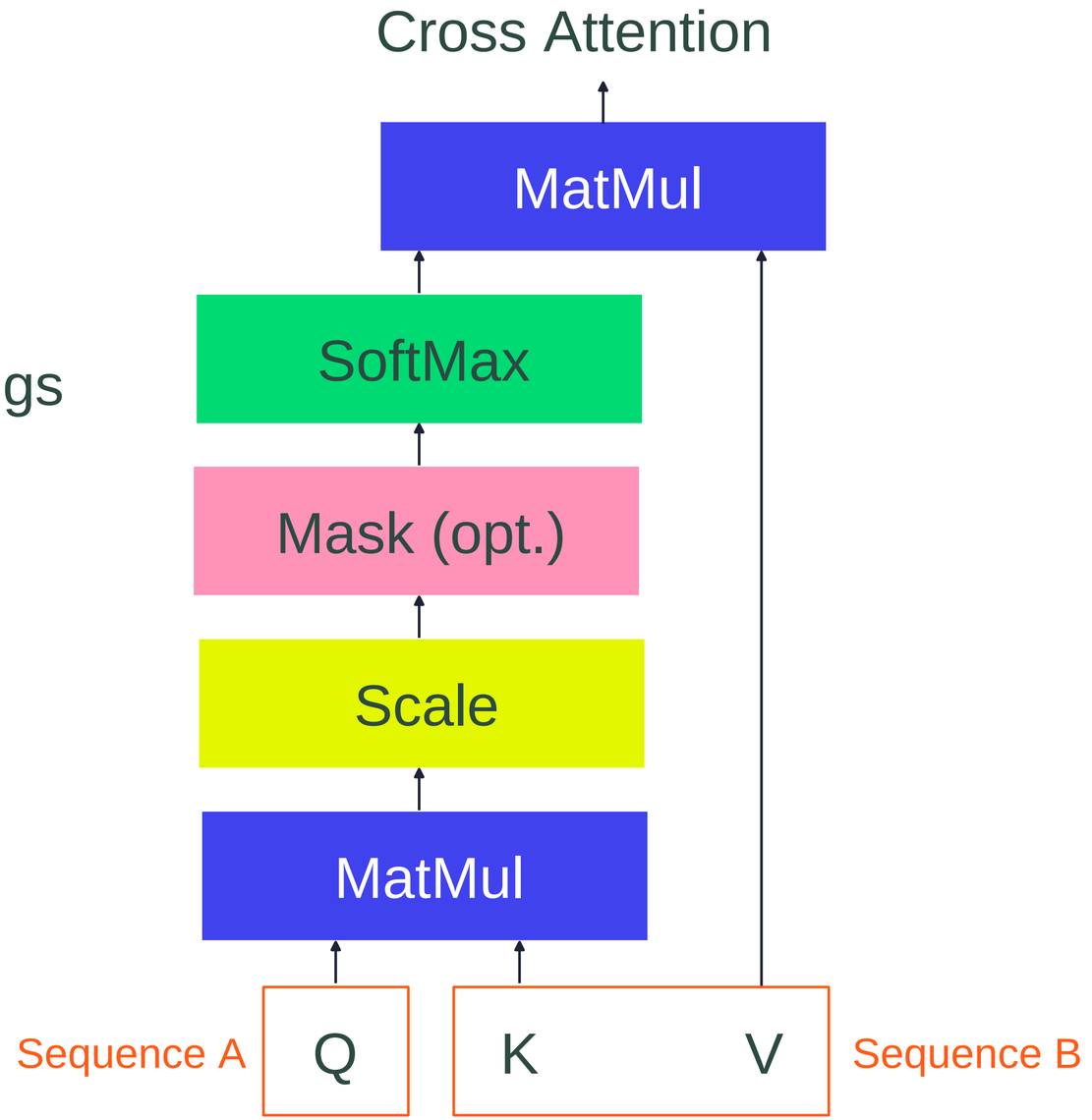
Proposed training recipes achieved new SOTA on ImageNet-512 class-conditional generation

Text-to-image generation

- Goal: condition the image generation process on a given text prompt
- LLM and CLIPs can produce decent text embeddings
- The output is a sequence of tokens (vectors)

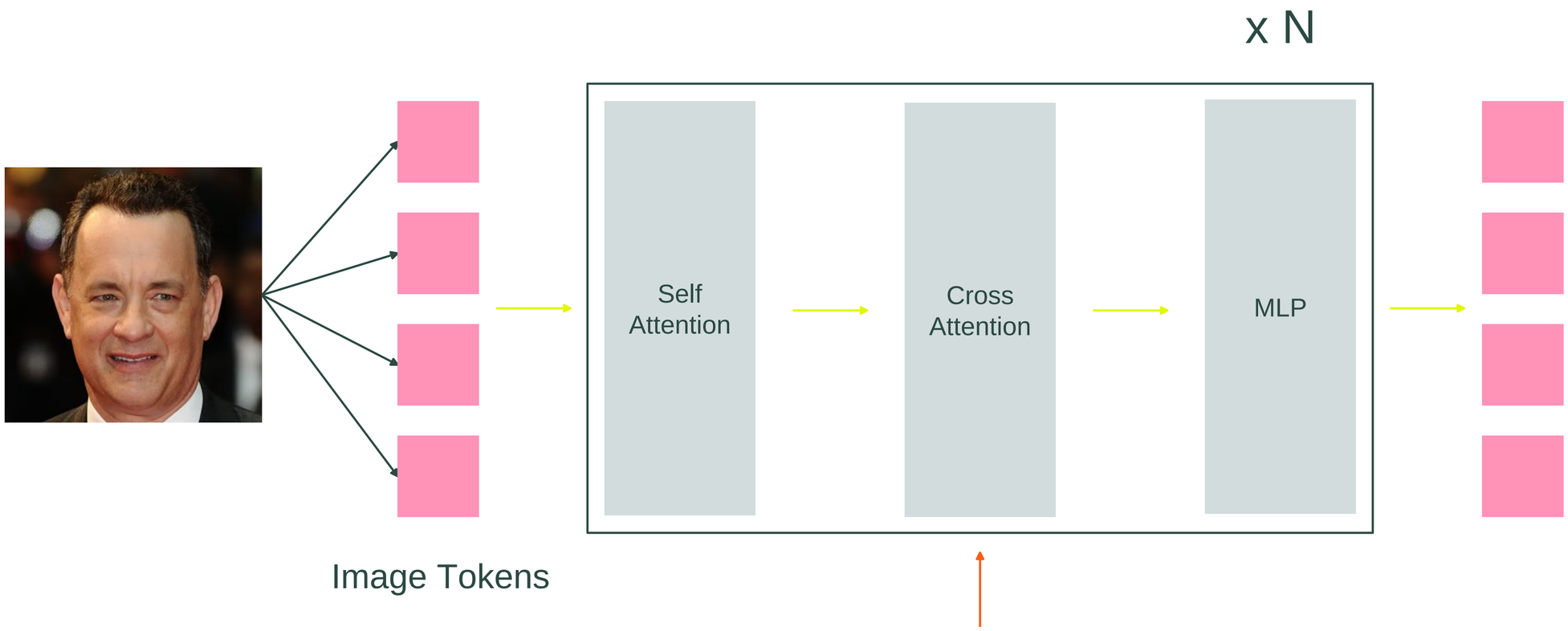


Same as time conditioning



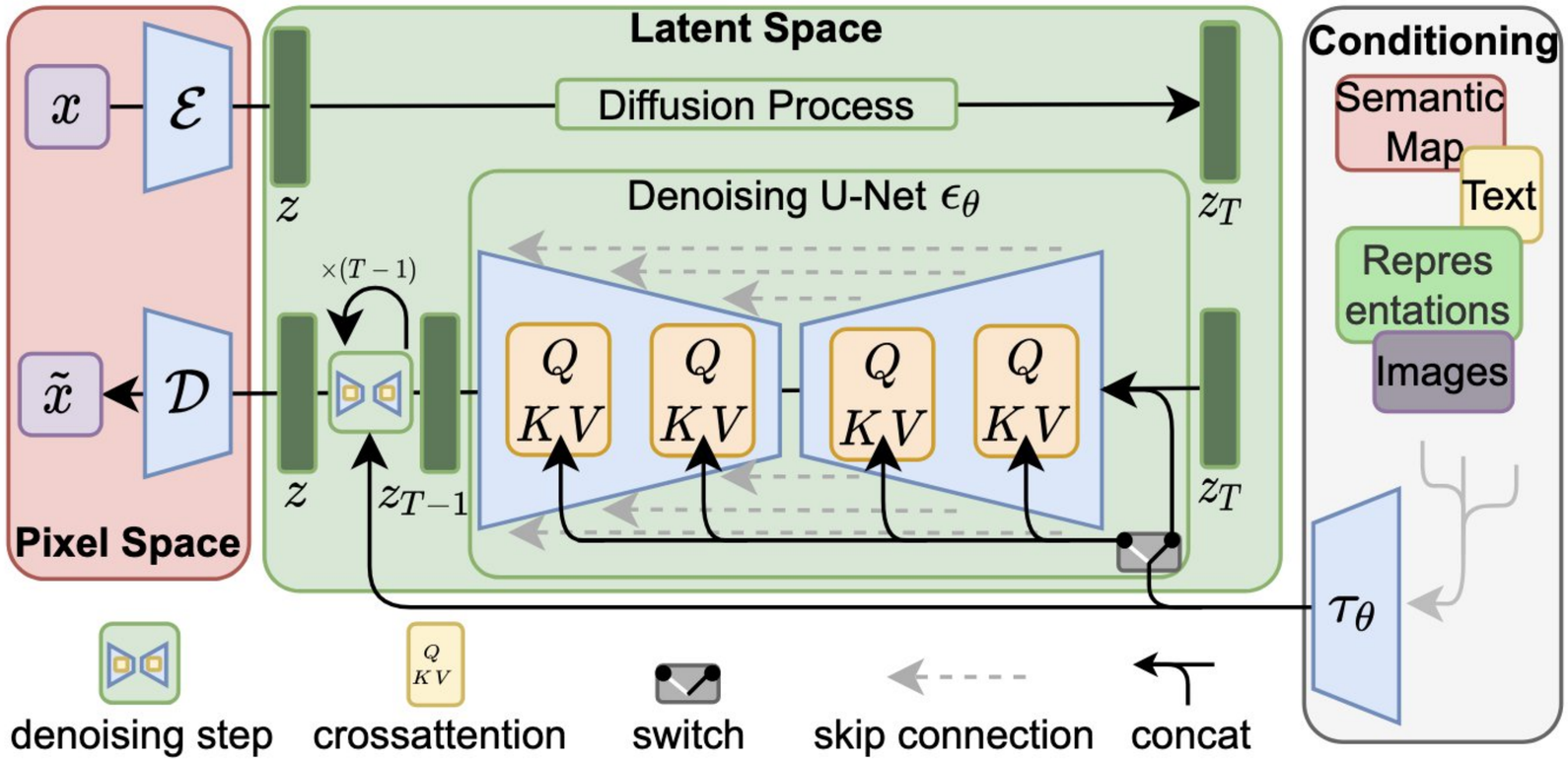
The resulting training recipe achieved new SOTA on ImageNet-512 class-conditional generation

Text-to-image generation



Text-to-image generation

Illustration of the text-to-image generation process using an LDM



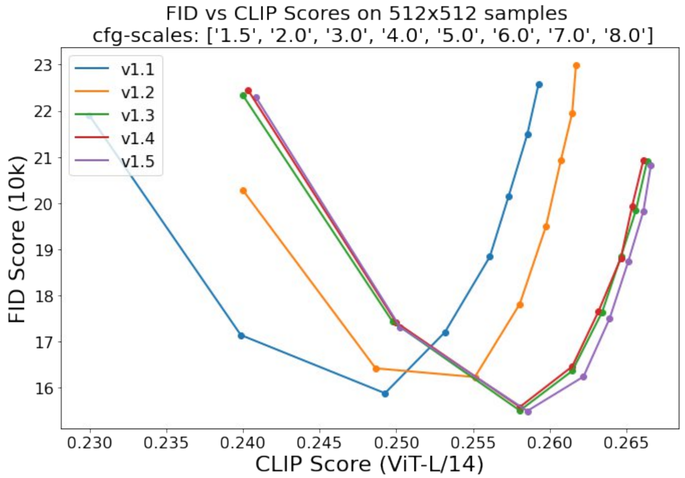
Rombach et al., “High-Resolution Image Synthesis with Latent Diffusion Models”, CVPR 2022.

Examples of T2I diffusion models

Stable Diffusion v 1.4/1.5

- Trained on LAION-2B
- Generates 512x512 images
- Architecture
  — UNet (860M parameters)
  — Text CLIP encoder (120M parameters)
  — 4-channel VAE with x8 downsampling
- 150’000 A100 GPU training hours

better alignment to distribution



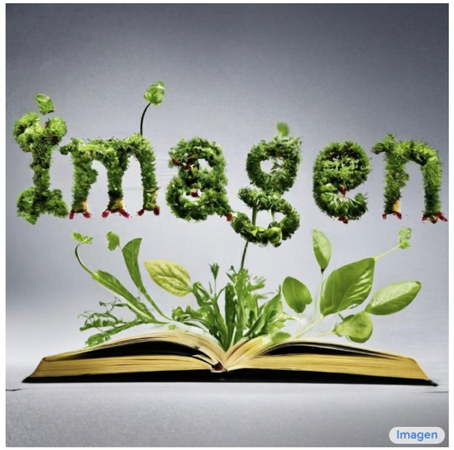
better alignment to text

Rombach et al., “High-Resolution Image Synthesis with Latent Diffusion Models”, CVPR 2022.

Examples of T2I diffusion models

Imagen

- Cascaded architecture:
  — Base 64 px
  — Super Resolution 64 → 256 px
  — Super Resolution 256 → 1024 px
- Large text encoder T5-XXL
- EfficientUNet: more parameters in deeper layers, less parameters in high-resolution layers



Sprouts in the shape of text ‘Imagen’ coming out of a fairytale book.

A photo of a Shiba Inu dog with a backpack riding a bike. It is wearing sunglasses and a beach hat.

A high contrast portrait of a very happy fuzzy panda dressed as a chef in a high end kitchen making dough. There is a painting of flowers on the wall behind him.

Teddy bears swimming at the Olympics 400m Butterfly event.

A cute corgi lives in a house made out of sushi.

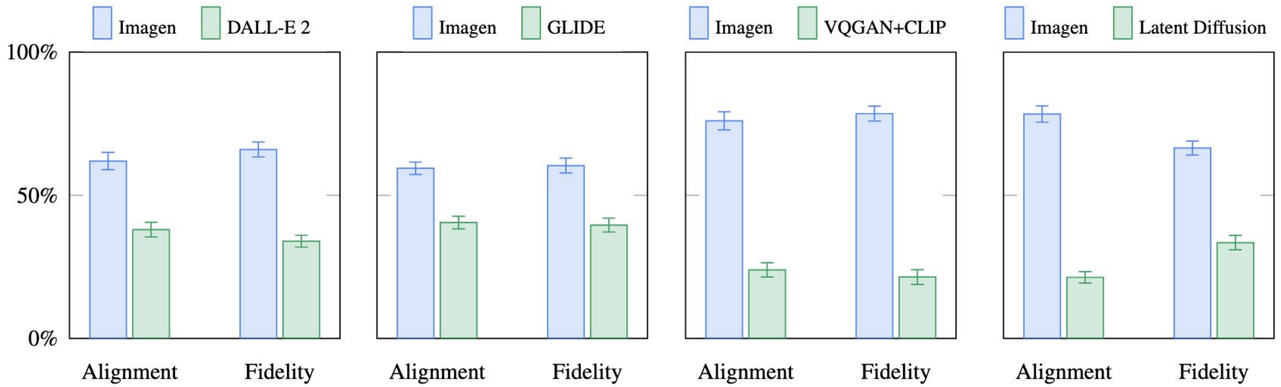
A cute sloth holding a small treasure chest. A bright golden glow is coming from the chest.

Saharia C. et al., Photorealistic text-to-image diffusion models with deep language understanding, NeuirIPS 2022

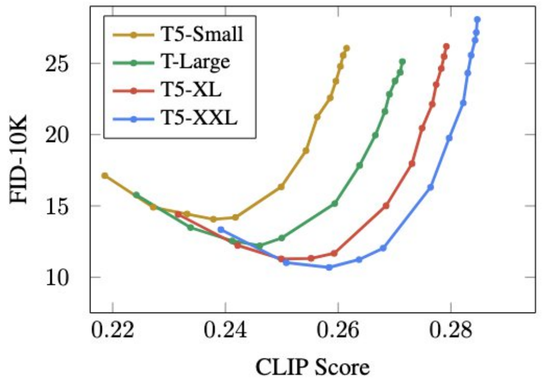
Examples of T2I diffusion models

Imagen

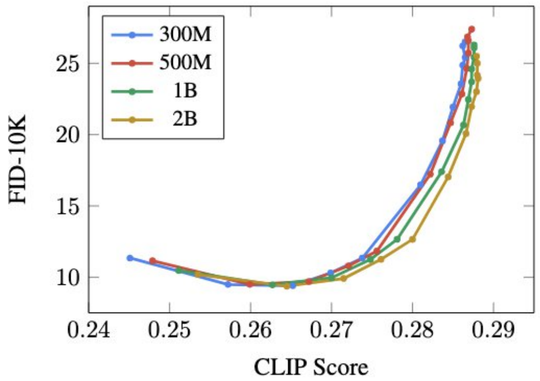
Claimed to be SOTA on side-by-side comparisons



Scaling the text encoder is more important than the model size



(a) Impact of encoder size.



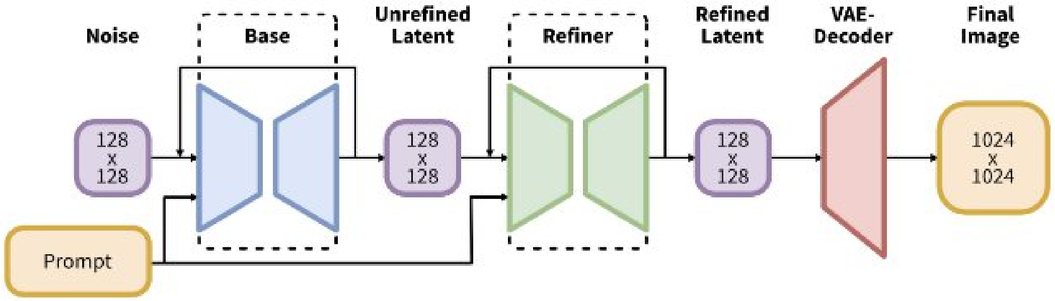
(b) Impact of U-Net size.

Saharia C. et al., Photorealistic text-to-image diffusion models with deep language understanding, NeuirIPS 2022

Examples of T2I diffusion models

SDXL

- Generate images up to 1024x1024 and rectangular images (4:3, 3:4, 16:9, 9:16)
- Improved VAE
- Architecture
  — Scaled UNet (~2.6B parameters)
  — Updated model structure (x16 downsampling stage removed)
  — More transformer blocks
  — 2 text encoders (CLIP ViT-L & OpenCLIP ViT-bigG)
  — Refiner module
  — Crop and image size conditioning during training

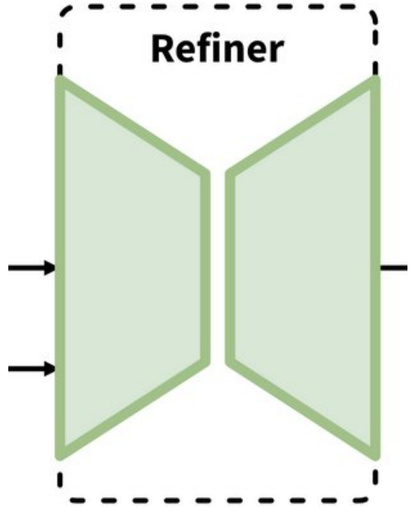


Podell D. et al., Sdxl: Improving latent diffusion models for high-resolution image synthesis, arXiv:2307.01952

Examples of T2I diffusion models

SDXL

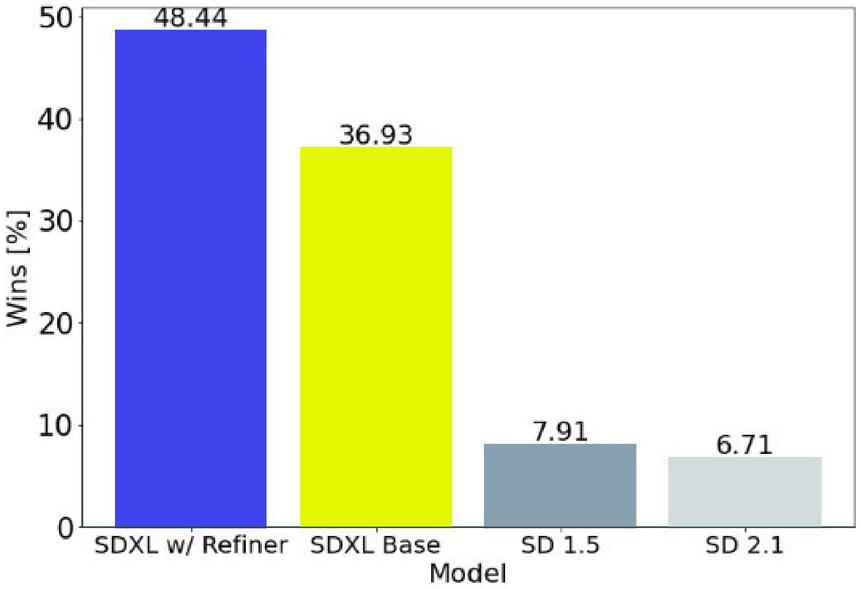
- Images produced by the original model are noised back (to a relatively small noise level)
- The refiner model, a separate, prompt-conditioned diffusion model, is used to denoise the generated image at the final diffusion steps
- While it may increase inference costs, it produces higher-quality images



Podell D. et al., Sdxl: Improving latent diffusion models for high-resolution image synthesis, arXiv:2307.01952

Examples of T2I diffusion models

SDXL



User preference study

Left — w/o Refiner, right — w/ Refiner

Podell D. et al., Sdxl: Improving latent diffusion models for high-resolution image synthesis, arXiv:2307.01952

Examples of T2I diffusion models

Stable Diffusion 3

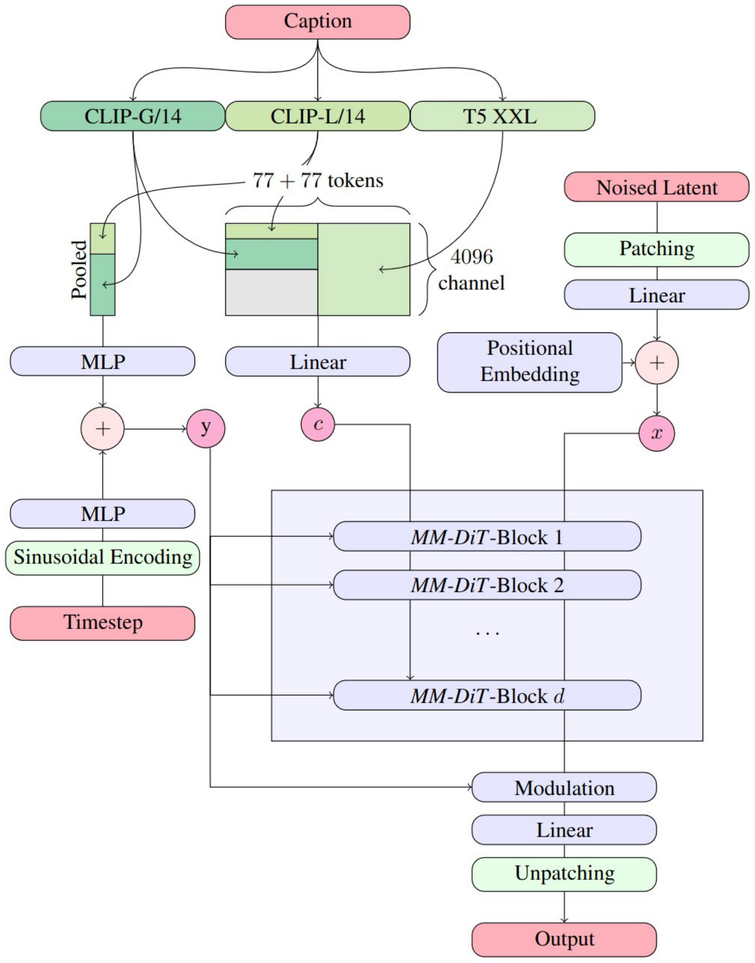
- DiT instead of UNet
- Largest model: 8B parameters
- 3! text encoders
- Concatenation of image and text tokens into a single sequence inside Attention
- 16-channel VAE
- Separate diffusion process
- QK-normalization in Attention

Only 2B model is open-sourced, and it is far from being amazing :(

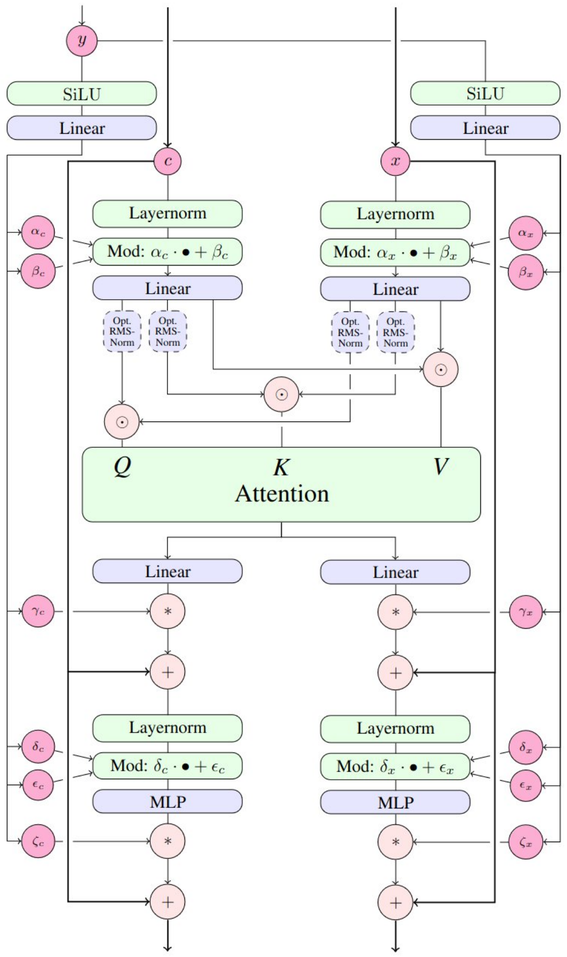
Esser P. et al., Scaling rectified flow transformers for high-resolution image synthesis. ICML 2024

Examples of T2I diffusion models

Stable Diffusion 3



(a) Overview of all components.

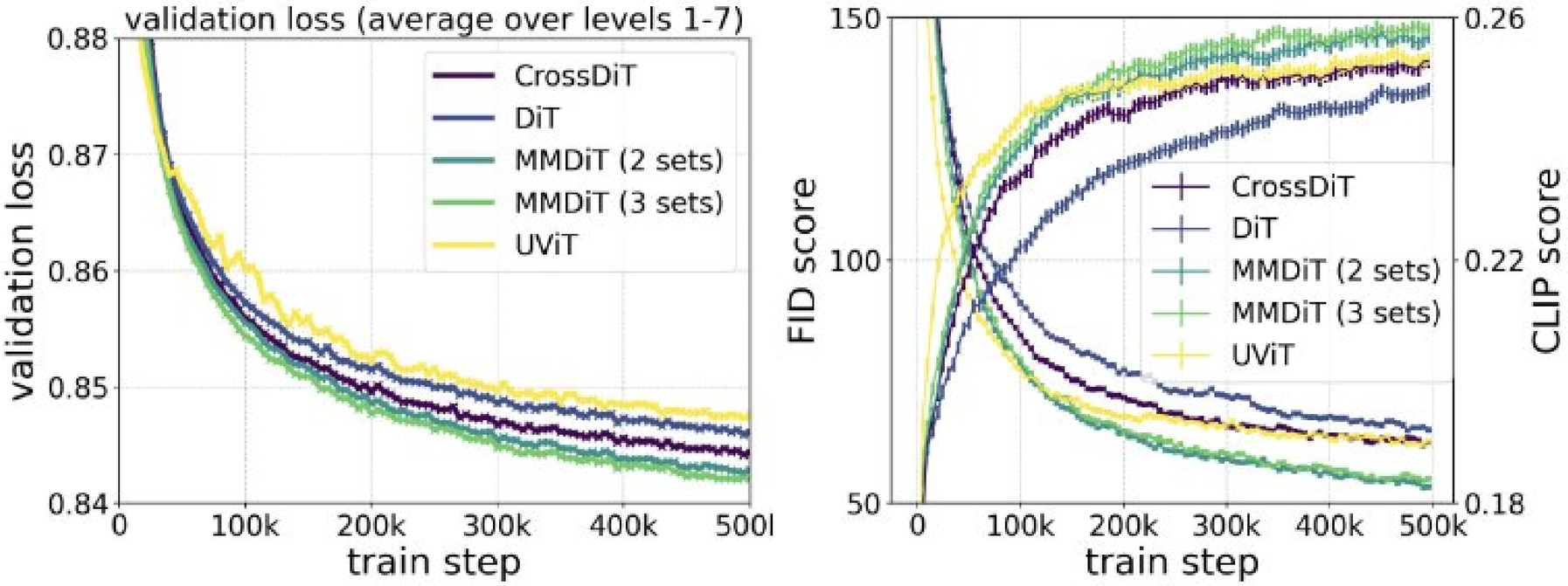


(b) One *MM-DiT* block

Esser P. et al., Scaling rectified flow transformers for high-resolution image synthesis. ICML 2024

Examples of T2I diffusion models

Stable Diffusion 3



Architecture tricks are beneficial

Esser P. et al., Scaling rectified flow transformers for high-resolution image synthesis. ICML 2024

Examples of T2I diffusion models

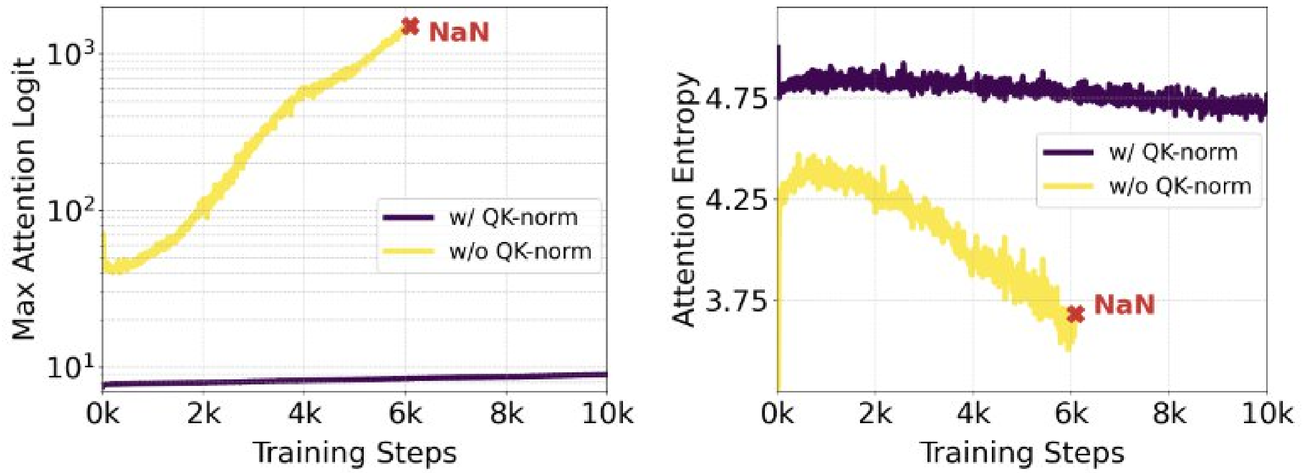
Stable Diffusion 3

- In QK-normalized attention, queries and keys are normalized to a unit norm before attention operation

$$\widehat{Q} = Q/\|Q\|$$

$$\widehat{K} = K/\|K\|$$

$$A = \text{softmax}\left(\frac{\widehat{Q}\widehat{K}^\top}{\sqrt{d}}\right)$$

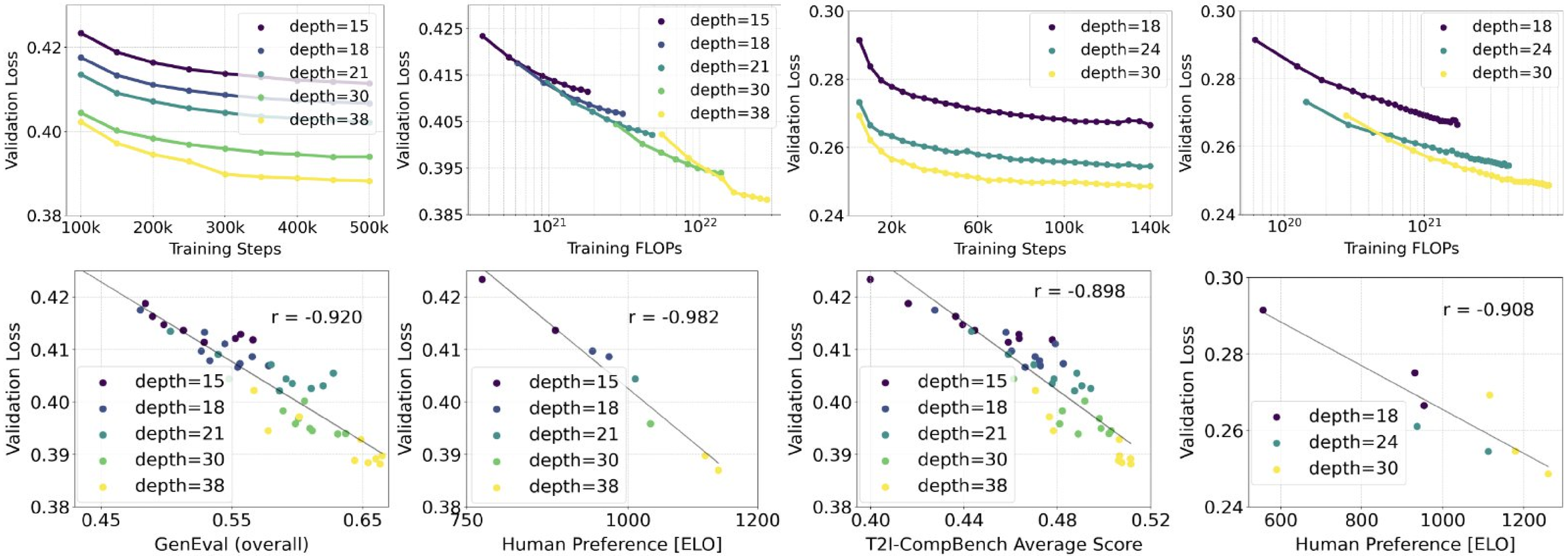


QK-normalization is crucial for training stability

Esser P. et al., Scaling rectified flow transformers for high-resolution image synthesis. ICML 2024

Examples of T2I diffusion models

Stable Diffusion 3

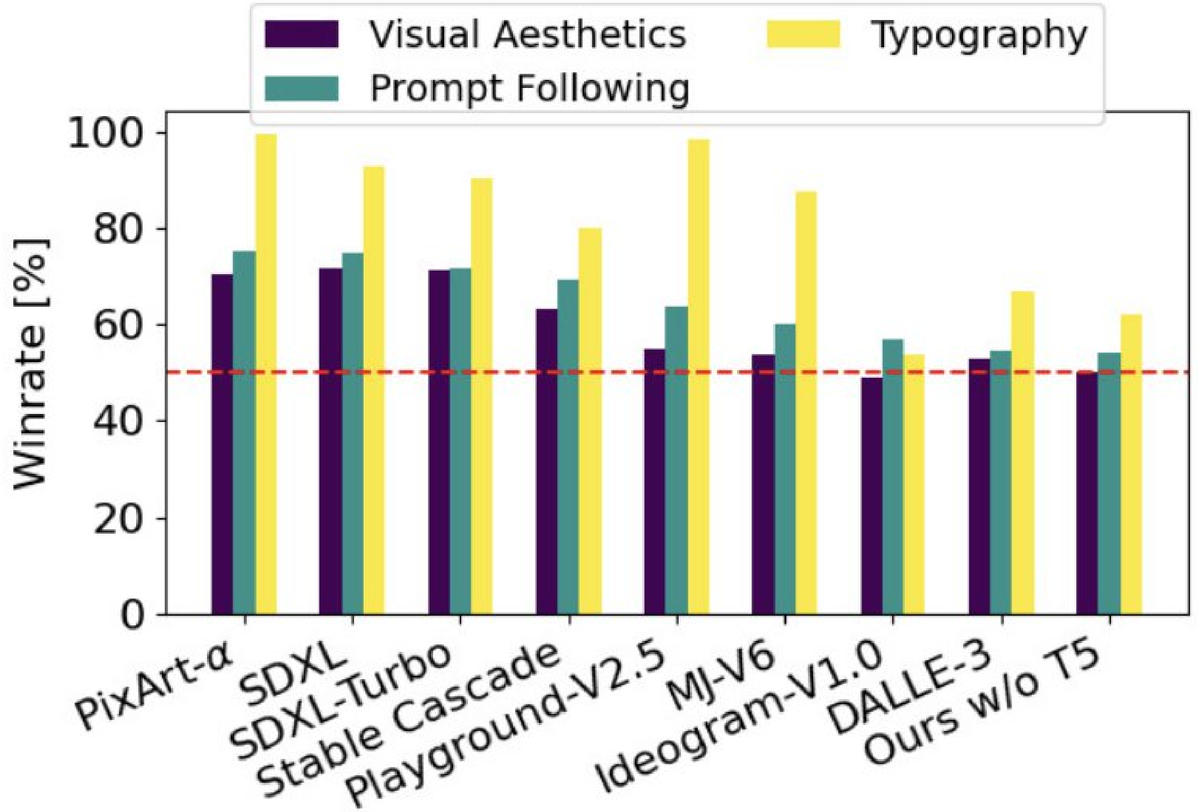


Scaling improves quality

Esser P. et al., Scaling rectified flow transformers for high-resolution image synthesis. ICML 2024

Examples of T2I diffusion models

Stable Diffusion 3



MJ “Killer”

Esser P. et al., Scaling rectified flow transformers for high-resolution image synthesis. ICML 2024

Examples of T2I diffusion models

FLUX

- From creators of Stable Diffusion…
- DiT with 12B parameters
- Same diffusion process as in SD3
- 3 version:
  — [pro] — closed-source
  — [dev] — standard
  — [schnell] – faster (fewer sampling steps)

blackforestlabs.ai

Examples of T2I diffusion models

FLUX

Generation examples

blackforestlabs.ai

Школа анализа данных

Thanks for

$$\text{softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right)$$