Generative AI for Image Synthesis

Dr. Aliaksei Mikhailiuk

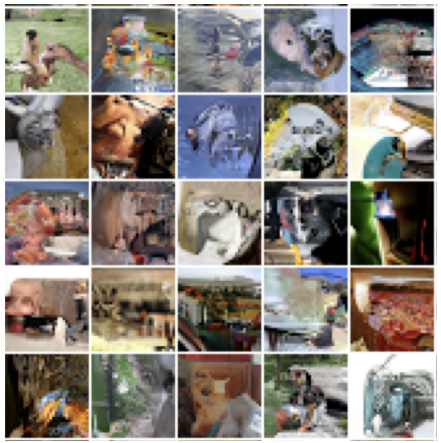
What this talk is about?

1) How did we get here? - Evolution of generative methods for image synthesis
2) What is the state-ot-the-art? - Current industry standard
3) Where do we go from here? - Beyond image synthesis
4) What challenges do we need to overcome to get where we want to get?

How did we get here?

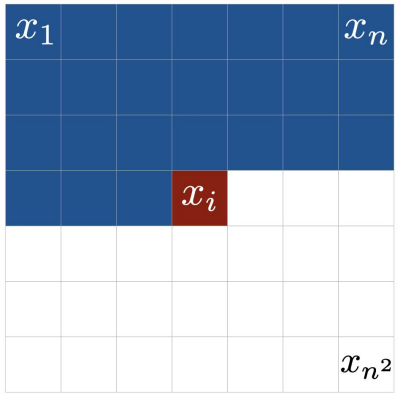
Historical perspective

(early–2010s) Autoregressive Models
(2013 onward) Variational Autoencoders (VAEs)
(2014 onward) Generative Adversarial Networks (GANs)
(2015 onward) Normalizing Flows
(2019 onward) Diffusion & Score-Based Models
(2023 onward) Flow Matching & Flux-type Models

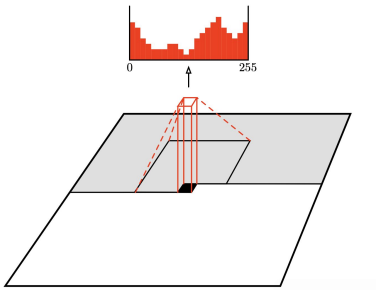


Autoregressive Models (early–2010s)

- **Core idea:** Generate an image pixel-by-pixel (or patch-by-patch), conditioning each new element on the previously generated ones.
- **Examples:**
  - PixelRNN, PixelCNN [1]
  - Conditional PixelCNN [2]
- **Strengths:** Exact likelihood, sharp local structure.
- **Limitations:** Slow sampling, struggles with global coherence.



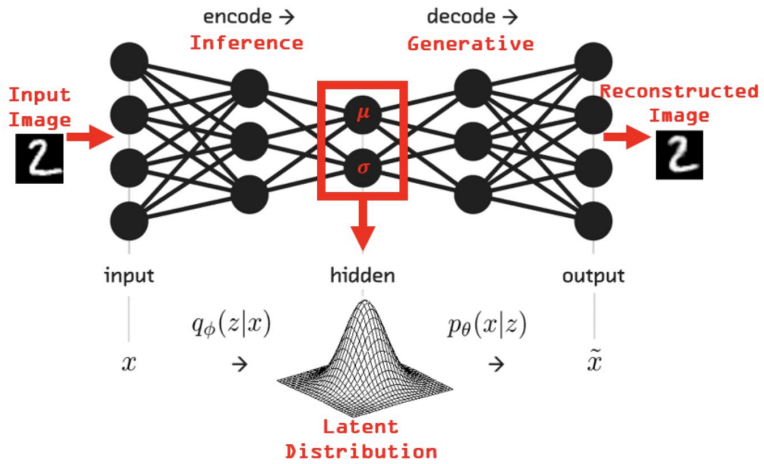
$$p(\mathbf{x}) = \prod_{i=1}^{n^2} p(x_i | x_1, ..., x_{i-1})$$



[1] Oord et. al, 2016. “Pixel recurrent neural networks”. ICML
[2] Oord et. al, 2016. “Conditional image generation with PixelCNN decoders”. NIPS

Variational Autoencoders (VAEs) (2013)

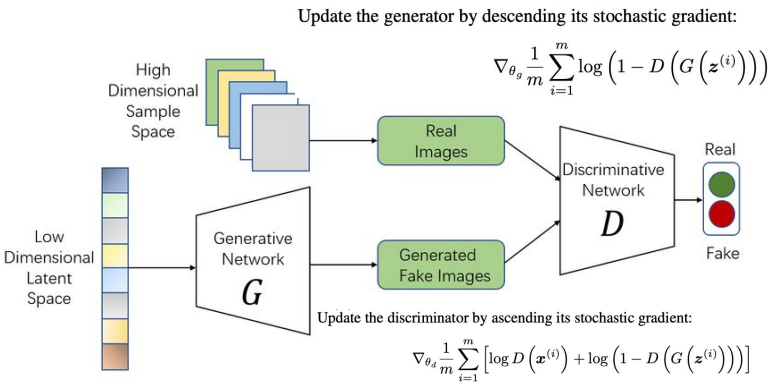
- **Core idea:** Learn a compressed latent space + probabilistic decoding.
- **Examples:**
  - Kingma & Welling's VAE [1]
  - VQ-VAE [2] & VQ-VAE-2 [3]
- **Strengths:** Efficient latent representations
- **Limitations:** Blurry generations.



[1] Kingma et.al., 2014 Auto-Encoding Variational Bayes, ICLR
[2] Oord et.al 2017, "Neural Discrete Representation Learning", NeurIPS
[3] Razavi, et.al, 2019, "Generating Diverse High-Fidelity Images with VQ-VAE-2", NeurIPS

Generative Adversarial Networks (GANs) (2014)

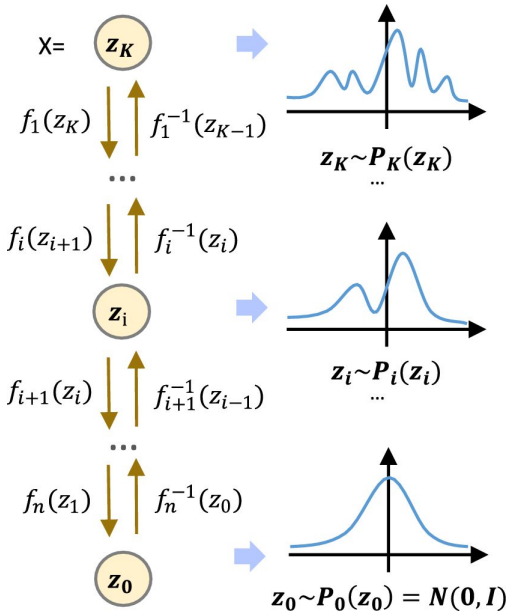
- **Core idea:** A generator vs. discriminator game [1]
- **Milestones:**
  - DCGAN [2]
  - Progressive GANs [3]
  - StyleGAN [4], StyleGAN-NADA [5]
- **Strengths:** High-quality, realistic images.
- **Limitations:** Training instability, mode collapse, hard likelihood estimation, conditioning challenges



[1] Goodfellow et.al. , 2014. “Generative adversarial networks”, Comms. ACM
[2] Radford et.al, 2015. “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, ICLR
[3] Karras et.al, 2018. “Progressive Growing of GANs for Improved Quality, Stability, and Variation”, ICLR
[4] Karras et.al. 2019. “A Style-Based Generator Architecture for Generative Adversarial Networks”, CVPR
[5] Gal et.al. 2022 “StyleGAN-NADA: CLIP-Guided Domain Adaptation of Image Generators”, ACM ToG

Normalizing Flows (2015)

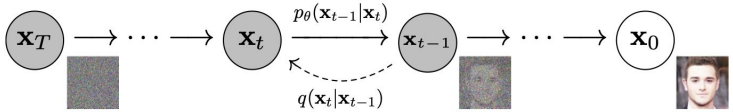
- **Core idea:** Exact likelihood modeling via invertible transformations.
- **Examples:**
  - RealNVP [1]
  - Glow [2]
  - TARFLOW [3]
- **Strengths:** Exact log-likelihood, invertible mapping between data and latent space.
- **Limitations:** High memory/computation cost, weaker sample quality compared to GANs.



[1] Dinh et.al., 2017, “Density Estimation Using Real NVP”, ICLR
[2] Kingma et.al, 2018, “Glow: Generative Flow with Invertible 1×1 Convolutions”, NeurIPS
[3] Zhai et.al. , 2025. “Normalizing Flows are Capable Generative Models”, ICML

Diffusion & Score-Based Models (2019)

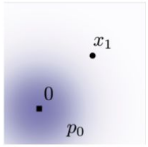
- **Core idea:** Learn to reverse a gradual noise process to generate clean images.
- **Examples:**
  - DDPM [1]
  - Score-based generative models [2]
  - LDM [3]
- **Strengths:** State-of-the-art quality, stable training, flexible conditioning.
- **Limitations:** Slow sampling.

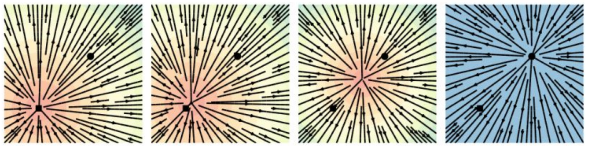


[1] Ho et.al, 2020, “Denoising Diffusion Probabilistic Models”, NeurIPS
[2] Song et.al. 2021, “Score-based Generative Modeling Through Stochastic Differential Equations“, ICLR
[3] Rombach et.al, 2022, “High-Resolution Image Synthesis with Latent Diffusion Models”, CVPR

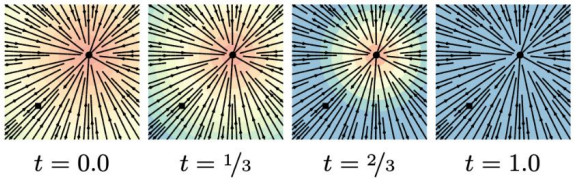
Flow Matching & Flux-type Models (2023)

- **Core idea:** Unify ODE-based generative modeling (e.g., flows & diffusion) with efficient training via *flow matching* or *rectified flows*.
- **Examples:**
  - Flow Matching Models [1]
  - Rectified Flow Transformers [2]
  - Flux [3]
- **Strengths:** Faster generation, continuous bridge between data & noise distributions.
- **Limitations:** Still emerging, not as widely benchmarked as diffusion yet.

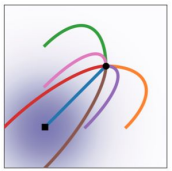




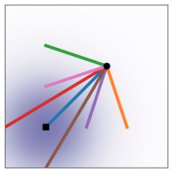
Diffusion path – conditional score function



OT path – conditional vector field



Diffusion



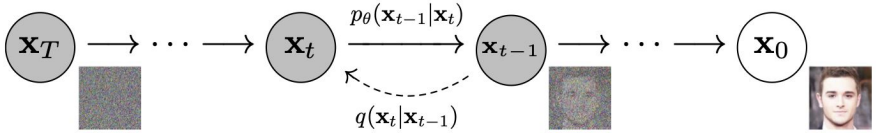
OT

[1] Lipman et.al ,2023, “Flow Matching For Generative Modeling” ICLR
[2] Esser et.al. 2024, “Scaling Rectified Flow Transformers for High-Resolution Image Synthesis”
[3] Black Forest Labs, 2025. “FLUX.1 Kontext: Flow Matching for In-Context Image Generation and Editing in Latent Space”, arxiv
[4] https://mlg.eng.cam.ac.uk/blog/2024/01/20/flow-matching.html

Where are we now?
Diffusion Models

Image based diffusion models

- **Two stage training -** corruption and reconstruction
- **In the corruption stage:** sequentially add noise to an image;
- **In reconstruction stage:** learn to reconstruct the added noise at each step;
  - Because we control the noise generation process, we have the exact formula for the noisy image at any given time step t.
- **During inference:** start with a noisy image and gradually reconstruct:
  - The quality of the final image depends on the number of steps we take in the noise removal process.



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

$$L_{\text{simple}}(\theta) \coloneqq \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right]$$

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

Recommended tutorial: https://www.chenyang.co/diffusion.html

Inference and choosing the right noise schedule

- The way we add noise impacts how well we can reconstruct the image:
  - The bigger the steps, the harder it is to accurately predict the noise;
  - The smaller the noise, the more steps we need to denoise.
- How much noise shall we add at each step?
  - Shall we add less noise in the earlier stages and more in the later?
  - Or shall we add more noise in the earlier and less in the later?
- This is an active area of research.
- Common schedules: LogLinear, DDPM, LDM

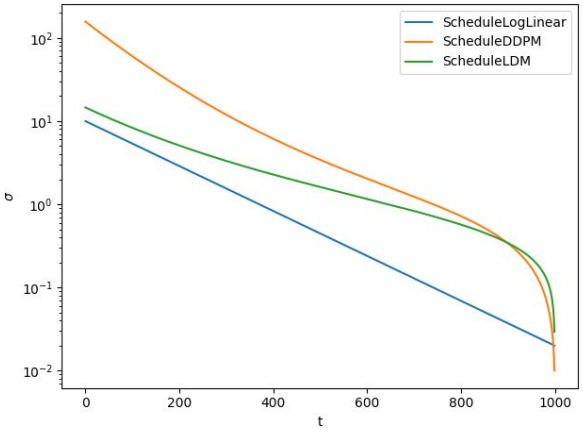


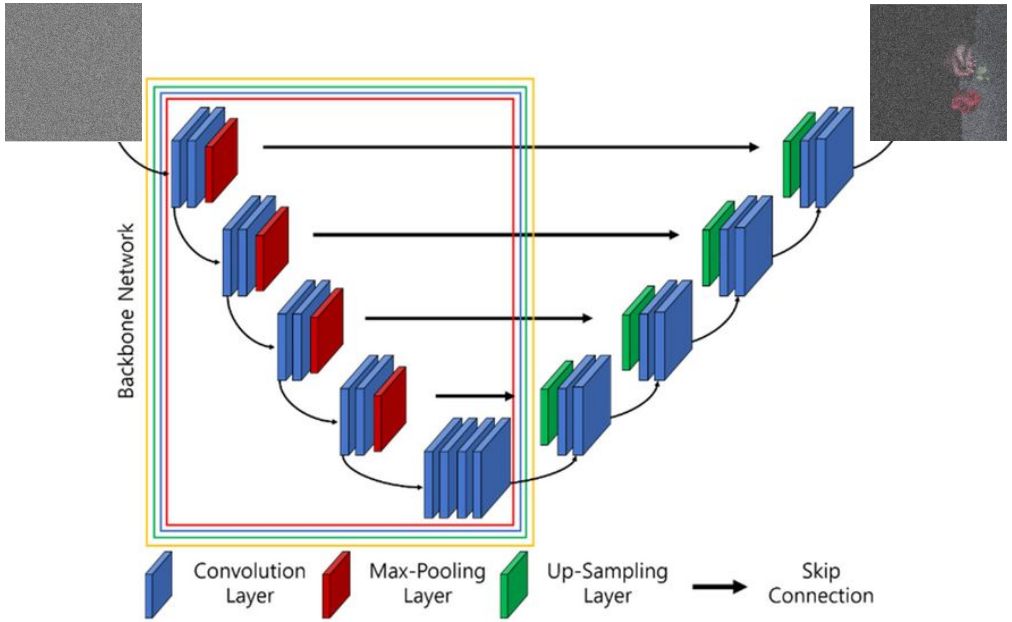
Image based latent denoising diffusion models

In practice we train a U-Net architecture to predict amount of noise at each timestep

repeat
$\mathbf{x}_0 \sim q(\mathbf{x}_0)$
$t \sim \text{Uniform}(\{1, \dots, T\})$
$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
Take gradient descent step on
$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$
until converged

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

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



Controlling generation

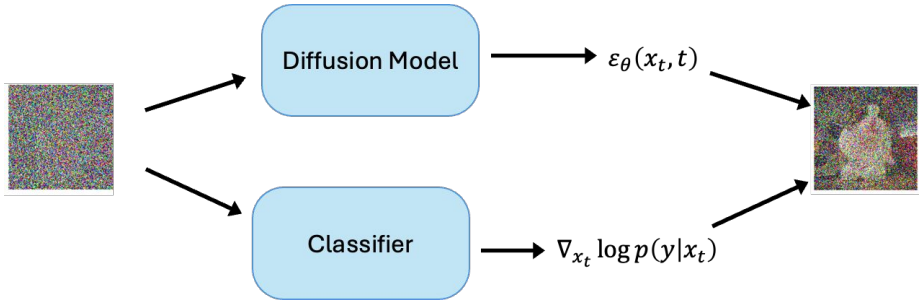
Classifier guidance

Classifier guidance:

- Use a separate pre-trained classifier gradients as guidance in denoising process.
- Issues:
  - Need a separate classifier trained on noisy images;
  - Classifier gradients perturb the image;
  - Limited number of classes.

Algorithm 1 Classifier guided diffusion sampling, given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, classifier $p_\phi(y|x_t)$, and gradient scale s.

Input: class label y, gradient scale s
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
for all t from T to 1 **do**
  $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$
  $x_{t-1} \leftarrow$ sample from $\mathcal{N}(\mu + s\Sigma \nabla_{x_t} \log p_\phi(y|x_t), \Sigma)$
end for
return x_0



Dhariwal et.al, 2021, “Diffusion Models Beat GANs on Image Synthesis”, NeurIPS

Classifier-free guidance

Classifier-free guidance:

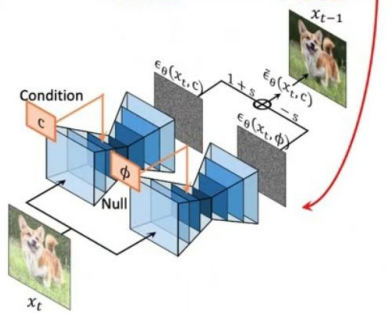
- Add conditioning into the diffusion layers and train the model with and without conditioning;
- During inference combine conditional and unconditional generation.

Algorithm 1 Joint training a diffusion model with classifier-free guidance

Require: p_{uncond}: probability of unconditional training

1: **repeat**
2: $(\mathbf{x}, \mathbf{c}) \sim p(\mathbf{x}, \mathbf{c})$ ▷ Sample data with conditioning from the dataset
3: $\mathbf{c} \leftarrow \varnothing$ with probability p_{uncond} ▷ Randomly discard conditioning to train unconditionally
4: $\lambda \sim p(\lambda)$ ▷ Sample log SNR value
5: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
6: $\mathbf{z}_\lambda = \alpha_\lambda \mathbf{x} + \sigma_\lambda \boldsymbol{\epsilon}$ ▷ Corrupt data to the sampled log SNR value
7: Take gradient step on $\nabla_\theta \left\| \boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) - \boldsymbol{\epsilon} \right\|^2$ ▷ Optimization of denoising model
8: **until** converged

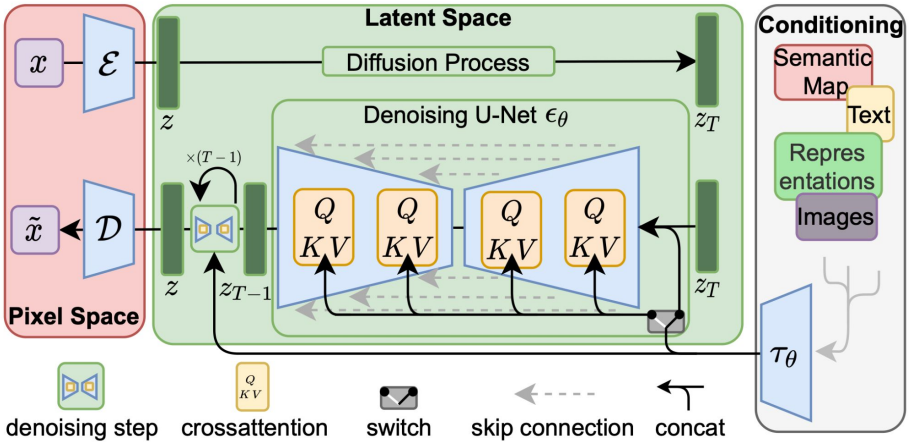
$$\tilde{\epsilon}_\theta(\mathrm{x}_t, \mathrm{y}, t) = \underbrace{\epsilon_\theta(\mathrm{x}_t, \mathrm{y}, t)}_{\text{Conditional Model}} + s\big(\underbrace{\epsilon_\theta(\mathrm{x}_t, \mathrm{y}, t)}_{\text{Conditional Model}} - \underbrace{\epsilon_\theta(\mathrm{x}_t, t)}_{\text{Unconditional Model}}\big) = \epsilon_\theta(\mathrm{x}_t, \mathrm{y}, t) + s\Delta_t$$



Ho et.al, 2021, “Classifier-Free Diffusion Guidance”, NeurIPS

Text-to-image models and latent diffusion models

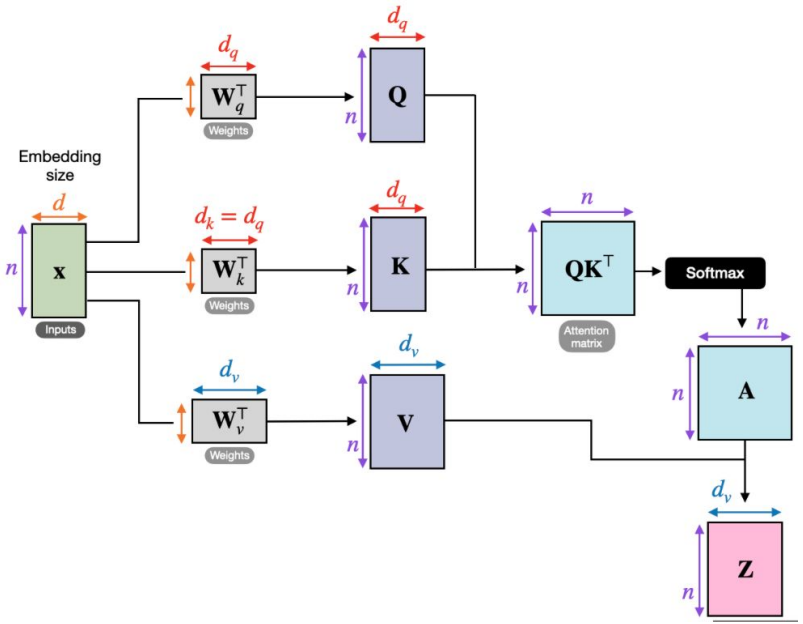
- Images can have high resolution and hence training and inference are slow.
  - Move from pixel space to latent space;
- Introduce cross-attention layers in the U-net, allowing for conditioning;
- Introduce an additional encoder for conditioning.



[1] Saharia, et.al. 2022. “Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding“, NeurIPS
[2] Rombach et.al, 2022, “High-Resolution Image Synthesis with Latent Diffusion Models”, CVPR

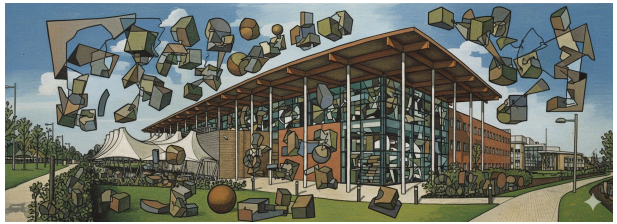
Attention mechanism and transformers

1) Have trainable parameter matrices W to get query (Q), key (K) and value (V)
2) Compute attention matrix from multiplication of Q and K and softmax (to scale to 0-1) - telling us about relationships between inputs
3) Compute the final output by multiplying attention by value

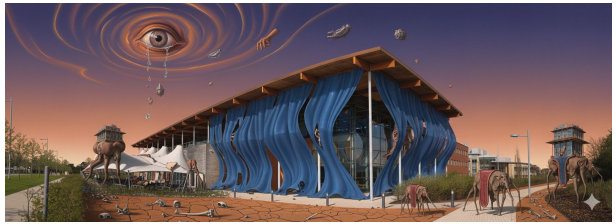


Fine-tuning

- Once we have a high-quality model we might want to extend it, for example to model well:
  - A specific person;
  - A new style;
- Re-training is expensive, we want to have a cheap way to adapt the model to our use-case;
- How can we do that?



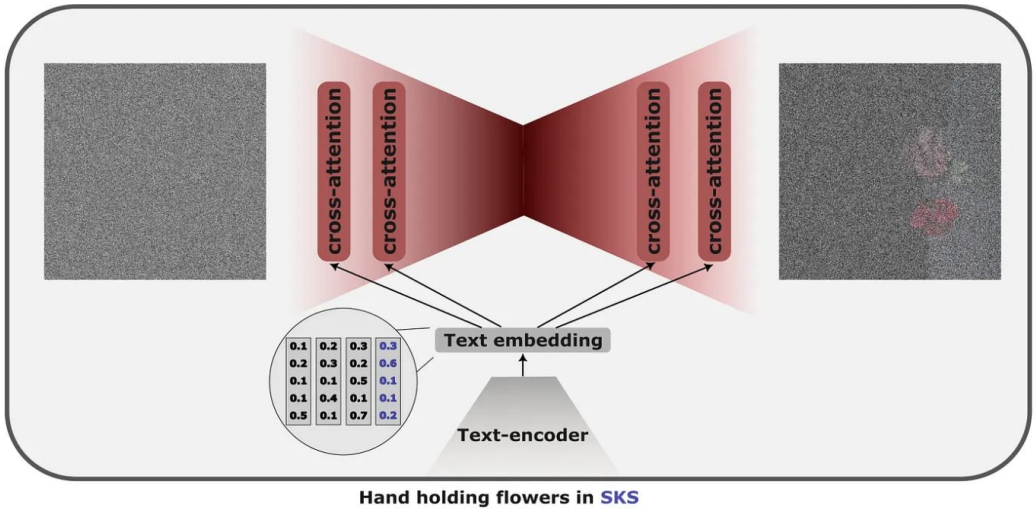
Cubism style



Surrealism style

Dreambooth

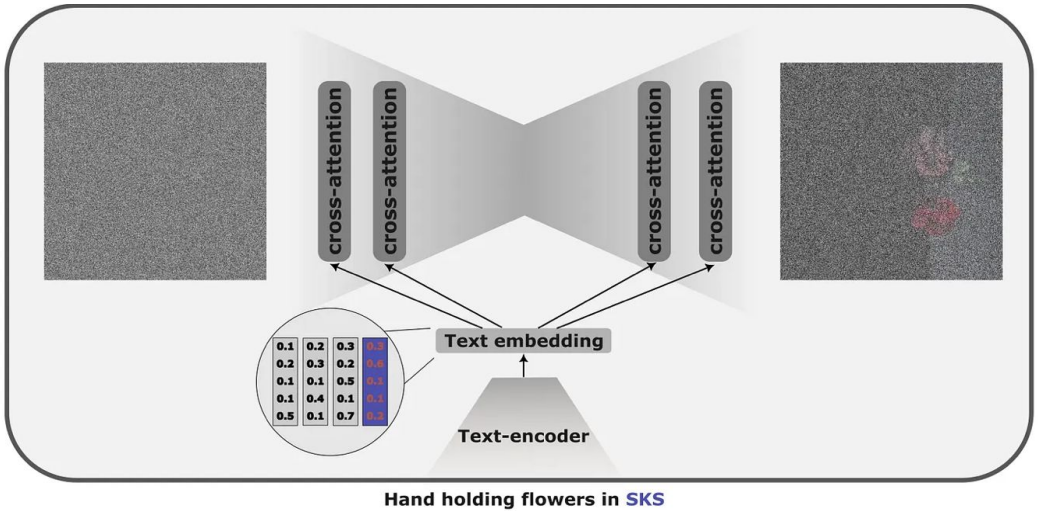
- Take a rare word; e.g. {SKS} and teach the model to map the word {SKS} to a target feature;
- That might, for example, be a style that the model has never seen;
- We would show a dozen of images in this style and fine-tune to the phrase “A painting of X in the {SKS} style”.



Ruiz et.al. 2023.“DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation”. CVPR

Textual inversion

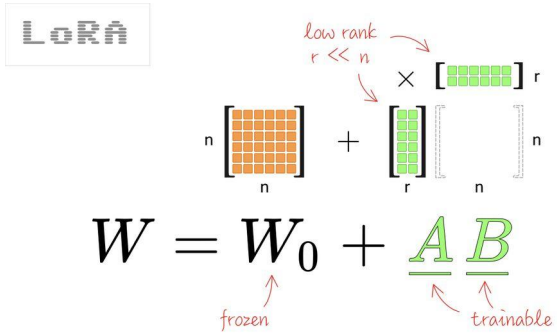
- Instead of fine-tuning the model to reproduce the desired output with rare words, we are optimizing a special textual embedding responsible for the change.
- Assumption - the knowledge stored in the latent space of diffusion models is vast. The condition we want to reproduce with the model is already known to it, but we just don’t have the token to access it.

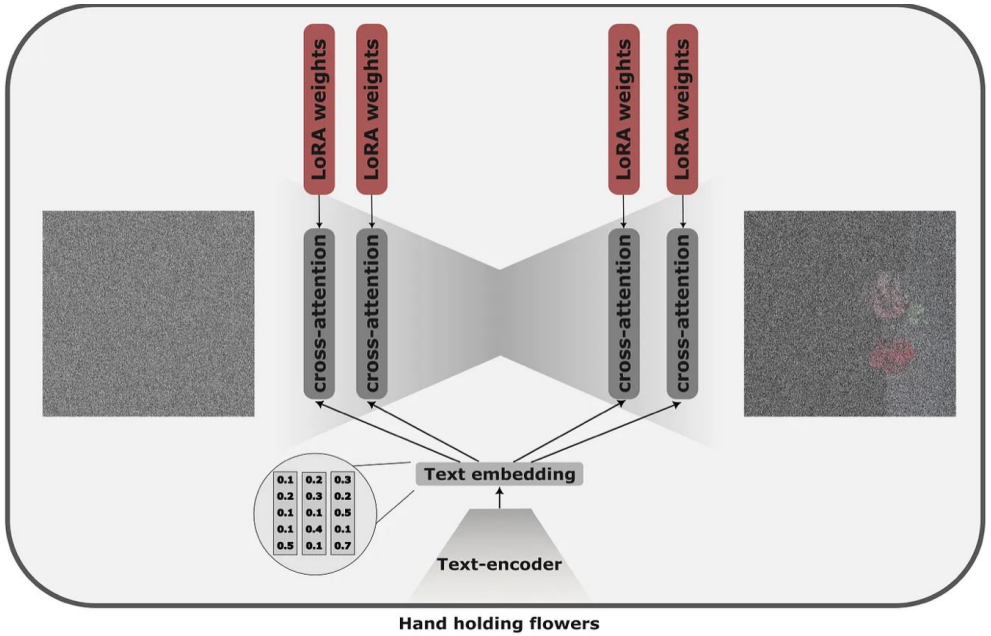


Gal et.al. 2023.“An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion”. ICLR

LoRA

- Instead of fine-tuning the whole model, which can be rather costly, we can blend a fraction of new fine-tuned weights into the original model.
- In diffusion models, LoRA is applied to cross-attention layers.

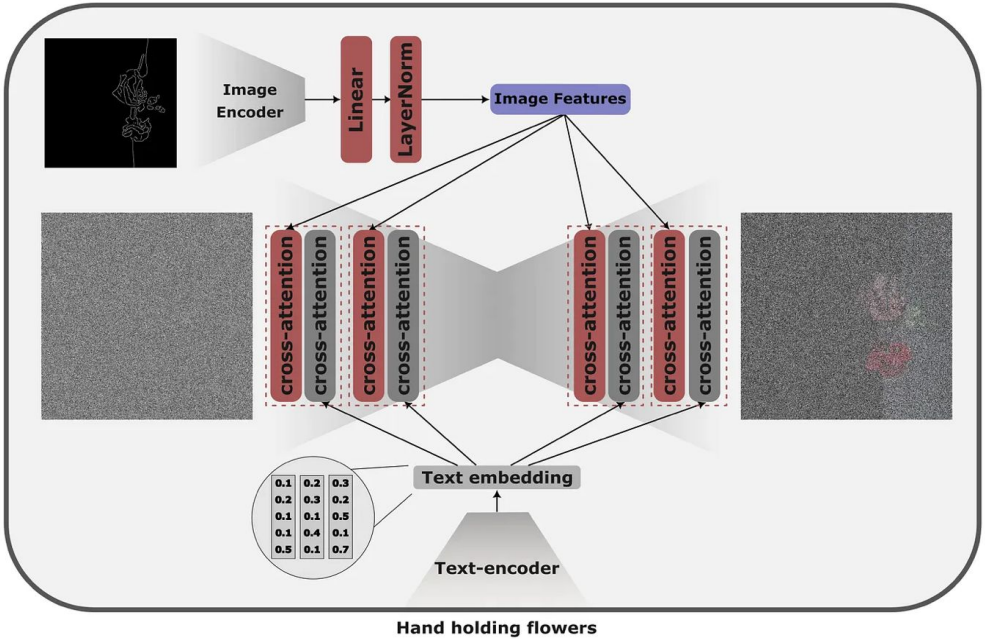




Hu et.al. 2022.“LoRA: Low-Rank Adaptation of Large Language Models”. ICLR

IP-Adaptor

- The core idea behind the IP adapter is a decoupled cross-attention mechanism that allows the combination of source images with text and generated image features.
- This is achieved by adding a separate cross-attention layer, allowing the model to learn image-specific features.



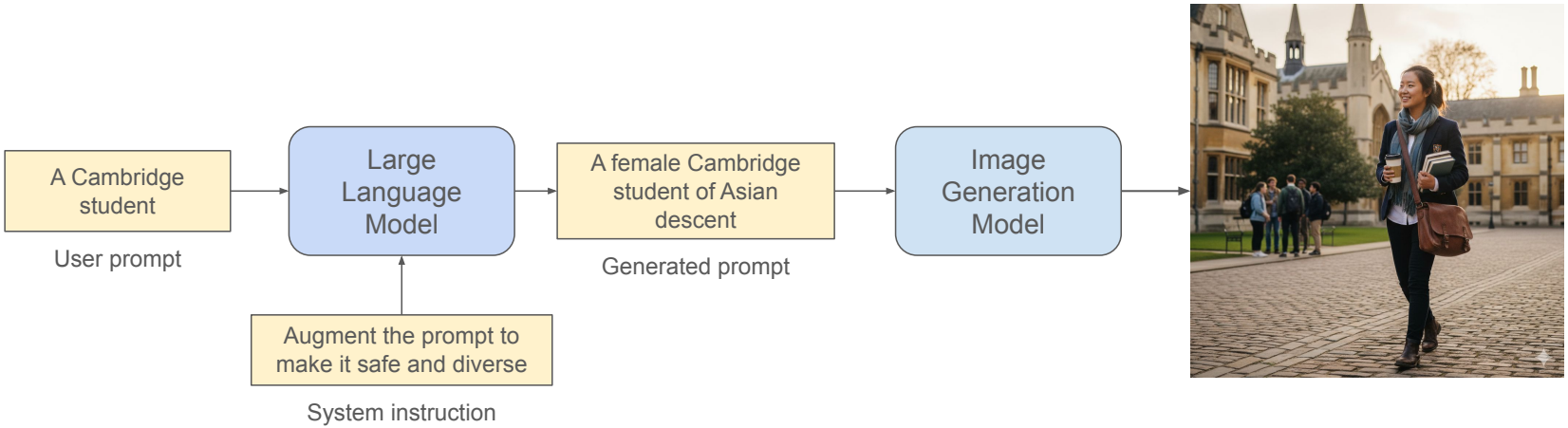
Ye et.al. 2023.“IP-Adapter: Text Compatible Image Prompt Adapter for Text-to-Image Diffusion Models”. arXiv

Generation control Image-to-Image models

- **DreamBooth:** Full fine-tuning of models for custom subjects of styles, high control level; however, it takes long time to train and are fit for one purpose only.
- **Textual Inversion:** Embedding-based learning for new concepts, low level of control, fast to train.
- **LoRA:** Lightweight fine-tuning of models for new styles/characters, medium level of control, quick to train.
- **IP-Adapter:** Soft style/content guidance via reference images, medium level of stylistic control, lightweight and efficient.

Eliminating bias in practical applications

- Biases from the data propagate to models.
  - Clean the training dataset;
  - Augment the data with diverse examples.
- Add adaptors to augment user prompts for diversity;
- Post-process the results to make them more diverse.



Evaluation (conventional)

- Frechet Inception Distance (FID) looks how well the generated images fit into the natural image manifold by comparing generated image feature statistics to the original dataset.
  - Run training and generated images through a pre-trained image classification model (inception)
  - Extract and compare feature statistics for both sets

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

- Inception score looks at whether the generated image (x) is clearly coming from the expected class (y).
  - If p(y|x) is **sharp** (i.e. concentrated on one class), the image is likely **clear and meaningful**.
  - If p(y) is **diverse** (i.e. spread across many classes), the set of images is **varied**.

$$IS(p_{gen},p_{dis}) := \exp\left(\mathbb{E}_{x\sim p_{gen}}\left[D_{KL}\left(p_{dis}(\cdot|x)\|\int p_{dis}(\cdot|x)p_{gen}(x)dx\right)\right]\right)$$

Evaluation

- Neither FID nor Inception catch generation artifacts;
- Neither FID nor Inception can tell you about biases in the generations;
- LLMs are not great with that either;
- You need manual verification and benchmark prompt / image datasets to test models;
- In practice you need multiple generations per prompt to catch issues.

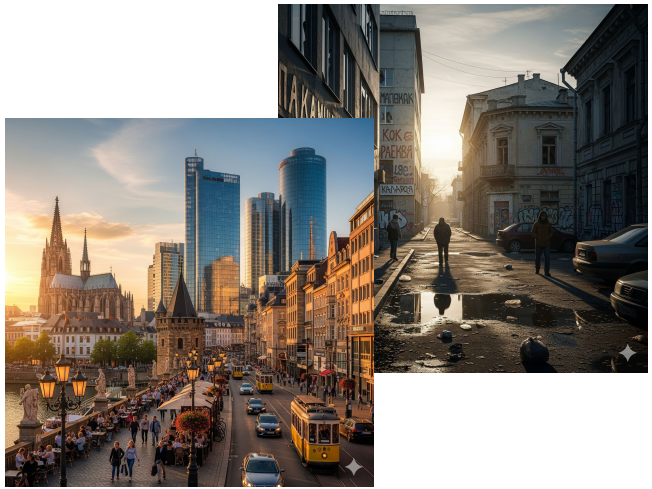


Image Generation Models Challenges

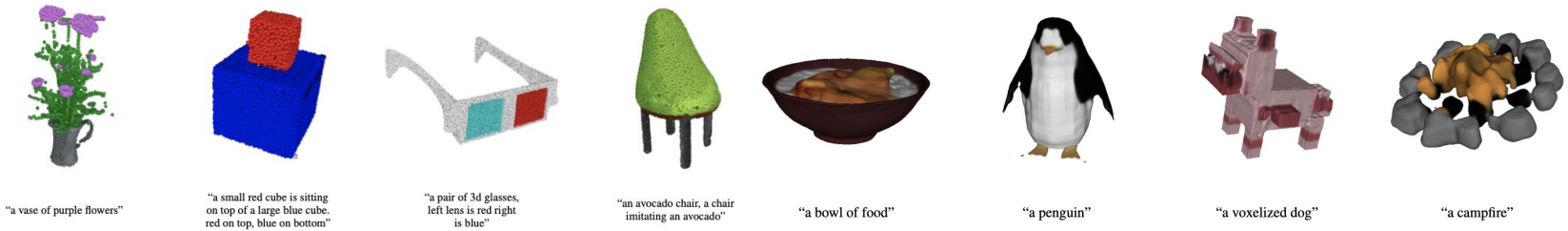
- Quality of generations at scale;
- Biases propagating to the model outputs;
- Vast compute resources required to train;
- Fairly slow inference still;
- Models are too big to run on edge devices.

Beyond image generation

3D & Graphics

Valevski et.al. 2025, “Diffusion Models Are Real-Time Game Engines”, ICLR

Poole et.al. 2022. ”DreamFusion: Text-to-3D using 2D Diffusion”. ICLR



Nichol, et.al. 2022. “Point-E: A System for Generating 3D Point Clouds from Complex Prompts”, arXiv

Jun et.al. 2023. “Shap-E: Generating Conditional 3D Implicit Functions”, arXiv

More applications

Video:

- Imagen Video (Google, 2022);
- Make-A-Video (Meta, 2022);
- Stable Video Diffusion (StabilityAI, 2023);
- MagicVideo-V2 (Bytedance, 2023);
- SoRA (OpenAI, 2024).
- SoRA 2 (OpenAI, 20245).

Diffusion for science:

- **Diffusion for molecule generation:**
  - Xu et.al., 2022. “GeoDiff: a Geometric Diffusion Model for Molecular Conformation Generation”, ICLR
  - Whatson et.al. 2023. “De novo design of protein structure and function with RFdiffusion”. Nature
- **Diffusion for weather/climate:**
  - Pathak et.al. 2022, “FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators”, arXiv

Questions?

1) Diffusion models are current industry state-of-the-art for large-scale applications;
2) Fine-tuning unlocks personalization – IP Adaptors, LoRA, etc. enable custom styles & subjects cheaply;
3) Dataset bias propagates to models - need to always beware and have mechanisms of dealing with it;
4) Don’t blindly trust the metrics - check results manually and run A/Bs with customers;
5) Don’t trust results in the papers re-run yourself to be sure the results are as reported;
6) There are open-challenges and a new fleet of models coming, so the area is still hot!

Website: https://mikhailiuk.github.io/
LinkedIn: https://www.linkedin.com/in/aliakseimikhailiuk/
Blog: https://medium.com/@mikhailiuk