Friedrich-Alexander-Universität
Technische Fakultät

Advanced Deep Learning

Diffusion Models

K. Breininger, V. Christlein

Artificial Intelligence in Medical Imaging + Pattern Recognition Lab,

Friedrich-Alexander-Universität Erlangen-Nürnberg SoSe 2023

Fake or Real?

Fake!

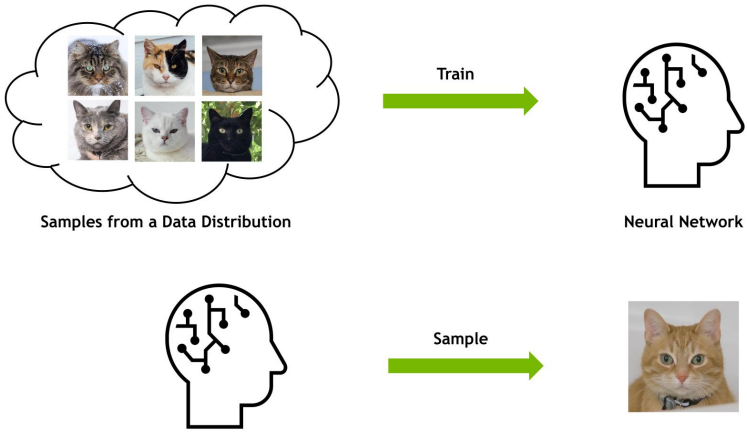
Fake!

Source: https://www.reddit.com/r/StableDiffusion/comments/z7ghbf/not_only_is_stable_diffusion_20_not_bad_but/

1. Denoising Diffusion Probabilistic Models
2. From the concept to success: Ingredients
3. Latent Diffusion Models
4. Applications
5. Beyond Image Generation

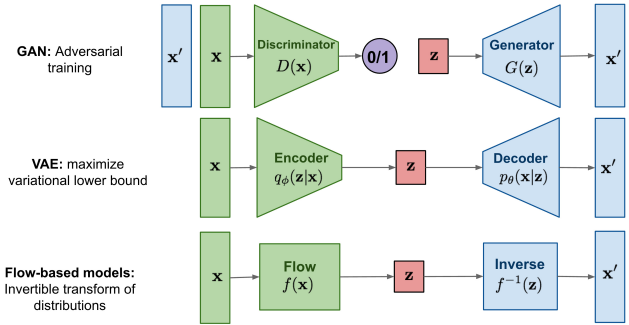
1. Denoising Diffusion Probabilistic Models
2. From the concept to success: Ingredients
3. Latent Diffusion Models
4. Applications
5. Beyond Image Generation

Deep Generative Learning



Source: https://cvpr2022-tutorial-diffusion-models.github.io/

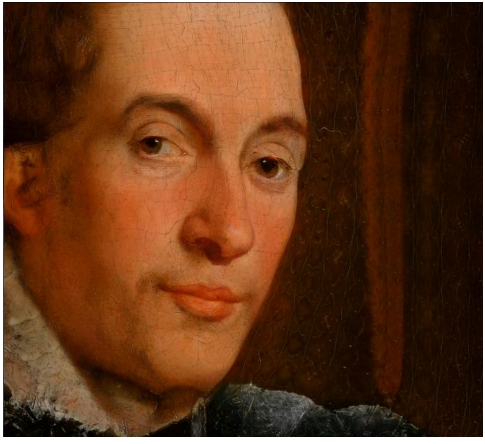
Generative Model Types



Source: https://lilianweng.github.io/lil-log/2021/07/11/diffusion-models.html

Content Generation

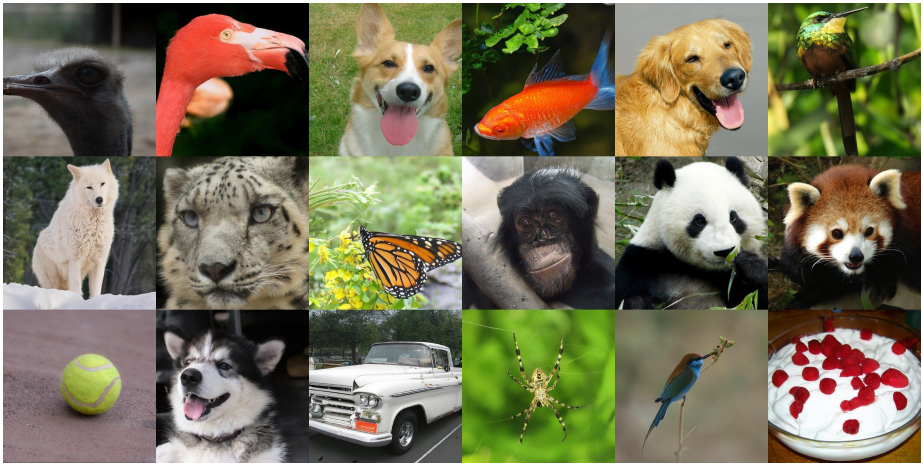
StyleGAN v3



Source: https://nvlabs.github.io/stylegan3/ [3]

Denoising Diffusion Models

Outperform GANs



Source: Dhariwal et al. 2021 [2]

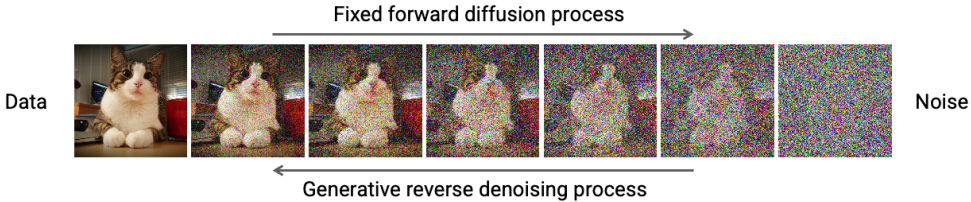
Denoising Diffusion Probabilistic Models (DDPM)

Source: https://cvpr2022-tutorial-diffusion-models.github.io/

Principle

Learning to generate by denoising

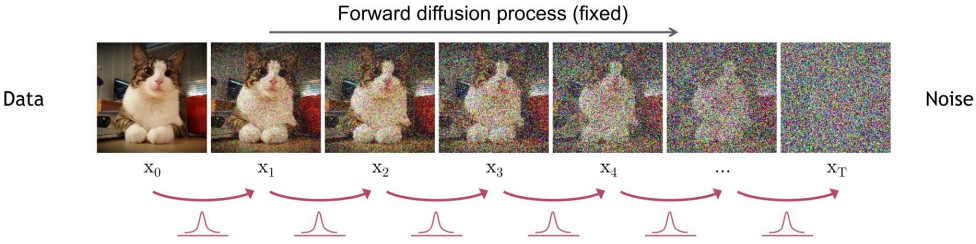
- Two steps:
  1. **Diffusion Process** gradually adds noise to image
  2. **Reverse Denoising Process** generates new data by denoising



Source: https://cvpr2022-tutorial-diffusion-models.github.io/

Forward Diffusion

Forward process in T steps: Application of noise as Markov chain

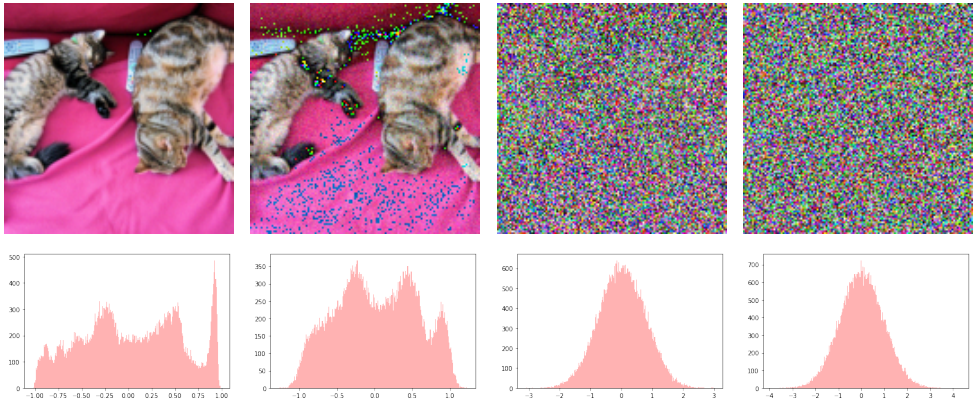


$$q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right) = \mathcal{N}\left(\mathbf{x_t}; \sqrt{\mathbf{1}-\beta_t}\mathbf{x_{t-1}}, \beta_t \mathbf{I}\right) \quad \rightarrow \quad q\left(\mathbf{x}_{1:T} \mid \mathbf{x_0}\right) = \prod_{t=1}^{T} q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)$$

β_t : variance schedule

Source: https://cvpr2022-tutorial-diffusion-models.github.io/

Noising Process



Source: https://huggingface.co/blog/annotated-diffusion

Forward Diffusion

Closed Form

x_t = $\sqrt{\bar{\alpha}_t}x_0$ + $\sqrt{1-\bar{\alpha}_t}\epsilon$

Reparametrization trick

Recall: $z \sim \mathcal{N}(\mu, \sigma^2)$ → $z = \mu + \sigma\epsilon$ with $\epsilon \sim \mathcal{N}(0, 1)$

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_{t-1} = \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1}$$

$$= \sqrt{\alpha_t}\left(\sqrt{\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_{t-1}}\epsilon_{t-2}\right) + \sqrt{1-\alpha_t}\epsilon_{t-1} = \cdots$$

$$= \sqrt{\alpha_t\alpha_{t-1}\ldots\alpha_1}x_0 + \sqrt{1-\alpha_t\alpha_{t-1}\ldots\alpha_1}\epsilon = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$

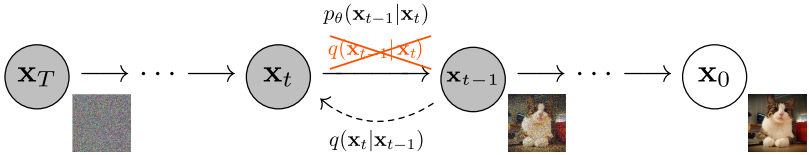
$\epsilon_0, \ldots, \epsilon_{t-2}, \epsilon_{t-1} \sim \mathcal{N}(0, I)$

$\alpha_t = 1 - \beta_t$

$\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$

Source: https://medium.com/@steinsfu/diffusion-model-clearly-explained-cd331bd41166

Reverse Process: Denoising



- Target distribution: $q\left(x_{t-1} \mid x_t\right) = \mathcal{N}\left(x_{t-1}; \tilde{\mu}_t\left(x_t, x_0\right), \tilde{\beta}_t I\right)$
- Approximated distribution:

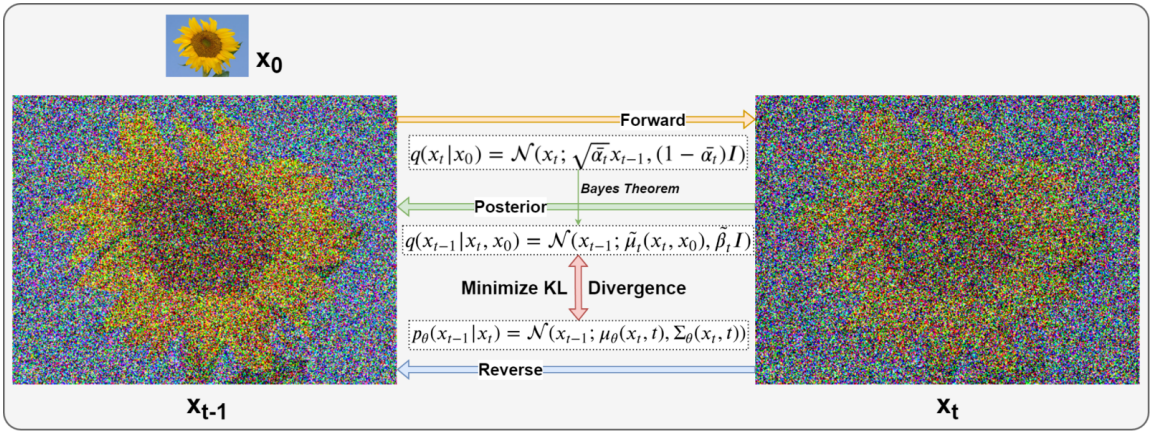
$$p\left(\mathbf{x}_T\right) = \mathcal{N}\left(\mathbf{x}_T; \mathbf{0}, \mathbf{I}\right)$$

$$p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right) = \mathcal{N}\Big(\mathbf{x}_{t-1};\ \mu_\theta\left(\mathbf{x}_t, t\right), \Sigma_\theta(x_t, t)\Big) \quad \mu_\theta\left(\mathbf{x}_t, t\right), \Sigma_\theta(x_t, t)\Big)$$

➔ Trainable Network (U-net, Denoising Autoencoder)

Source: Ho, Jain, and Abbeel 2020 [10] (adapted)

Training Overview



Source: https://learnopencv.com/denoising-diffusion-probabilistic-models/

Reverse Process: Denoising

Loss

Negative log-likelihood: $-\log(p_\theta(\mathbf{x_0}))$ → untractable → optimize variational lower bound

$$
\begin{aligned}
\mathbb{E}\left[-\log(p_\theta(\mathbf{x_0}))\right] &\leq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x_0})}\right] \\
&\leq \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \textstyle\sum_{t\geq \mathbf{1}} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] \\
&\vdots \quad [10] \\
&\leq \mathbb{E}_q\Big[\underbrace{+ D_{\mathrm{KL}}\left(q\left(\mathbf{x}_T \mid \mathbf{x_0}\right) \| p_\theta\left(\mathbf{x}_T\right)\right)}_{L_T}\Big] \quad \text{→ constant → ignore} \\
&\qquad + \textstyle\sum_{t=\mathbf{2}}^{T} \underbrace{D_{\mathrm{KL}}\left(q\left(\mathbf{x}_{t-\mathbf{1}} \mid \mathbf{x}_t, \mathbf{x_0}\right) \| p_\theta\left(\mathbf{x}_{t-\mathbf{1}} \mid \mathbf{x}_t\right)\right)}_{L_{t-1}} \\
&\qquad \underbrace{- \log p_\theta\left(\mathbf{x_0} \mid \mathbf{x_1}\right)}_{L_\mathbf{0}} \quad \text{→ reconstruction term → can be ignored}
\end{aligned}
$$

Reverse Process: Denoising

Loss

VLB Loss $\mathbb{E}[-log\ p_\theta(x_0)] \leq \mathbb{E}_q[L_T + \sum_{t>1} D_{KL}(\ q(x_{t-1}|x_t, x_0)\ \|\ p_\theta(x_{t-1}|x_t)\ ) + L_0]$

$\mathbb{E}[-log\ p_\theta(x_0)] \leq \mathbb{E}_q[L_T + \sum_{t>1} D_{KL}(\ q(x_{t-1}|x_t, x_0)\ \|\ p_\theta(x_{t-1}|x_t)\ ) + L_0]$

↓

Simple loss [10] $L_{simple} = \mathbb{E}_{t,x_0,\epsilon}\| \epsilon - \epsilon_\theta(x_t, t) \|^2$

Surrogate objective L_{Simple}

- Model ϵ_θ to predict added noise ϵ
- Estimated noise ϵ_θ allows derivation of mean μ_θ
- L_{Simple} provides no learning signal for variance Σ_θ
  - → [10]: $\Sigma_\theta = \sigma^2\mathbf{I}$ with $\sigma^2 = \beta_t$
  - → [11]: improved variation using a hybrid loss

Training Process Overview

1. Select random timestep (encode it)
2. Generate noise ϵ
3. Generate noisy image x_t given random image x_0

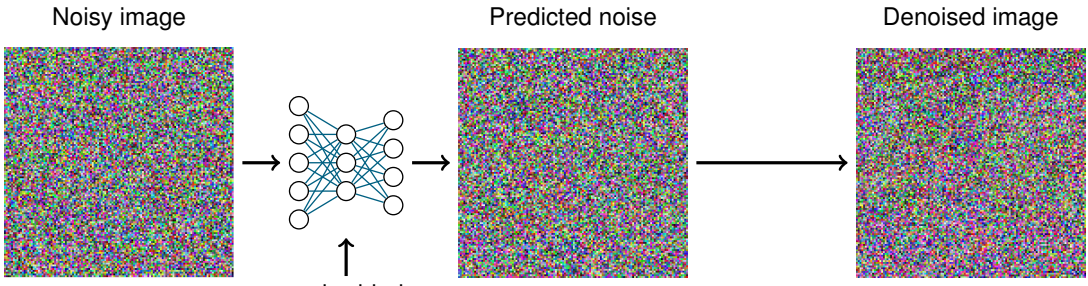
$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$

4. Given timestep & image
5. Estimate noise
6. Update model:

$\nabla_\theta \|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x_0} + \sqrt{1 - \bar{\alpha}_t}\epsilon, t)\|^2$

Reverse Diffusion

Sampling

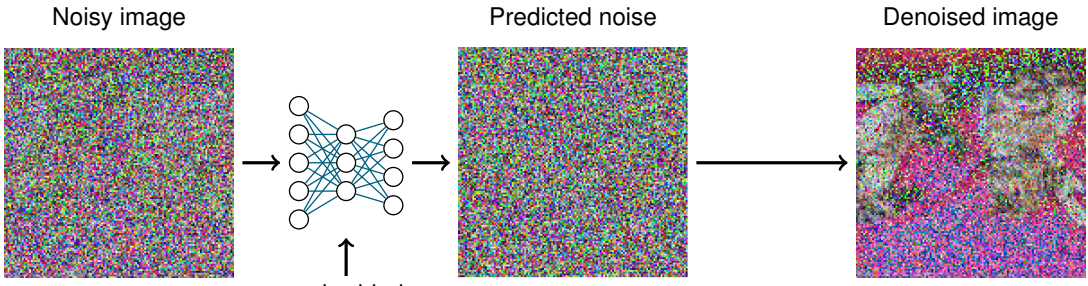


$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \sigma_t \epsilon$$

Source: https://huggingface.co/blog/annotated-diffusion

Reverse Diffusion

Sampling

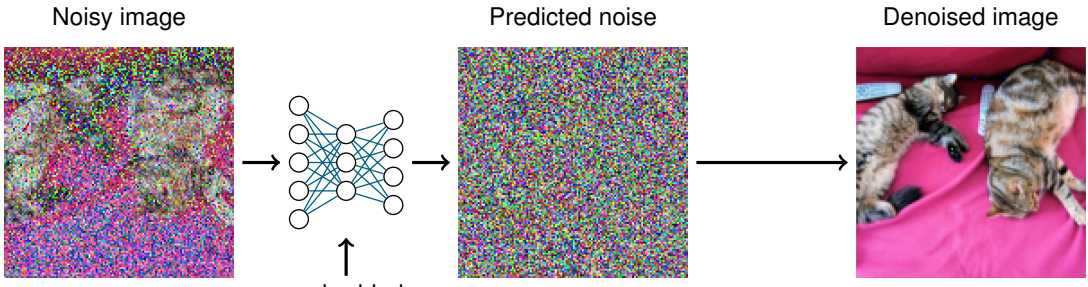


$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \sigma_t \epsilon$$

Source: https://huggingface.co/blog/annotated-diffusion

Reverse Diffusion

Sampling



$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \sigma_t \epsilon$$

Source: https://huggingface.co/blog/annotated-diffusion

Applications

Image Generation

Source: Dhariwal et al. 2021 [2]

1. Denoising Diffusion Probabilistic Models
2. From the concept to success: Ingredients
3. Latent Diffusion Models
4. Applications
5. Beyond Image Generation

Overview of ingredients

"For example, it takes around 20 hours to sample 50k images of size 32 × 32 from a DDPM, but less than a minute to do so from a GAN on an Nvidia 2080 Ti GPU." [12]

- **Beta / variance scheduling**
- Guided diffusion:
  ➔ Classifier-based and classifier-free
- Denoising Diffusion Implicit Models (DDIMs) [12] and diffusion step subsampling [11]
- Latent Diffusion Models
- ... and a lot of implementation & further tricks

Non-guided image generation. Source: [13]

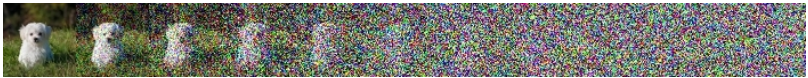
Beta Scheduling

Reminder:

$$q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right) = \mathcal{N}\left(\mathbf{x_t}; \sqrt{\mathbf{1}-\beta_t}\mathbf{x_{t-1}}, \beta_t \mathbf{I}\right)$$

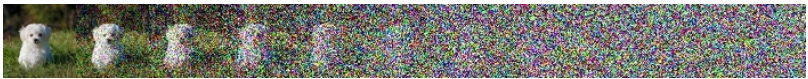
where β_t describes the **variance schedule**.

Straight-forward idea: Use linear beta schedule



Beta Scheduling (cont.)

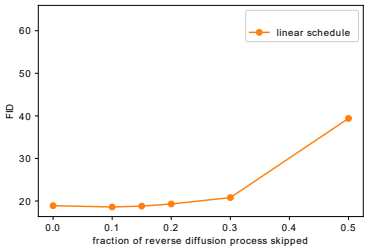
Straight-forward idea: Use linear beta schedule



Observations:

- Fast increase in noise level, little information left in later steps
- A lot of steps can be dropped without compromising quality

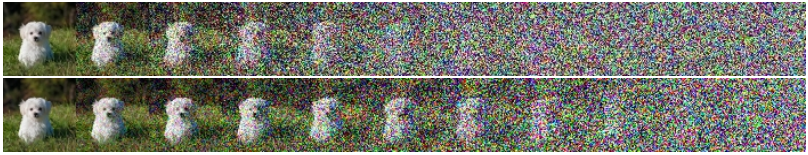
→ Nichol and Dhariwal [11]: Guesstimate a more sensible schedule



Source: Nichol and Dhariwal 2021 [11]

Beta Scheduling (cont.)

Straight-forward idea: Use linear beta schedule

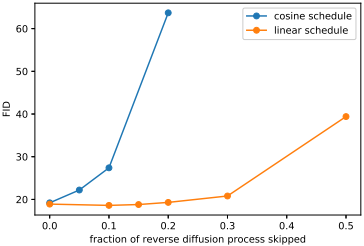


Alternative: Cosine schedule

$$\bar{\alpha}_t = \frac{f(t)}{f(\mathbf{0})}, \quad f(t) = \cos\left(\frac{t/T + s}{\mathbf{1} + s} \cdot \frac{\pi}{\mathbf{2}}\right)^{\mathbf{2}} \tag{1}$$

$$\beta_t = \mathbf{1} - \frac{\bar{\alpha}_t}{\bar{\alpha}_{t-\mathbf{1}}} \tag{2}$$

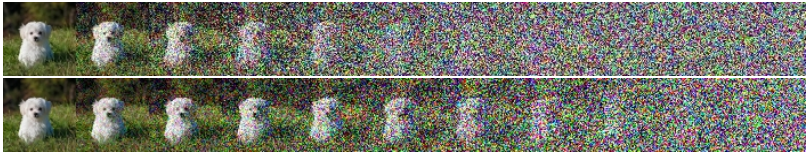
+ clipping for β_t to 0.999 to avoid singularities



Source: Nichol and Dhariwal 2021 [11]

Beta Scheduling (cont.)

Straight-forward idea: Use linear beta schedule

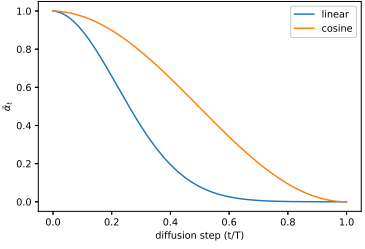


Alternative: Cosine schedule

$$\bar{\alpha}_t = \frac{f(t)}{f(\mathbf{0})}, \quad f(f) = \cos\left(\frac{t/T + s}{\mathbf{1} + s} \cdot \frac{\pi}{\mathbf{2}}\right)^{\mathbf{2}} \tag{3}$$

$$\beta_t = \mathbf{1} - \frac{\bar{\alpha}_t}{\bar{\alpha}_{t-\mathbf{1}}} \tag{4}$$

+ clipping for β_t to 0.999 to avoid singularities



Source: Nichol and Dhariwal 2021 [11]

Overview of ingredients

"For example, it takes around 20 hours to sample 50k images of size 32 × 32 from a DDPM, but less than a minute to do so from a GAN on an Nvidia 2080 Ti GPU." [12]

- Beta / variance scheduling
- **Guided diffusion:**
  → Classifier-based and classifier-free
- Denoising Diffusion Implicit Models (DDIMs) [12] and diffusion step subsampling [11]
- Latent Diffusion Models
- ... and a lot of implementation & further tricks

Non-guided image generation. Source: [13]

Guided Diffusion

- Unconditioned diffusion models generate images - with little use
  ➔ need for **guidance** / **conditioning**
- Success of current models: Efficient prompting, editing, etc.
- Different ways to insert conditioning information
  Conditional, classifier-guided and classifier-free

Non-guided image generation. Source: [13]

Guided Diffusion – Classifier Guidance [2]

Straight-forward idea: Exploit a classifier as additional component[1]:

$$p_{\theta,\phi}(x_t|x_{t+\mathbf{1}}, y) = Z \cdot \underbrace{p_\theta(x_t|x_t t + \mathbf{1})}_{\text{unconditional reverse process}} \cdot \underbrace{p_\phi(y|x_t)}_{\text{classifier for noisy images}}$$

where

- Z: normalizing constants
- y: class label
- θ / ϕ: parameters of reverse model / classifier

[1] We drop t from the conditional distributions for brevity.

Guided Diffusion – Classifier Guidance [2]

Straight-forward idea: Exploit a classifier as additional component[1]:

$$p_{\theta,\phi}(x_t|x_{t+1}, y) = Z \cdot \underbrace{p_\theta(x_t|x_t t + 1)}_{\text{unconditional reverse process}} \cdot \underbrace{p_\phi(y|x_t)}_{\text{classifier for noisy images}}$$

$p_{\theta,\phi}(x_t|x_{t+1}, y)$ is typically intractable but it can be conveniently approximated [2]:

$$p_{\theta,\phi}(x_t|x_{t+1}, y) \propto \mathcal{N}(\mu_\theta(x_t) \cdot s\,\Sigma_\theta(x_t)\nabla_{x_t}\log p_\phi(y|x_t), \Sigma_\theta(x_t))$$

➔ the"usual" transition but with a shifted mean according to the classification gradient
➔ scale parameter s to control classifier guidance strength

[1] We drop t from the conditional distributions for brevity.

Guided Diffusion – Classifier Guidance [2] (cont.)

Insights:

- Classifier scaling matters:
  s controls trade-off between **sample fidelity** vs. **diversity**
- Classifier can be trained by sampling noised images
- Combination of guidance and conditioning possible

Conditional	Guidance	Scale	FID (↓)	sFID (↓)	Precision	Recall
✗	✗		26.21	**6.35**	0.61	0.63
✗	✓	1.0	33.03	6.99	32.92	**0.65**
✗	✓	10.0	**12.00**	10.40	**0.76**	0.44
✓	✗		10.94	6.02	0.69	**0.63**
✓	✓	1.0	**4.59**	**5.25**	0.82	0.52
✓	✓	10.0	9.11	10.93	**0.88**	0.32

Table: Effect of classifier guidance on sample quality. Models trained for 2M iterations on ImageNet 256×256 with batch size 256.

$s = 1$

$s = 10$.

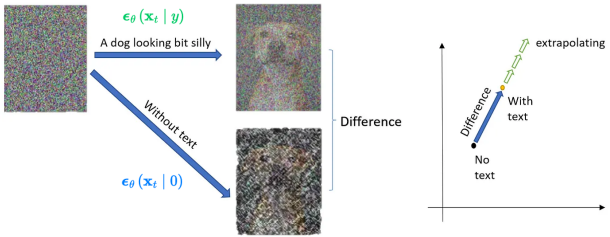
Source: Dhariwal and Nichol 2021 [2] (adapted)

Guided Diffusion - Classifier-free Guidance

- Problem with classifier guidance → additional network
- New idea - smart conditioning:

$$\hat{\epsilon}_\theta\left(\mathbf{x}_t \mid y\right) = (\mathbf{1} + s) \cdot \epsilon_\theta\left(\mathbf{x}_t \mid y\right) - s \cdot \epsilon_\theta\left(\mathbf{x}_t \mid \mathbf{0}\right)$$

- Generate output with and without text
- Enhance impact of text while generating image

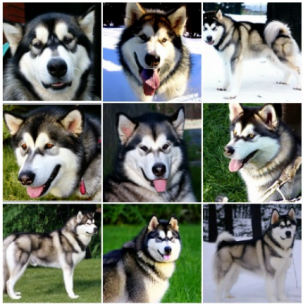


Source: https://medium.com/aiguys/googles-imagen-vs-openai-s-dalle-2-f760b60de800

Trade-off

No guidance, FID=7.27, IS=82.45

$s = \mathbf{1}$, FID=7.86, IS=297.98



$s = \mathbf{4}$, FID=21.53, IS=421.03

Source: Ho and Salimans 2021 [13]

Overview of ingredients

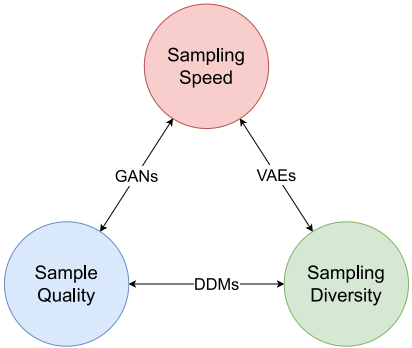
"For example, it takes around 20 hours to sample 50k images of size 32 × 32 from a DDPM, but less than a minute to do so from a GAN on an Nvidia 2080 Ti GPU." [12]

- Beta / variance scheduling
- Guided diffusion:
  ➔ Classifier-based and classifier-free
- Denoising Diffusion Implicit Models (DDIMs) [12] and diffusion step subsampling [11]
- **Latent Diffusion Models**
- ... and a lot of implementation & further tricks

Non-guided image generation. Source: [13]

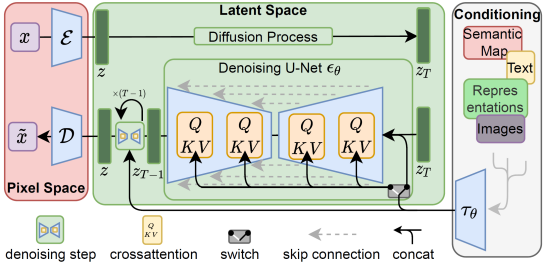
1. Denoising Diffusion Probabilistic Models
2. From the concept to success: Ingredients
3. **Latent Diffusion Models**
4. Applications
5. Beyond Image Generation

A Trilema


Sampling Speed
GANs
VAEs
Sample Quality
DDMs
Sampling Diversity

Latent Diffusion

- **Motivation:** Capacity wasted on imperceptible details
- **Core idea:** Move to latent space
- **State-of-the-art** results
- Main idea behind **Stable Diffusion**

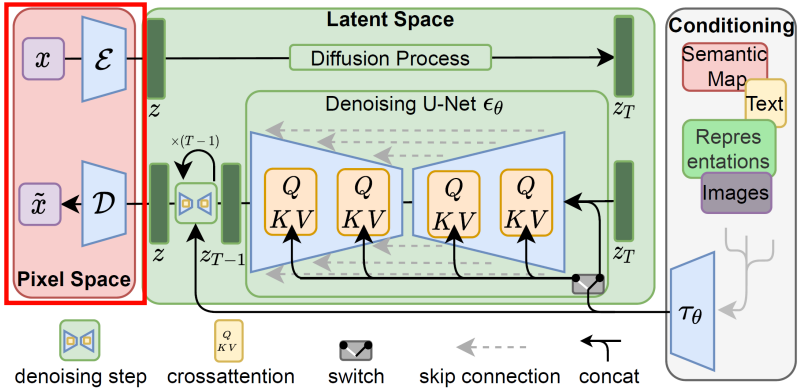


Source: Rombach, Blattmann, Lorenz, et al. 2022 [5]

Robin Rombach, Andreas Blattmann, Dominik Lorenz, et al. "High-Resolution Image Synthesis with Latent Diffusion Models". In: CVPR. June 2022, pp. 10674–10685

https://github.com/Stability-AI/stablediffusion

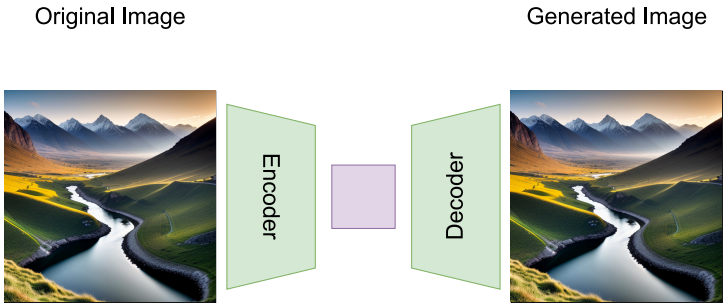
Latent Diffusion



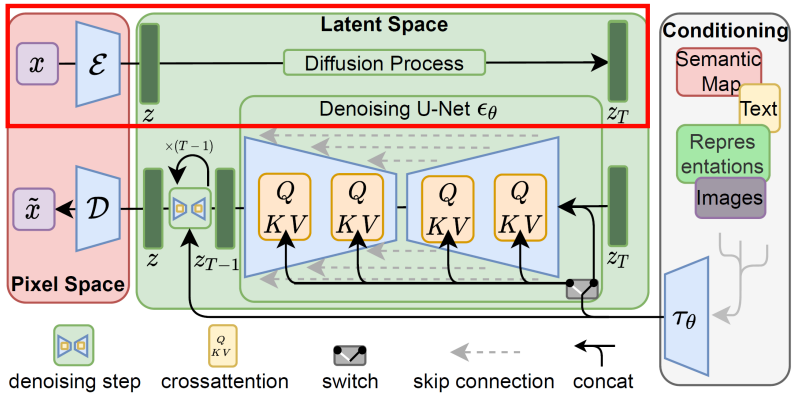
Source: Rombach, Blattmann, Lorenz, et al. 2022 [5]

Latent Diffusion

- **VAE:** Compress, Reconstruct

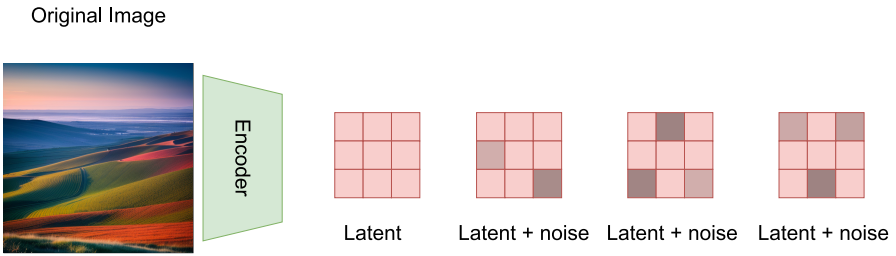


Latent Diffusion

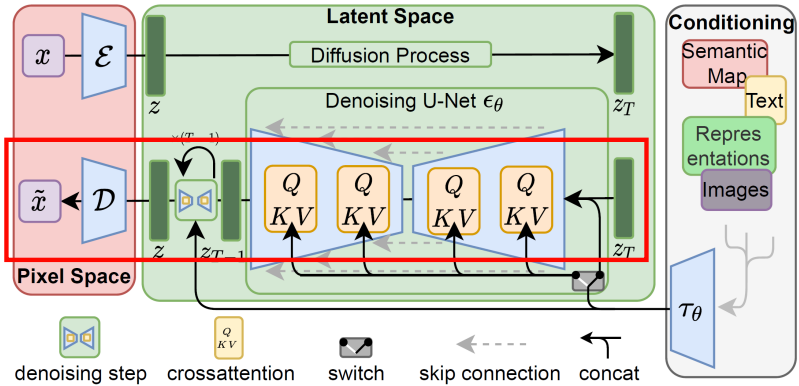


Source: Rombach, Blattmann, Lorenz, et al. 2022 [5]

Latent Diffusion

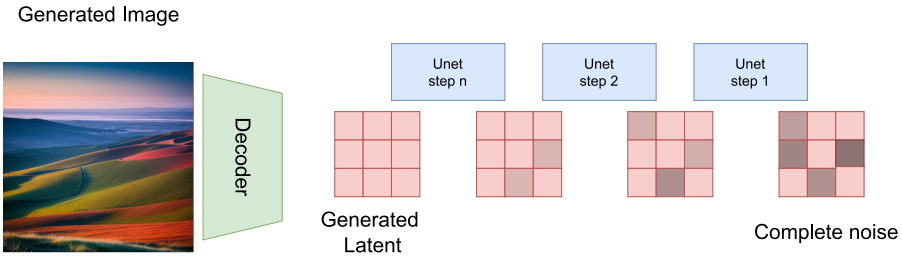


Latent Diffusion

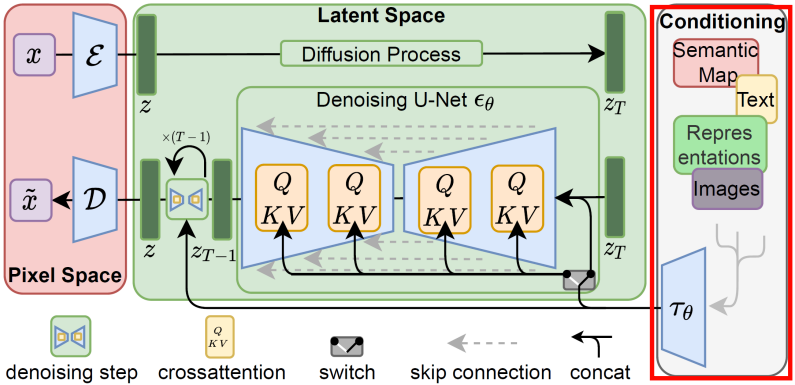


Source: Rombach, Blattmann, Lorenz, et al. 2022 [5]

Latent Diffusion



Latent Diffusion



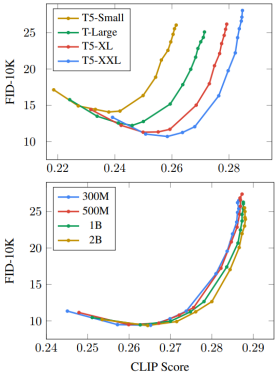
Source: Rombach, Blattmann, Lorenz, et al. 2022 [5]

Conditioning

- Transformer model used for language understanding
- Token embeddings from text prompts
- Larger language models impact image quality more than larger image generation components.
- Stable Diffusion model uses **CLIP**

Additional explanation for these plots: FID-10k represents image fidelity (↓ is better), CLIP score measures image-text alignment (↑ is better alignment). Dots represent increasing guidance values [1, 1.25, 1.5, 1.75, 2, 3, 4, 5, 6, 7, 8, 9, 10] for classifier-free guidance.

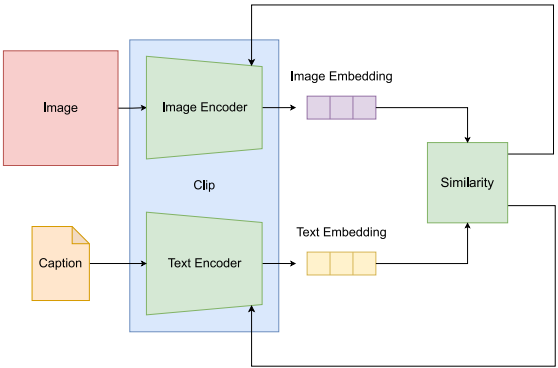
- → The larger the text-encoder, the better
- → U-Net size makes little difference
- → Guidance around **1.5** results in high fidelity & good CLIP scores, strong guidance (>2) reduces fidelity



Pareto curved with different guidance values: Impact of text-encoder size (top) and impact of U-Net size (bottom).

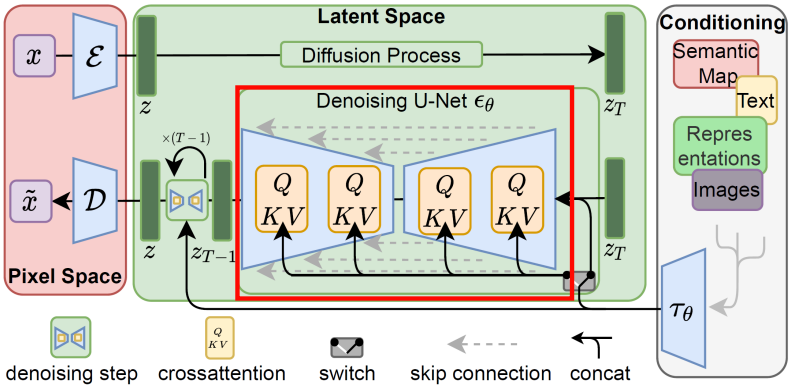
Chitwan Saharia, William Chan, Saurabh Saxena, et al. "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding". In: NeurIPS. vol. 35. 2022, pp. 36479–36494

CLIP



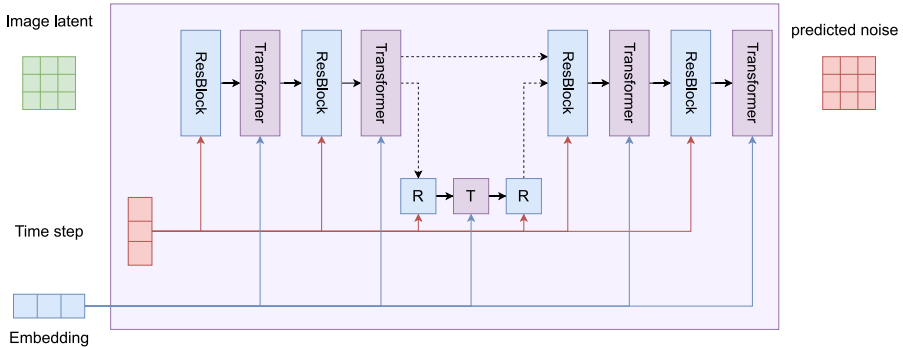
Alec Radford, Jong Wook Kim, Chris Hallacy, et al. "Learning Transferable Visual Models From Natural Language Supervision". In: Proceedings of the 38th International Conference on Machine Learning. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 8748–8763

Latent Diffusion



Source: Rombach, Blattmann, Lorenz, et al. 2022 [5]

Conditioning



Some examples

Stable Diffusion

"Realistic Homer Simpson"

Source: https://www.reddit.com/r/StableDiffusion/comments/13hlh9r/realistic_homer_simpson/

Some Examples

Midjourney

Text prompt

medium-full off-center shot,
35 mm Kodachrome film still,
capturing a Japanese woman peacing out and waving down a taxi,
wearing a gingham print dress made of silk,
blue/white palette,
accessorized by sleek pearl earrings,
another moody late-night in Tokyo
–ar 1:1

Generate →

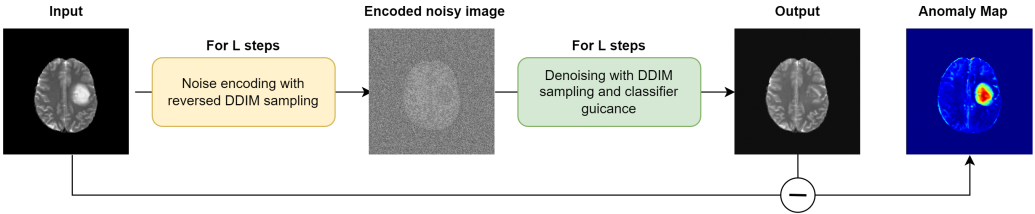
Source: https://twitter.com/nickfloats/status/1645522764084772875

1. Denoising Diffusion Probabilistic Models
2. From the concept to success: Ingredients
3. Latent Diffusion Models
4. **Applications**
5. Beyond Image Generation

Anomaly Detection

Idea: Uses diffusion model to find anomaly in the diseased images.

1. Train a DDPM and binary classifier (diseased and healthy subjects)
2. Denoising process with classifier guidance to generate a healthy image of the diseased image
3. Create anomaly map



Sampling scheme for image-to-image translation between a diseased input image and healthy output image

Source: Wolleb, Bieder, Sandkühler, et al. 2022 [4]

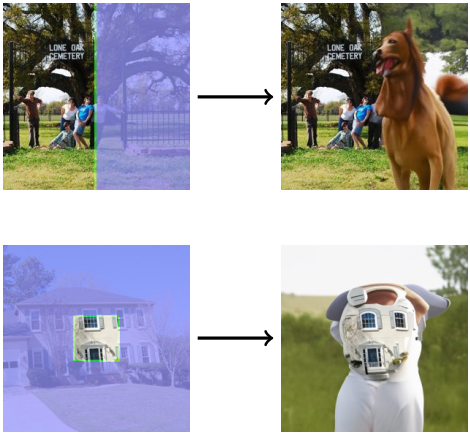
Julia Wolleb, Florentin Bieder, Robin Sandkühler, et al. "Diffusion Models for Medical Anomaly Detection". In: MICCAI 2022. 2022, pp. 35–45

Image Inpainting

Source: Lugmayr, Danelljan, Romero, et al. 2022 [1]

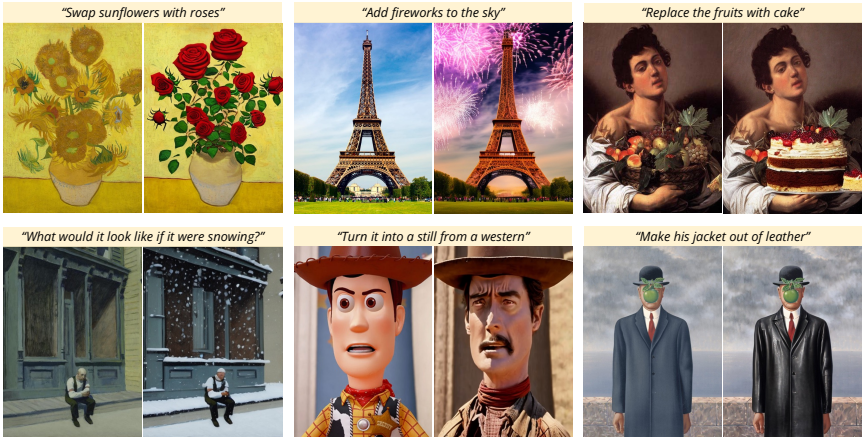
Image Inpainting

Failure Cases



Source: Lugmayr, Danelljan, Romero, et al. 2022 [1]

Image Editing

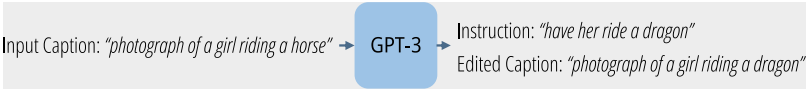


Source: Brooks, Holynski, and Efros 2023 [14]

Image Editing

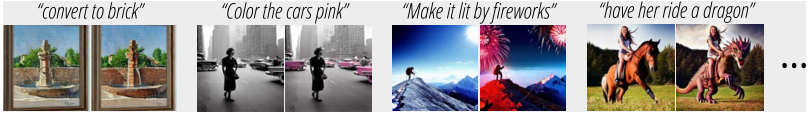
Synthetic Training Data Generation

1. Generate text edits



2. Generate paired images. Prompt2Prompt: encourage multiple generations to be similar

3. Generate 450k training images



Source: Brooks, Holynski, and Efros 2023 [14]

Image Editing

Top and Flops

"Zoom into the image"

"Move it to Mars"

"Color the tie blue"

"Have the people swap places"

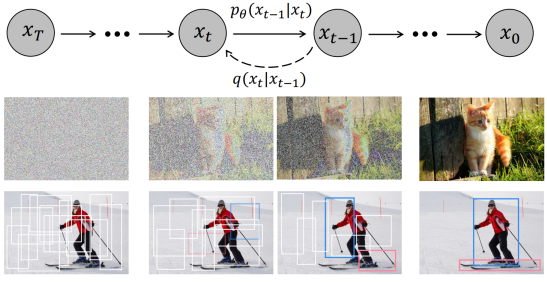
Source: Brooks, Holynski, and Efros 2023 [14]

1. Denoising Diffusion Probabilistic Models
2. From the concept to success: Ingredients
3. Latent Diffusion Models
4. Applications
5. **Beyond Image Generation**

Object Detection

DiffusionDet

- Noisy boxes are constructed with Gaussian noise during training
- Noisy boxes are randomly sampled during inference

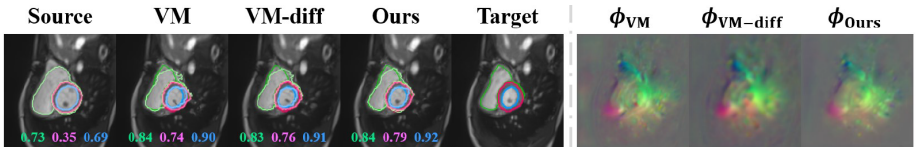


Source: Chen, Sun, Song, et al. 2022 [15]

Shoufa Chen, Peize Sun, Yibing Song, et al. DiffusionDet: Diffusion Model for Object Detection. 2022

DiffuseMorph

- Process of aligning multiple medical images, volumes or surfaces to a common coordinate system
- Deformation fields are used to represent changes between two images

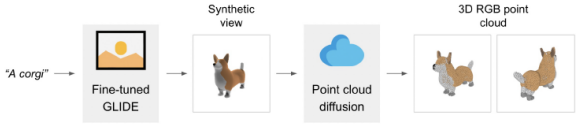


Source: Kim, Han, and Ye 2022 [16]

Boah Kim, Inhwa Han, and Jong Chul Ye. "DiffuseMorph: Unsupervised Deformable Image Registration Using Diffusion Model". In: Computer Vision – ECCV 2022. Cham: Springer Nature Switzerland, 2022, pp. 347–364

3D Object Generation

Point-E



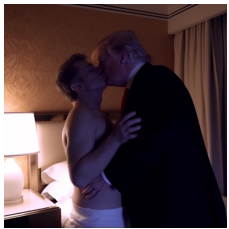
Source: Nichol, Jun, Dhariwal, et al. 2022 [17]

Alex Nichol, Heewoo Jun, Prafulla Dhariwal, et al. Point-E: A System for Generating 3D Point Clouds from Complex Prompts. 2022

Misinformation/Defamation


Obama


Mike Pence


Trump


Mitch McConnell

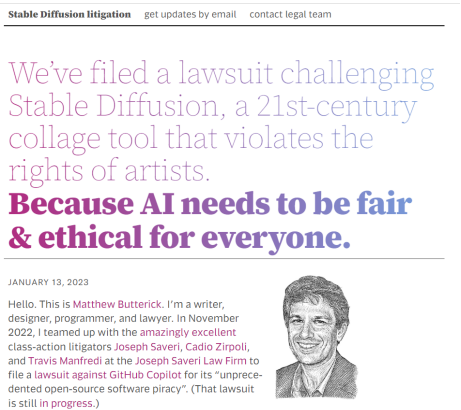

Biden

... cheating on his wife

Source: https://twitter.com/JuuustinBrown/status/1662225616635305985

Ethical Considerations

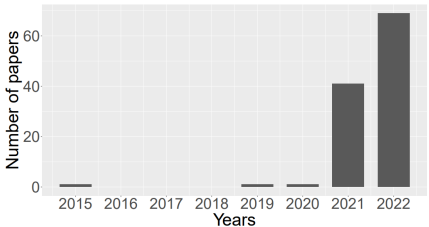
- Privacy and security: misinformation, defamation
- Creativity and jobs:
  - Automate creative work
  - → Potential negative impact on creators
- Intellectual property rights
  - Ongoing debate
  - Copyright infringement



Source: https://stablediffusionlitigation.com/

Conclusion

- Denoising Diffusion Models iterative forward/backward
- Speedup through Latent Diffusion Models
- Various applications: image generation, improvement, anomaly detection, etc.
- New research in the same direction



Source: Croitoru, Hondru, Ionescu, et al. 2023 [8]

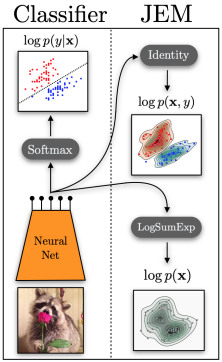
Source: Generated by Dall-E, prompt: “questions”

NEXT TIME

ADVANCED

ON DEEP LEARNING

Joint energy-based models



- Calibration
- Generation
- Out-of-distribution classification
- ...

Source: Grathwohl et al. 2020 **Grathwohl20**

Comprehensive Questions

- What impact do the different conditioning / guidance approaches have during image generation - and why?
- What do we actually learn when training diffusion models - and why?
- What is the difference between the VBL, the simple and the hybrid loss?
- Why does it make sense to learn the variance and why can this be difficult?
- How can we combine language conditioning and diffusion models?

Further Reading

- Very detailed blog: https://medium.com/@steinsfu/diffusion-model-clearly-explained-cd331bd41166
- Lilian Weng's blog: https://lilianweng.github.io/posts/2021-07-11-diffusion-models/
- Overview & implementation: https://betterprogramming.pub/diffusion-models-ddpms-ddims-and-classifier-free-guidance-e07b297b2869

References

References I

[1] Andreas Lugmayr, Martin Danelljan, Andres Romero, et al. “RePaint: Inpainting using Denoising Diffusion Probabilistic Models”. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). June 2022, pp. 11451–11461.

[2] Prafulla Dhariwal and Alexander Nichol. “Diffusion Models Beat GANs on Image Synthesis”. In: Advances in Neural Information Processing Systems. Vol. 34. Curran Associates, Inc., 2021, pp. 8780–8794.

[3] Tero Karras, Miika Aittala, Samuli Laine, et al. “Alias-Free Generative Adversarial Networks”. In: Advances in Neural Information Processing Systems. Vol. 34. Curran Associates, Inc., 2021, pp. 852–863.

[4] Julia Wolleb, Florentin Bieder, Robin Sandkühler, et al. “Diffusion Models for Medical Anomaly Detection”. In: MICCAI 2022. 2022, pp. 35–45.

References II

[5] Robin Rombach, Andreas Blattmann, Dominik Lorenz, et al. “High-Resolution Image Synthesis with Latent Diffusion Models”. In: CVPR. June 2022, pp. 10674–10685.

[6] Chitwan Saharia, William Chan, Saurabh Saxena, et al. “Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding”. In: NeurIPS. Vol. 35. 2022, pp. 36479–36494.

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, et al. “Learning Transferable Visual Models From Natural Language Supervision”. In: Proceedings of the 38th International Conference on Machine Learning. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 8748–8763.

[8] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, et al. “Diffusion Models in Vision: A Survey”. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (2023), pp. 1–20.

[9] Alex Kendall and Yarin Gal. “What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?” In: Advances in Neural Information Processing Systems. Vol. 30. Curran Associates, Inc., 2017.

References III

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising Diffusion Probabilistic Models”. In: Advances in Neural Information Processing Systems. Vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851.

[11] Alexander Quinn Nichol and Prafulla Dhariwal. “Improved Denoising Diffusion Probabilistic Models”. In: Proceedings of the 38th International Conference on Machine Learning. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 8162–8171.

[12] Jiaming Song, Chenlin Meng, and Stefano Ermon. “Denoising Diffusion Implicit Models”. In: International Conference on Learning Representations. 2021.

[13] Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance. 2021.

[14] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. “InstructPix2Pix: Learning To Follow Image Editing Instructions”. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). June 2023, pp. 18392–18402.

References IV

[15] Shoufa Chen, Peize Sun, Yibing Song, et al. DiffusionDet: Diffusion Model for Object Detection. 2022.

[16] Boah Kim, Inhwa Han, and Jong Chul Ye. “DiffuseMorph: Unsupervised Deformable Image Registration Using Diffusion Model”. In: Computer Vision – ECCV 2022. Cham: Springer Nature Switzerland, 2022, pp. 347–364.

[17] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, et al. Point-E: A System for Generating 3D Point Clouds from Complex Prompts. 2022.

[18] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, et al. “Improved Techniques for Training GANs”. In: Advances in Neural Information Processing Systems. Vol. 29. Curran Associates, Inc., 2016.