Me, teaching Diffusion Models that I knew nothing about two weeks ago!
Recap
Forward Diffusion Process

\[ q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t} x_{t-1}, \beta_t I) \quad \Rightarrow \quad \text{Sample:} \quad x_t = \sqrt{1 - \beta_t} x_{t-1} + \sqrt{\beta_t} \epsilon_{t-1} \]

where, \( \epsilon_{t-1} \sim \mathcal{N}(0, I) \)

Define, \( \bar{\alpha}_t = \prod_{s=1}^{t}(1 - \beta_s) \quad \Rightarrow \quad q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t)I) \) (Diffusion Kernel)

For sampling: \( x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{(1 - \bar{\alpha}_t)} \epsilon \quad \text{where} \quad \epsilon \sim \mathcal{N}(0, I) \)
Reverse Diffusion Process

\[ L_{\text{simple}} = \mathbb{E}_{x_0 \sim q(x_0), \epsilon \sim \mathcal{N}(0, I), t \sim \mathcal{U}(1, T)} \left[ \| \epsilon - \epsilon_\theta(\sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t} \epsilon, t) \|_2^2 \right] \]

Diffusion models often use U-Net architectures with ResNet blocks and self-attention layers to represent \( \epsilon_\theta(x_t, t) \)

---

Algorithm 2 Sampling

1: \( x_T \sim \mathcal{N}(0, I) \)
2: \( \text{for } t = T, \ldots, 1 \text{ do} \)
3: \( z \sim \mathcal{N}(0, I) \)
4: \( x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( x_t - \frac{1 - \alpha_t}{\sqrt{1 - \alpha_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z \)
5: \( \text{end for} \)
6: \( \text{return } x_0 \)
The Generative Reverse Stochastic Differential Equation

Forward Diffusion SDE:
\[ dx_t = -\frac{1}{2} \beta(t)x_t \, dt + \sqrt{\beta(t)} \, d\omega_t \]

Reverse Generative Diffusion SDE:
\[ dx_t = \left[ -\frac{1}{2} \beta(t)x_t - \beta(t) \nabla x_t \log q_t(x_t) \right] \, dt + \sqrt{\beta(t)} \, d\tilde{\omega}_t \]

How do we obtain the “Score Function”? Simulate reverse! m noise!

Song et al., ICLR, 2021
Anderson, in Stochastic Processes and their Applications, 1982
Outline for today’s class

• Theory:
  • How do we condition diffusion model?
  • How do people sample in practice? – DDIM sampling
  • How do we generate high-resolution images?

• Application:
  • Text to Image Generation
  • Image to Image Translation
  • Video Generation
Outline for today’s class

• Theory:
  • How do we condition diffusion model?
    • How do people sample in practice? – DDIM sampling
    • How do we generate high-resolution images?

• Application:
  • Text to Image Generation
  • Image to Image Translation
  • Video Generation
Conditional diffusion models

Include condition as input to reverse process

Reverse process: 
\[ p_\theta(x_{0:T}|c) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t, c), \quad p_\theta(x_{t-1}|x_t, c) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t, c), \Sigma_\theta(x_t, t, c)) \]

Variational upper bound:
\[ L_\theta(x_0|c) = \mathbb{E}_q \left[ L_T(x_0) + \sum_{t>1} D_{KL}(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t, c)) - \log p_\theta(x_0|x_1, c) \right]. \]

Incorporate conditions into U-Net

- **Scalar conditioning:** encode scalar as a vector embedding, simple spatial addition or adaptive group normalization layers.

- **Image conditioning:** channel-wise concatenation of the conditional image.

- **Text conditioning:** single vector embedding - spatial addition or adaptive group norm / a seq of vector embeddings - cross-attention.
Classifier guidance
Using the gradient of a trained classifier as guidance

Recap: What is a score function?

Forward Diffusion SDE:
\[ dx_t = -\frac{1}{2} \beta(t) x_t \, dt + \sqrt{\beta(t)} \, d\omega_t \]

Reverse Generative Diffusion SDE:
\[ dx_t = \left[ -\frac{1}{2} \beta(t) x_t - \beta(t) \nabla x_t \log q_t(x_t) \right] \, dt + \sqrt{\beta(t)} \, d\bar{\omega}_t \]

“Score Function”

\[ \min_{\theta} \mathbb{E}_{t \sim \mathcal{U}(0,T)} \mathbb{E}_{x_t \sim q_t(x_t)} \left\| S_\theta(x_t, t) - \nabla x_t \log q_t(x_t) \right\|_2^2 \]

- diffusion time \( t \)
- diffused data \( x_t \)
- neural network
- score of diffused data (marginal)
Classifier guidance

Using the gradient of a trained classifier as guidance

Applying Bayes rule to obtain conditional score function $\nabla_x \log q_t(x_t | y)$

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

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

$$\implies \nabla_x \log p(x | y) = \nabla_x \log p(y | x) + \nabla_x \log p(x),$$

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

Guidance scale: value $>1$ amplifies the influence of classifier signal.

$$p_\gamma(x | y) \propto p(x) \cdot p(y | x)^\gamma.$$
Classifier guidance

Using the gradient of a trained classifier as guidance

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

Samples from an unconditional diffusion model with classifier guidance, for guidance scales 1.0 (left) and 10.0 (right), taken from Dhariwal & Nichol (2021).
Classifier guidance
Using the gradient of a trained classifier as guidance

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 \(s\).

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

- Train unconditional Diffusion model
- Take your favorite classifier, depending on the conditioning type
- During inference / sampling mix the gradients of the classifier with the predicted score function of the unconditional diffusion model.
Classifier guidance
Problems of classifier guidance

\[ \nabla_x \log p_{\gamma}(x \mid y) = \nabla_x \log p(x) + \gamma \nabla_x \log p(y \mid x). \]

Guidance scale: value $> 1$ amplifies the influence of classifier signal.

- At each step of denoising the input to the classifier is a noisy image $x_t$. Classifier is never trained on noisy image. So one needs to re-train classifier on noisy images! Can’t use existing pre-trained classifiers.

- Most of the information in the input $x$ is not relevant to predicting $y$, and as a result, taking the gradient of the classifier w.r.t. its input can yield arbitrary (and even adversarial) directions in input space.
Classifier-free guidance

Get guidance by Bayes’ rule on conditional diffusion models

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

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

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

We proved this in classifier guidance.

\[ \nabla_x \log p_{\gamma}(x \mid y) = \nabla_x \log p(x) + \gamma \nabla_x \log p(y \mid x). \]

\[ \nabla_x \log p_{\gamma}(x \mid y) = \nabla_x \log p(x) + \gamma (\nabla_x \log p(x \mid y) - \nabla_x \log p(x)), \]

\[ \nabla_x \log p_{\gamma}(x \mid y) = (1 - \gamma) \nabla_x \log p(x) + \gamma \nabla_x \log p(x \mid y). \]

Score function for unconditional diffusion model

Score function for conditional diffusion model
Classifier-free guidance

Get guidance by Bayes’ rule on conditional diffusion models

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

This is a barycentric combination of the conditional and the unconditional score function. For \( \gamma = 0 \), we recover the unconditional model, and for \( \gamma = 1 \) we get the standard conditional model. But \( \gamma > 1 \) is where the magic happens. Below are some examples from OpenAI’s GLIDE model\(^8\), obtained using classifier-free guidance.

In practice

\[ \hat{\epsilon} = (1 + \omega)\epsilon_\theta(x_t, y) - \omega \epsilon_\theta(x_t) \]

Two sets of samples from OpenAI’s GLIDE model, for the prompt ‘A stained glass window of a panda eating bamboo.’, taken from their paper. Guidance scale 1 (no guidance) on the left, guidance scale 3 on the right.
Classifier-free guidance

Get guidance by Bayes’ rule on conditional diffusion models

\[
\hat{\epsilon} = (1 + \omega) \epsilon_{\theta}(x_t, y) - \omega \epsilon_{\theta}(x_t)
\]

\[
\nabla_x \log p_{\gamma}(x \mid y) = (1 - \gamma) \nabla_x \log p(x) + \gamma \nabla_x \log p(x \mid y).
\]

In practice:

- Train a conditional diffusion model \(p(x \mid y)\), with *conditioning dropout*: some percentage of the time, the conditioning information \(y\) is removed (10-20% tends to work well).

- The conditioning is often replaced with a special input value representing the absence of conditioning information.

- The resulting model is now able to function both as a conditional model \(p(x \mid y)\), and as an unconditional model \(p(x)\), depending on whether the conditioning signal is provided.

- During inference / sampling simply mix the score function of conditional and unconditional diffusion model based on guidance scale.
Classifier-free guidance

Trade-off for sample quality and sample diversity

Large guidance weight ($\omega$) usually leads to better individual sample quality but less sample diversity.

Classifier guidance

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

Guidance scale

Classifier

X Need to train a separate "noise-robust" classifier + unconditional diffusion model.

X Gradient of the classifier w.r.t. input yields arbitrary values.

Classifier-free guidance

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

Score function for unconditional diffusion model

Score function for conditional diffusion model

+ Train conditional & unconditional diffusion model jointly via drop-out.

+ All pixels in input receive equally ‘good’ gradients.

Rather than constructing a generative model from classifier, we construct a classifier from a generative model!

Most recent papers use classifier-free guidance! Very simple yet very powerful idea!
Outline for today’s class

• Theory:
  • How do we condition diffusion model?
  • How do people sample in practice? – DDIM sampling
  • How do we generate high-resolution images?

• Application:
  • Text to Image Generation
  • Image to Image Translation
  • Video Generation
How to accelerate sampling process?

• During test time, one needs to run the diffusion model > 1000 steps, which is slow!
• If we naively sample less steps, the quality is worse.
• Can we somehow sample less steps and have the quality?
• DDIM sampling shows how this can be done! (Most recent papers use this trick)
• Note that: During training, we only train for t-1 to t and randomly sample t between [0,T]. We do no need to run the whole generation process
DDIM sampling overview

Key Idea:
• A regular diffusion model is Markovian process -> Generation at time t-1 only depends on time t, and independent of all other time stamps.
• Diffusion Markov process has optimal solution when we minimize this loss function:
  \[ L_{\text{simple}} (\theta) := \mathbb{E}_{t,x_0,\epsilon} \left[ \| \epsilon - \epsilon_\theta (\sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon, t) \|^2 \right] \]
• We want to use the trained Markovian Diffusion process at test time, but for faster sampling we define a non-Markovian process whose optimal solution is obtained by minimizing the same loss function as Markovian one!
DDIM sampling overview

Markovian Diffusion (often called DDPM sampling – Denoising Diffusion Probabilistic Model):
• Given noisy image $X_t$ -> predict noise map $\epsilon_\theta(X_t, t)$ -> subtract noise map from $X_t$ (with some weighting) to obtain $X_{t-1}$

$$z \sim \mathcal{N}(0, I)$$
$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( x_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z$$

Non-Markovian Diffusion (DDIM sampling):
• Given noisy image $X_t$ -> predict noise map $\epsilon_\theta(X_t, t)$ -> generate clean image $X'_0$ -> add the predicted noise map to $X'_0$ to obtain $X_{t-1}$.

$$x_{t-1} = \sqrt{\alpha_{t-1}} \left( x_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}} \epsilon^{(i)}_\theta(x_t) \right) + \sqrt{1-\alpha_{t-1} - \sigma_t^2} \epsilon^{(i)}_\theta(x_t) + \sigma_t \epsilon_t$$

- Different choice of $\sigma$ results in different generative process without re-training the model
- When $\sigma = 0$ for all $t$, we have a deterministic generative process, with randomness from only $t=T$ (the last step).
Denoising Diffusion Implicit Models (DDIM)

Why can we do this?

\[
\begin{align*}
q(x_t \mid x_{t-1}) &= N(x_t; \sqrt{1 - \beta_t} x_{t-1}, \beta_t I) \\
p_\theta(x_{t-1} \mid x_t) &= N(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I)
\end{align*}
\]

Define a family of forward processes that meets the above requirement:

\[
q(x_{t-1} \mid x_t, x_0) = N \left( \sqrt{\alpha_{t-1}} x_0 + \sqrt{1 - \alpha_{t-1} - \tilde{\sigma}_t^2} \cdot \frac{x_t - \sqrt{\alpha_t} x_0}{\sqrt{1 - \alpha_t}}, \tilde{\sigma}_t^2 I \right)
\]

The corresponding reverse process is

\[
p(x_{t-1} \mid x_t) = N \left( \sqrt{\alpha_{t-1}} \tilde{x}_0 + \sqrt{1 - \alpha_{t-1} - \tilde{\sigma}_t^2} \cdot \frac{x_t - \sqrt{\alpha_t} \tilde{x}_0}{\sqrt{1 - \alpha_t}}, \tilde{\sigma}_t^2 I \right)
\]

\[
:= (x_t - \sqrt{1 - \alpha_t} \cdot c_\theta^{(t)}(x_t))/\sqrt{\alpha_t}.
\]

This has same loss function as the regular/Markovian diffusion model. Thus we can use this strategy without re-training.
Outline for today’s class

• Theory:
  • How do we condition diffusion model?
  • How do people sample in practice? – DDIM sampling
  • How do we generate high-resolution images?

• Application:
  • Text to Image Generation
  • Image to Image Translation
  • Video Generation
Cascaded generation

Class Conditioned Diffusion Model

Class ID = 213
“Irish Setter”

Model 1

Model 2

Model 3

32×32

64×64

256×256

Similar cascaded / multi-resolution image generation also exist in GAN (Big-GAN & StyleGAN)

Cascaded Diffusion Models outperform Big-GAN in FID and IS and VQ-VAE2 in Classification Accuracy Score.

Noise conditioning augmentation

Reduce compounding error

Problem:

• During training super-resolution models are trained on original low-res images from the dataset.

• During inference, these low-res images are generated by class conditioned diffusion model, which has artifacts and poor quality than original low-res images used for training.

Solution: Noise conditioning augmentation.

• During training, add varying amounts of Gaussian noise (or blurring by Gaussian kernel) to the low-res images.

• During inference, sweep over the optimal amount of noise added to the low-res images.

• BSR-degradation process: applies JPEG compressions noise, camera sensor noise, different image interpolations for downsampling, Gaussian blur kernels and Gaussian noise in a random order to an image.

Outline for today’s class

• Theory:
  • How do we condition diffusion model?
  • How do people sample in practice? – DDIM sampling
  • How do we generate high-resolution images?

• Application:
  • Text to Image Generation
  • Image to Image Translation
  • Video Generation
GLIDE
OpenAI

- A 64x64 base model + a 64x64 → 256x256 super-resolution model.
- Tried classifier-free and CLIP guidance. Classifier-free guidance works better than CLIP guidance.

Samples generated with classifier-free guidance (256x256)

CLIP guidance

What is a CLIP model?

- Trained by contrastive cross-entropy loss:

\[-\log \frac{\exp(f(x_i) \cdot g(c_j)/\tau)}{\sum_k \exp(f(x_i) \cdot g(c_k)/\tau)} - \log \frac{\exp(f(x_i) \cdot g(c_j)/\tau)}{\sum_k \exp(f(x_k) \cdot g(c_j)/\tau)}\]

- The optimal value of $f(x) \cdot g(c)$ is

$\log \frac{p(x, c)}{p(x)p(c)} = \log p(c|x) - \log p(c)$

CLIP guidance

Replace the classifier in classifier guidance with a CLIP model

- Sample with a modified score:

\[
\nabla_{x_t} [\log p(x_t|c) + \omega \log p(c|x_t)] \\
= \nabla_{x_t} [\log p(x_t|c) + \omega (\log p(c|x_t) - \log p(c))] \\
= \nabla_{x_t} [\log p(x_t|c) + \omega (f(x_t) \cdot g(c))] \\
\]

• Fine-tune the model especially for inpainting: feed randomly occluded images with an additional mask channel as the input.

“an old car in a snowy forest”

“a man wearing a white hat”

Text-conditional image inpainting examples

DALL·E 2
OpenAI

1kx1k Text-to-image generation.
Outperform DALL-E (autoregressive transformer).

DALL·E 2

Model components

Prior: produces CLIP image embeddings conditioned on the caption.
Decoder: produces images conditioned on CLIP image embeddings and text.

Why conditional on CLIP image embeddings?

CLIP image embeddings capture high-level semantic meaning.

Latents in the decoder model take care of the rest.

The bipartite latent representation enables several text-guided image manipulation tasks.
Prior: produces CLIP image embeddings conditioned on the caption.

- Option 1. autoregressive prior: quantize image embedding to a seq. of discrete codes and predict them autoregressively.
- Option 2. diffusion prior: model the continuous image embedding by diffusion models conditioned on caption.
**DALL·E 2**

Model components (2/2): decoder model

Decoder: produces images conditioned on CLIP image embeddings (and text).

- Cascaded diffusion models: 1 base model (64x64), 2 super-resolution models (64x64 → 256x256, 256x256 → 1024x1024).
- Largest super-resolution model is trained on patches and takes full-res inputs at inference time.
- Classifier-free guidance & noise conditioning augmentation are important.

DALL·E 2

Bipartite latent representations

Bipartite latent representations $(z, x_T)$

$z$: CLIP image embeddings

$x_T$: inversion of DDIM sampler (latents in the decoder model)

Near exact reconstruction

DALL·E 2

Image variations

Fix the CLIP embedding $z$
Decode using different decoder latents $x_T$
DALL·E 2

Image interpolation

Interpolate image CLIP embeddings $z$.

Use different $x_T$ to get different interpolation trajectories.

DALL·E 2

Text Diffs

a photo of a cat → an anime drawing of a super saiyan cat, artstation

a photo of a victorian house → a photo of a modern house

a photo of an adult lion → a photo of lion cub

Change the image CLIP embedding towards the difference of the text CLIP embeddings of two prompts.

Decoder latent is kept as a constant.
Imagen
Google Research, Brain team

Input: text; Output: 1kx1k images

- An unprecedented degree of photorealism
  - SOTA automatic scores & human ratings
- A deep level of language understanding
- Extremely simple
  - no latent space, no quantization

A brain riding a rocketship heading towards the moon.

Imagen
Google Research, Brain team

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

Imagen

Google Research, Brain team

A dragon fruit wearing karate belt in the snow.

Imagen
Google Research, Brain team

A relaxed garlic with a blindfold reading a newspaper while floating in a pool of tomato soup.

[Image: A garlic with a blindfold reading a newspaper in a pool of tomato soup.]

A cute hand-knitted koala wearing a sweater with 'CVPR' written on it.
Imagen

Key modeling components:

- Cascaded diffusion models
- Classifier-free guidance and dynamic thresholding.
- Frozen large pretrained language models as text encoders. (T5-XXL)

Key observations:

- Beneficial to use text conditioning for all super-res models.
  - Noise conditioning augmentation weakens information from low-res models, thus needs text conditioning as extra information input.
- Scaling text encoder is extremely efficient.
  - More important than scaling diffusion model size.
- Human raters prefer T5-XXL as the text encoder over CLIP encoder on DrawBench.

Imagen

Dynamic thresholding

- Large classifier-free guidance weights $\rightarrow$ better text alignment, worse image quality

Imagen
Dynamic thresholding

- Large classifier-free guidance weights $\rightarrow$ better text alignment, worse image quality

- Hypothesis: at large guidance weight, the generated images are saturated due to the very large gradient updates during sampling

- Solution - dynamic thresholding: adjusts the pixel values of samples at each sampling step to be within a dynamic range computed over the statistics of the current samples.

Imagen
Dynamic thresholding

- Large class dates during
- Hypothesis sampling
- Solution - range com...

Static thresholding
Dynamic thresholding

Imagen

DrawBench: new benchmark for text-to-image evaluations

- A set of 200 prompts to evaluate text-to-image models across multiple dimensions.
  - E.g., the ability to faithfully render different colors, numbers of objects, spatial relations, text in the scene, unusual interactions between objects.
  - Contains complex prompts, e.g. long and intricate descriptions, rare words, misspelled prompts.

Imagen

DrawBench: new benchmark for text-to-image evaluations

- A set of 200 prompts
  - E.g., the interaction between objects.
  - Contains complex prompts, e.g., long and intricate descriptions, rare words, misspelled prompts.

Imagen got SOTA automatic evaluation scores on COCO dataset

Imagen is preferred over recent work by human raters in sample quality & image-text alignment on DrawBench.

<table>
<thead>
<tr>
<th>Model</th>
<th>FID-30K</th>
<th>Zero-shot FID-30K</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttnGAN [76]</td>
<td>35.49</td>
<td></td>
</tr>
<tr>
<td>DM-GAN [83]</td>
<td>32.64</td>
<td></td>
</tr>
<tr>
<td>DF-GAN [69]</td>
<td>21.42</td>
<td></td>
</tr>
<tr>
<td>DM-GAN + CL [78]</td>
<td>20.79</td>
<td></td>
</tr>
<tr>
<td>XMC-GAN [81]</td>
<td>9.33</td>
<td></td>
</tr>
<tr>
<td>LAFITE [82]</td>
<td>8.12</td>
<td></td>
</tr>
<tr>
<td>Make-A-Scene [22]</td>
<td>7.55</td>
<td></td>
</tr>
<tr>
<td>DALL-E [53]</td>
<td>17.89</td>
<td></td>
</tr>
<tr>
<td>LAFITE [82]</td>
<td>26.94</td>
<td></td>
</tr>
<tr>
<td>GLIDE [41]</td>
<td>12.24</td>
<td></td>
</tr>
<tr>
<td>DALL-E 2 [54]</td>
<td>10.39</td>
<td></td>
</tr>
<tr>
<td><strong>Imagen (Our Work)</strong></td>
<td><strong>7.27</strong></td>
<td><strong>7.27</strong></td>
</tr>
</tbody>
</table>

Stable Diffusion is a latent text-to-image diffusion model. Thanks to a generous compute donation from Stability AI and support from LAION, we were able to train a Latent Diffusion Model on 512x512 images from a subset of the LAION-5B database. Similar to Google's Imagen, this model uses a frozen CLIP ViT-L/14 text encoder to condition the model on text prompts. With its 860M UNet and 123M text encoder, the model is relatively lightweight and runs on a GPU with at least 10GB VRAM. See this section below and the model card.
Stable Diffusion
Latest & Publicly available text-to-image generation

HW assignment: Use stable diffusion API to generate ‘interesting’ image from text prompt. All submissions will be rated for top 3!
Outline for today’s class

• Theory:
  • How do we condition diffusion model?
  • How do people sample in practice? – DDIM sampling
  • How do we generate high-resolution images?

• Application:
  • Text to Image Generation
  • Image to Image Translation
  • Video Generation
Diffusion Autoencoders

Learning semantic meaningful latent representations in diffusion models

To be discussed in detail in paper presentation

Diffusion Autoencoders
Learning semantic meaningful latent representations in diffusion models

Very similar to StyleGAN based editing. $z_{sem}$ is the latent representation similar to the W/W+ space of StyleGAN

Diffusion Autoencoders

Learning semantic meaningful latent representations in diffusion models

Image super-resolution can be considered as training $p(x|y)$ where $y$ is a low-resolution image and $x$ is the corresponding high-resolution image.

Train a score model for $x$ conditioned on $y$ using:

$$
\mathbb{E}_{x,y} \mathbb{E}_{\epsilon \sim \mathcal{N}(0,1)} \mathbb{E}_t \| \epsilon_\theta(x_t, t; y) - \epsilon \|_p^2
$$

The conditional score is simply a U-Net with $x_t$ and $y$ (resolution image) concatenated.
Super-Resolution
Super-Resolution via Repeated Refinement (SR3)

Natural Image Super-Resolution $64 \times 64 \rightarrow 256 \times 256$

Bicubic  Regression  SR3 (ours)  Reference

Saharia et al., Image Super-Resolution via Iterative Refinement, 2021
Many image-to-image translation applications can be considered as training $p(x|y)$ where $y$ is the input image.

For example, for colorization, $x$ is a colored image and $y$ is a gray-level image.

Train a score model for $x$ conditioned on $y$ using:

$$\mathbb{E}_{x,y} \mathbb{E}_{\epsilon \sim \mathcal{N}(0,I)} \mathbb{E}_t \left\| \epsilon_{\theta}(x_t, t; y) - \epsilon \right\|_P$$

The conditional score is simply a U-Net with $x_t$ and $y$ concatenated.
Image-to-Image Translation

Palette: Image-to-Image Diffusion Models

Saharia et al., Palette: Image-to-Image Diffusion Models, 2022
Conditional Generation
Iterative Latent Variable Refinement (ILVR)

To be discussed in detail in paper presentation

A simple technique to guide the generation process of an unconditional diffusion model using a reference image.

Given the conditioning (reference) image $y$ the generation process is modified to pull the samples towards the reference image.

```latex
\textbf{for } t = T, \ldots, 1 \textbf{ do}
\begin{align*}
    z &\sim N(0, I) \\
    x'_{t-1} &\sim p_{\theta}(x'_{t-1}|x_t) \\
    y_{t-1} &\sim q(y_{t-1}|y) \\
    x_{t-1} &\leftarrow \phi_N(y_{t-1}) + x'_{t-1} - \phi_N(x'_{t-1})
\end{align*}
\textbf{end for}
```

Unconditional proposal $\triangleright$
Condition encoding $\triangleright$

Low-pass filter
Downsampling / Upsampling by a factor of $N$

Choi et al., ILVR: Conditioning Method for Denoising Diffusion Probabilistic Models, ICCV 2021
Conditional Generation

Iterative Latent Variable Refinement (ILVR)

(a) Generation from various downsampling factors

(b) Image Translation

(c) Paint-to-Image

(d) Editing with Scribbles

Choi et al., ILVR: Conditioning Method for Denoising Diffusion Probabilistic Models, ICCV 2021
Semantic Segmentation
Label-efficient semantic segmentation with diffusion models

Can we use representation learned from diffusion models for downstream applications such as semantic segmentation?

Baranchuk et al., Label-Efficient Semantic Segmentation with Diffusion Models, ICLR 2022
The experimental results show that the proposed method outperforms Masked Autoencoders, GAN and VAE-based models.

### Results on Different Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Classes</th>
<th>Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFHQ</td>
<td>34 classes</td>
<td><img src="image1.png" alt="Images" /> <img src="image2.png" alt="Images" /> <img src="image3.png" alt="Images" /> <img src="image4.png" alt="Images" /></td>
</tr>
<tr>
<td>CelebAMask</td>
<td>19 classes</td>
<td><img src="image5.png" alt="Images" /> <img src="image6.png" alt="Images" /> <img src="image7.png" alt="Images" /> <img src="image8.png" alt="Images" /></td>
</tr>
<tr>
<td>LSUN-Bedroom</td>
<td>28 classes</td>
<td><img src="image9.png" alt="Images" /> <img src="image10.png" alt="Images" /> <img src="image11.png" alt="Images" /> <img src="image12.png" alt="Images" /></td>
</tr>
<tr>
<td>ADE-Bedroom</td>
<td>30 classes</td>
<td><img src="image13.png" alt="Images" /> <img src="image14.png" alt="Images" /> <img src="image15.png" alt="Images" /> <img src="image16.png" alt="Images" /></td>
</tr>
<tr>
<td>LSUN-Cat</td>
<td>15 classes</td>
<td><img src="image17.png" alt="Images" /> <img src="image18.png" alt="Images" /> <img src="image19.png" alt="Images" /> <img src="image20.png" alt="Images" /></td>
</tr>
<tr>
<td>LSUN-Horse</td>
<td>21 classes</td>
<td><img src="image21.png" alt="Images" /> <img src="image22.png" alt="Images" /> <img src="image23.png" alt="Images" /> <img src="image24.png" alt="Images" /></td>
</tr>
</tbody>
</table>

*Baranchuk et al., Label-Efficient Semantic Segmentation with Diffusion Models, ICLR 2022*
Image Editing

SDEdit

Goal: Given a stroke painting with color, generate a photorealistic image

Key Idea:

- Latent Distribution of stroke and real images do not overlap.
- But once we apply forward diffusion on them, their distribution start overlapping as finally it becomes gaussian noise.

Meng et al., SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations, ICLR 2022
Image Editing

SDEdit

Meng et al., SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations, ICLR 2022
Outline for today’s class

• Theory:
  • How do we condition diffusion model?
  • How do people sample in practice? – DDIM sampling
  • How do we generate high-resolution images?

• Application:
  • Text to Image Generation
  • Image to Image Translation
  • Video Generation
Video Generation

Samples from a text-conditioned video diffusion model, conditioned on the string *fireworks*.

Ho et al., “Video Diffusion Models”, arXiv, 2022
Harvey et al., “Flexible Diffusion Modeling of Long Videos”, arXiv, 2022
Voleti et al., “MCVD: Masked Conditional Video Diffusion for Prediction, Generation, and Interpolation”, arXiv, 2022
Video Generation

Video Generation Tasks:

• Unconditional Generation (Generate all frames)
• Future Prediction (Generate future from past frames)
• Past Prediction (Generate past from future frames)
• Interpolation (Generate intermediate frames)

Learn a model of the form:

$$p_\theta(x^{t_1}, \ldots, x^{t_K} | x^{T_1}, \ldots, x^{T_M})$$

Given frames: $x^{T_1}, \ldots, x^{T_M}$
Frames to be predicted: $x^{t_1}, \ldots, x^{t_K}$

Ho et al., “Video Diffusion Models”, arXiv, 2022
Harvey et al., “Flexible Diffusion Modeling of Long Videos”, arXiv, 2022
Voleti et al., “MCVD: Masked Conditional Video Diffusion for Prediction, Generation, and Interpolation”, arXiv, 2022
Video Generation

Learn one model for everything:

- Architecture as one diffusion model over all frames concatenated.
- Mask frames to be predicted; provide conditioning frames; vary applied masking/conditioning for different tasks during training.
- Use time position encodings to encode times.

Video Generation

Architecture Details:

Data is 4D (image height, image width, #frames, channels)

- Option (2): Spatial 2D Convolutions + Attention Layers along frame axis.

⇒ Additional Advantage:

Ignoring the attention layers, the model can be trained additionally on pure image data!

Video Generation

Results

Long term video generation in hierarchical manner:

- 1. Generate future frames in sparse manner, conditioning on frames far back
- 2. Interpolate in-between frames

1+ hour coherent video generation possible!

Make-A-Video

Make-A-Video research builds on the recent progress made in text-to-image generation technology built to enable text-to-video generation. The system uses images with descriptions to learn what the world looks like and how it is often described. It also uses unlabeled videos to learn how the world moves. With this data, Make-A-Video lets you bring your imagination to life by generating whimsical, one-of-a-kind videos with just a few words or lines of text.

A dog wearing a Superhero outfit with red cape flying through the sky
Figure 2: Make-A-Video high-level architecture. Given input text $x$ translated by the prior $P$ into an image embedding, and a desired frame rate $fps$, the decoder $D^t$ generates 16 $64 \times 64$ frames, which are then interpolated to a higher frame rate by $\uparrow_F$, and increased in resolution to $256 \times 256$ by $SR^t_I$ and $768 \times 768$ by $SR_h$, resulting in a high-spatiotemporal-resolution generated video $\hat{y}$. 
Open Problems

• Sampling from diffusion model is still slow (even with DDIM you need 250 steps)
  • How can one sample with even fewer steps?

• How good is the latent space of diffusion model for downstream tasks?
  • ResNet on ImageNet gives us great image features
  • LLM gives great text features
  • Can diffusion model beat imagenet feature?
  • Can diffusion model help us in discriminative tasks?
Open Problems

• How do we better control diffusion model for conditional generation?
  • With GANs we can do very fine-grained condition and can even use 3D intrinsics.
  • While Diffusion model produces impressive image quality, it often ignores detailed conditioning like segmentation mask, and works best for vague conditioning like text or class.
  • How do we enable that?

• Can we generate more involved videos with diffusion models?
  • This is something that is limited with GAN
  • Current diffusion models are limited in the kinds of video in can generate, but already a huge progress from GAN

• Many many cool and creative applications that we haven’t imagined before due to lack of such a powerful tool!
GAN vs Diffusion Model

Do you think GAN would disappear in few years and Diffusion model would be the de-facto for generative models?