High-Resolution Image Synthesis with Latent Diffusion Models

Robin Rombach\textsuperscript{1} *  Andreas Blattmann\textsuperscript{1} *  Dominik Lorenz\textsuperscript{1}  Patrick Esser\textsuperscript{2}  Björn Ommer\textsuperscript{1}

\textsuperscript{1}Ludwig Maximilian University of Munich & IWR, Heidelberg University, Germany  \textsuperscript{2}Runway ML

https://github.com/CompVis/latent-diffusion

<table>
<thead>
<tr>
<th>Input</th>
<th>ours ((f = 4))</th>
<th>DALL-E ((f = 8))</th>
<th>VQGAN ((f = 16))</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR: 27.4</td>
<td>PSNR: 22.8</td>
<td>PSNR: 19.9</td>
<td>R-FID: 0.58</td>
</tr>
</tbody>
</table>
Research behind Stable Diffusion
Stable Diffusion Public Release

It is our pleasure to announce the public release of stable diffusion following our release for researchers [https://stability.ai/blog/stable-diffusion-announcement]
Image Synthesis

• Settings
  • Conditional Image Generation
    • Text-to-Img —— DALLE Series, DaVinci, Stable Diffusion
    • Inpainting —— LAMA
    • Super Resolution —— SR3
    • ……

Text2Image Example

Inpainting Example
Diffusion Models

• DDPM
  • Noising and Denoising, Markov Chain
  ➢ Noising, No training required

➢ Denoising: Probabilistic Prediction
  • Target: $L_{DM} = \mathbb{E}_{\epsilon, \epsilon \sim \mathcal{N}(0,1), t} \left[ ||\epsilon - \epsilon_\theta(x_t, t)||^2 \right]$
  • $\epsilon_\theta$: U-Net
    • Whole Image to Whole Image

---

Algorithm 1 Training
1: repeat
2: $x_0 \sim q(x_0)$
3: $t \sim \text{Uniform}\{1, \ldots, T\}$
4: $\epsilon \sim \mathcal{N}(0, I)$
5: Take gradient descent step on
6: $\nabla_\theta \left[ ||\epsilon - \epsilon_\theta(\sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}\epsilon, t)||^2 \right]$ until converged
7: end repeat

Algorithm 2 Sampling
1: $x_T \sim \mathcal{N}(0, I)$
2: for $t = T, \ldots, 1$ do
3: $z \sim \mathcal{N}(0, I)$ if $t > 1$, else $z = 0$
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: end for
6: return $x_0$
Diffusion Models

• DDPM
  • Noising and Denoising, Markov Chain
  ➢ Noising, No training required

➢ Denoising: Probabilistic Prediction
  • Target: $L_{DM} = \mathbb{E}_{x_t, \epsilon \sim \mathcal{N}(0, I), t} \left[ \| \epsilon - \epsilon_\theta(x_t, t) \|^2_2 \right]$
  • $\epsilon_\theta$: U-Net
    • Whole Image to Whole Image

---

**Algorithm 1 Training**

1: repeat
2:  $x_0 \sim q(x_0)$
3:  $t \sim \text{Uniform}(\{1, \ldots, T\})$
4:  $\epsilon \sim \mathcal{N}(0, I)$
5:  Take gradient descent step on $\nabla_\theta \| \epsilon - \epsilon_\theta(\sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t} \epsilon, q_\theta) \|^2$
6: until converged

**Algorithm 2 Sampling**

1: $x_T \sim \mathcal{N}(0, I)$
2: for $t = T, \ldots, 1$ do
3:  $z \sim \mathcal{N}(0, I)$ if $t > 1$, else $z = 0$
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: end for
6: return $x_0
Drawbacks and Improvements

• Slow — Thousands of A100 hours for training, several hours for evaluation
  • …...
  • **DDIM:** 20-50 times speed up!
    • DDPM time steps $T = 1000$
    • DDIM $\text{dim}(T) = T/20$

• Resolution Limitations
  • U-Net for whole image * T steps
  • Solution: Denoising on latent space —— LDM(Latent Diffusion Models)
LDM

- Two Stage Training for both **Pixel Space** and **Latent Space**
- Stage 1 —— Image AutoEncoder
- Stage 2 —— Denoising U-Net

![Diagram](image-url)

**Stage 1**

**Stage 2**
LDM

- **Stage 1 —— Perceptual Image Compression**
  - Encoder – Decoder model
    - Image Generation from statistics encoding and generation from distribution
    - Latent code is not high level feature ($Z_c = 4$)
    - Load in an Gaussian Distribution and send into decoder to get image output

- **Constraint Target (Loss Function)**
  - Reconstruction Loss (L2 for original image and reconstructed one)
  - LPIPS Loss (Perceptual loss for better reality)
    - Pretrained VGG 16
  - KL Loss for constraints on Latent Space
LDM

- Stage 2 —— Latent Diffusion Models
  - With Stage 1 Autoencoder frozen
  - Effect on different size latent code (with different rate of encoder)
  - Structure
    - Resnet Block with cross-attention module
    - U-Net like
  - Diffusion
    - Target: $L_{LDM} := \mathbb{E}_{(x, \epsilon) \sim \mathcal{N}(0,1)} \left[ \left\| \epsilon - \epsilon_{\theta}(z_t, t) \right\|_2^2 \right]$
    - Can receive Conditioning
      - Text
      - Images
      - Semantic maps
      - ......
LDM

• Conditioning Mechanisms
  • Different (domain specific) $\tau_\theta$ for different modalities
  • Using Attention to let conditioning effects on generation process:
    \[
    \text{Attention}(Q, K, V) = \text{softmax} \left( \frac{Q K^T}{\sqrt{d}} \right) \cdot V, \text{ with}
    \]
    \[
    Q = W_Q^{(i)} \cdot \varphi_i(z_t), \quad K = W_K^{(i)} \cdot \tau_\theta(y), \quad V = W_V^{(i)} \cdot \tau_\theta(y).
    \]
  • Optimization target comes to:
    \[
    L_{LDM} := \mathbb{E}_{\varepsilon(x), y, \varepsilon \sim N(0,1), t} \left[ \| \varepsilon - \varepsilon_\theta(z_t, t, \tau_\theta(y)) \|_2^2 \right]
    \]
Experiments

• On Perceptual Compression Tradeoffs
  • Ablation on different downsampling factor \( f \in \{1, 2, 4, 8, 16, 32\} \)

• Image Generation with Latent Diffusion
  • On four regular datasets

• Conditional Latent Diffusion
  • Text to Image
  • Layout to Image
  • Semantic Map to Image

• SR and Inpainting
Experiments

• On Perceptual Compression Tradeoffs

Figure 6. Analyzing the training of class-conditional LDMs with different downsampling factors $f$ over 2M train steps on the ImageNet dataset. Pixel-based LDM-1 requires substantially larger train times compared to models with larger downsampling factors (LDM-{$4$-$16$}). Too much perceptual compression as in LDM-32 limits the overall sample quality. All models are trained on a single NVIDIA A100 with the same computational budget. Results obtained with 100 DDIM steps [84] and $\kappa = 0$.

Figure 7. Comparing LDMs with varying compression on the CelebA-HQ (left) and ImageNet (right) datasets. Different markers indicate {10, 20, 50, 100, 200} sampling steps using DDIM, from right to left along each line. The dashed line shows the FID scores for 200 steps, indicating the strong performance of LDM-$\{4$-$8\}$. FID scores assessed on 5000 samples. All models were trained for 500k (CelebA) / 2M (ImageNet) steps on an A100.
Experiments

• Image Generation with Latent Diffusion

<table>
<thead>
<tr>
<th>Method</th>
<th>CelebA-HQ 256 × 256</th>
<th>FFHQ 256 × 256</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FID ↓</td>
<td>Prec. ↑</td>
</tr>
<tr>
<td>DC-VAE [63]</td>
<td>15.8</td>
<td>-</td>
</tr>
<tr>
<td>VQGAN+T. [23] (k=400)</td>
<td>10.2</td>
<td>-</td>
</tr>
<tr>
<td>PGGAN [39]</td>
<td>8.0</td>
<td>-</td>
</tr>
<tr>
<td>LSGM [93]</td>
<td>7.22</td>
<td>-</td>
</tr>
<tr>
<td>UDM [43]</td>
<td>7.16</td>
<td>-</td>
</tr>
<tr>
<td><strong>LDM-4 (ours, 500-s†)</strong></td>
<td><strong>5.11</strong></td>
<td><strong>0.72</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>LSUN-Churches 256 × 256</th>
<th>LSUN-Bedrooms 256 × 256</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FID ↓</td>
<td>Prec. ↑</td>
</tr>
<tr>
<td>DDPM [30]</td>
<td>7.89</td>
<td>-</td>
</tr>
<tr>
<td>ImageBART [21]</td>
<td>7.32</td>
<td>-</td>
</tr>
<tr>
<td>PGGAN [39]</td>
<td>6.42</td>
<td>-</td>
</tr>
<tr>
<td>StyleGAN [41]</td>
<td>4.21</td>
<td>-</td>
</tr>
<tr>
<td>StyleGAN2 [42]</td>
<td>3.86</td>
<td>-</td>
</tr>
<tr>
<td>ProjectedGAN [76]</td>
<td><strong>1.59</strong></td>
<td><strong>0.61</strong></td>
</tr>
<tr>
<td><em><em>LDM-8</em> (ours, 200-s)</em>*</td>
<td><strong>4.02</strong></td>
<td><strong>0.64</strong></td>
</tr>
</tbody>
</table>
Experiments

• Conditional Latent Diffusion
  • Text to Image
    • BERT Tokenizer, Train on LAION-400M

<table>
<thead>
<tr>
<th>Method</th>
<th>FID</th>
<th>IS</th>
<th>Nparams</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CogView$^\dagger$ [17]</td>
<td>27.10</td>
<td>18.20</td>
<td>4B</td>
<td>self-ranking, rejection rate 0.017</td>
</tr>
<tr>
<td>LAFITE$^\dagger$ [109]</td>
<td>26.94</td>
<td>26.02</td>
<td>75M</td>
<td></td>
</tr>
<tr>
<td>GLIDE* [59]</td>
<td>12.24</td>
<td>-</td>
<td>6B</td>
<td>277 DDIM steps, c.f.g. [32] $s = 3$</td>
</tr>
<tr>
<td>Make-A-Scene* [26]</td>
<td>11.84</td>
<td>-</td>
<td>4B</td>
<td>c.f.g for AR models [98] $s = 5$</td>
</tr>
<tr>
<td>LDM-KL-8</td>
<td>23.31</td>
<td>20.03 $\pm 0.33$</td>
<td>1.45B</td>
<td>250 DDIM steps</td>
</tr>
<tr>
<td>LDM-KL-8:G*</td>
<td>12.63</td>
<td>30.29 $\pm 0.42$</td>
<td>1.45B</td>
<td>250 DDIM steps, c.f.g. [32] $s = 1.5$</td>
</tr>
</tbody>
</table>

Table 2. Evaluation of text-conditional image synthesis on the 256 × 256-sized MS-COCO [51] dataset: with 250 DDIM [84] steps our model is on par with the most recent diffusion [59] and autoregressive [26] methods despite using significantly less parameters. $^\dagger$*/: Numbers from [109]/ [26]

Text-to-Image Synthesis on LAION. 1.45B Model.
Experiments

- Conditional Latent Diffusion
  - Layout to Image
  - Semantic map to Image
Experiments

• Super Resolution & Inpainting

<table>
<thead>
<tr>
<th>Method</th>
<th>FID ↓</th>
<th>IS ↑</th>
<th>PSNR ↑</th>
<th>SSIM ↑</th>
<th>Nparams</th>
<th>(\frac{\text{ Params}}{\text{ MB}})</th>
<th>(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Regression [72]</td>
<td>15.2</td>
<td>121.1</td>
<td>27.9</td>
<td>0.801</td>
<td>625M</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>SR3 [72]</td>
<td>5.2</td>
<td>180.1</td>
<td>26.4</td>
<td>0.762</td>
<td>625M</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>LDM-4 (ours, 100 steps)</td>
<td>2.8(^{1}), 4.8(^{1})</td>
<td>166.3</td>
<td>24.4±3.8</td>
<td>0.69±0.14</td>
<td>169M</td>
<td>4.62</td>
<td></td>
</tr>
<tr>
<td>emphLDM-4 (ours, big, 100 steps)</td>
<td>2.4(^{1}), 4.3(^{1})</td>
<td>174.9</td>
<td>24.7±4.1</td>
<td>0.71±0.15</td>
<td>552M</td>
<td>4.5</td>
<td></td>
</tr>
<tr>
<td>LDM-4 (ours, 50 steps, guiding)</td>
<td>4.4(^{1}), 6.4(^{1})</td>
<td>153.7</td>
<td>25.8±2.2</td>
<td>0.74±0.12</td>
<td>184M</td>
<td>0.38</td>
<td></td>
</tr>
</tbody>
</table>

Table 5. \(\times 4\) upscaling results on ImageNet-Val. (256\(^2\)); \(^{1}\): FID features computed on validation split, \(^{\dagger}\): FID features computed on train split, \(*\): Assessed on a NVIDIA A100

<table>
<thead>
<tr>
<th>Method</th>
<th>FID ↓</th>
<th>LPIPS ↓</th>
<th>(\frac{\text{ Params}}{\text{ MB}})</th>
<th>(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDM-4 (ours, big, w/ ft)</td>
<td>9.39</td>
<td>0.246±0.042</td>
<td>0.137±0.080</td>
<td></td>
</tr>
<tr>
<td>LDM-4 (ours, big, w/o ft)</td>
<td>12.89</td>
<td>0.257±0.047</td>
<td>0.142±0.085</td>
<td></td>
</tr>
<tr>
<td>LDM-4 (ours, w/ attn)</td>
<td>11.87</td>
<td>0.259±0.042</td>
<td>0.144±0.084</td>
<td></td>
</tr>
<tr>
<td>LDM-4 (ours, w/o attn)</td>
<td>12.60</td>
<td>0.259±0.041</td>
<td>0.145±0.084</td>
<td></td>
</tr>
<tr>
<td>LaMa [88](^{\dagger})</td>
<td>12.31</td>
<td>0.243±0.038</td>
<td>2.23</td>
<td>0.134±0.080</td>
</tr>
<tr>
<td>LaMa [88]</td>
<td>12.0</td>
<td>0.24</td>
<td>2.21</td>
<td>0.14</td>
</tr>
<tr>
<td>CoModGAN [107]</td>
<td>10.4</td>
<td>0.26</td>
<td>1.82</td>
<td>0.15</td>
</tr>
<tr>
<td>RegionWise [52]</td>
<td>21.3</td>
<td>0.27</td>
<td>4.75</td>
<td>0.15</td>
</tr>
<tr>
<td>DeepFill v2 [104]</td>
<td>22.1</td>
<td>0.28</td>
<td>5.20</td>
<td>0.16</td>
</tr>
<tr>
<td>EdgeConnect [58]</td>
<td>30.5</td>
<td>0.28</td>
<td>8.37</td>
<td>0.16</td>
</tr>
</tbody>
</table>

Table 6. Assessing inpainting efficiency. \(^{\dagger}\): Deviations from Fig. 7 due to varying GPU settings/batch sizes cf. the supplement.

Table 7. Comparison of inpainting performance on 30k crops of size 512×512 from test images of Places [108]. The column 40-50% reports metrics computed over hard examples where 40-50% of the image region have to be inpainted. \(^{\dagger}\) recomputed on our test set, since the original test set used in [88] was not available.
Thanks for watching!