High-Resoluton Image Synthesis with Latent Diffusion Model

Suk-Hwan Lee

Artificial Intelligence

Creating the Future

Dong-A University

Division of Computer Engineering & Artificial Intelligence

References

Robin Rombach*, Andreas Blattmann*, Dominik Lorenz, Patrick Esser, Björn Ommer, "High-Resolution Image Synthesis with **Latent Diffusion Models**," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 10684-10695

https://github.com/CompVis/latent-diffusion
https://ommer-lab.com/research/latent-diffusion-models/

Stable Diffusion v1 (ComVis)

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

Stable Diffusion v2 (Stability-AI)

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

labml : https://nn.labml.ai/

Diffusion Model

https://nn.labml.ai/diffusion/stable_diffusion/index.html

High-Resoluton Image Synthesis with Latent Diffusion Model

Abstract

- **Diffusion Models (DMs)** : achieve state-of-the-art synthesis results on image data and beyond
  - Decompose the **image formation process** into **a sequential application of denoising autoencoders**,.
  - **Control the image generation process without retraining**.
  - [**Cons**] However, since these models typically **operate directly in pixel space**, optimization of powerful DMs often **consumes hundreds of GPU days** **and** **inference is expensive due to sequential evaluations**.

[Pros]

DM do **not** exhibit **mode-collapse** and **training instabilities** as GANs, being likelihood-based models.

Model highly complex distribution of natural images without involving billions of parameters as in AR models, **by heavily exploiting parameter sharing**.

- **Latent diffusion models (LDMs)**
  - To enable DM training on **limited computational resources** while **retaining their quality and flexibility**,
  - Apply them **in the latent space of powerful pretrained autoencoders**.
  - By introducing **cross-attention layers** into the model architecture, **turn diffusion models into powerful and flexible generators for general conditioning inputs** such as **text** or **bounding boxes** and **high-resolution synthesis** becomes possible in a convolutional manner.
  - Achieve a new state of the art for **image inpainting** and highly competitive performance on various tasks, including **unconditional image generation**, **semantic scene synthesis**, and **super-resolution**, while significantly reducing computational requirements compared to pixel-based DMs.

High-Resoluton Image Synthesis with Latent Diffusion Model

➢ **Democratizing High-Resolution Image Synthesis**

Diffusion Model

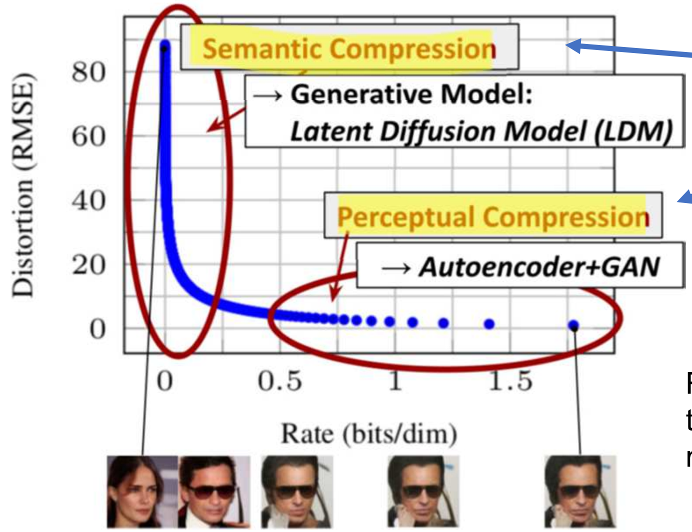
- Still **computationally demanding**, since **training and evaluating such a model requires repeated function evaluations** (and gradient computations) in the **high-dimensional space of RGB images**.
- Example
  - ✓ **Training** the most powerful DMs often takes **hundreds of GPU days** (e.g. 150-1000 V100 days in [15]) and repeated evaluations on a noisy version of the input space render
  - ✓ **Inference** expensive so that **producing 50k samples** takes approximately **5 days** [15] on **a single A100 GPU**

- Two consequences for the research community

1) Firstly, training such a model requires **massive computational resources** only available to a small fraction of the field, and leaves a huge carbon footprint [65, 86].
2) Secondly, **evaluating an already trained model** is also **expensive** in time and memory, since the same model architecture must run sequentially for a large number of steps (e.g. 25 - 1000 steps in [15]).

High-Resoluton Image Synthesis with Latent Diffusion Model

- **Departure to Latent Space**



Learn the sematic and conceptual composition of data

Remove high-frequency details but still learns little semantic variation

Rate-distortion trade-off of a trained model

Figure 2. Illustrating perceptual and semantic compression: Most bits of a digital image correspond to imperceptible details.

- While DMs allow to suppress this semantically meaningless information by minimizing the responsible loss term, gradients (during training) and the neural network backbone (training and inference) still need to be evaluated on all pixels, leading to superfluous computations and unnecessarily expensive optimization and inference.
- We propose latent diffusion models (LDMs) as an effective generative model and a separate mild compression stage that only eliminates imperceptible details. Data and images from [30].

Aim to first find a **perceptually equivalent**, but **computationally more suitable space**, in which we will train diffusion models for high-resolution image synthesis

- **Two distinct phases for training**

✓ **Train an autoencoder** which provides a **lower-dimensional** (and thereby efficient) **representational space** which is **perceptually equivalent to the data space**.

✓ **Train DMs** in **the learned latent space**, which exhibits **better scaling properties** with respect to the spatial dimensionality

- Notable advantage

✓ **Need to train the universe autoencoding stage only once and reuse it for multiple DM training**

✓ **Enable a large number of diffusion models for various image-to-image and test-to-image tasks**

High-Resoluton Image Synthesis with Latent Diffusion Model

➢ **Contributions**

I. **Scales more graceful to higher dimensional data** and can thus (a) **work on a compression level** which provides **more faithful and detailed reconstructions** than previous work (see Fig. 1) and (b) can be efficiently applied to **high-resolution synthesis of megapixel** images

II. Competitive performance on **multiple tasks** (unconditional image synthesis, inpainting, stochastic super-resolution) and **datasets** while significantly **lowering computational costs**.

III. Does **not require a delicate weighting of reconstruction and generative abilities**. This ensures extremely **faithful reconstructions** and requires **very little regularization of the latent space**

IV. Can be applied in a convolutional fashion and render large, consistent images of ~ 1024^2 px

V. Design a **general-purpose conditioning mechanism based on cross-attention**, enabling **multi-modal training**. We use it to train class-conditional, text-to-image and layout-to-image models.

Figure 1. Boosting the upper bound on achievable quality with less aggressive downsampling. Since diffusion models offer excellent inductive biases for spatial data, we do not need the heavy spatial downsampling of related generative models in latent space, but can still greatly reduce the dimensionality of the data via suitable autoencoding models, see Sec. 3. Images are from the DIV2K [1] validation set, evaluated at 512^2 px. We denote the spatial downsampling factor by f. Reconstruction FIDs [29] and PSNR are calculated on ImageNet-val. [12]; see also Tab. 8.

2. Related Work

Generative Models for Image Synthesis

- Explicit Model - Likelihood base Model : VAE(Variational autoencoders), DM(Diffusion Probabilistic Models) (SOTA in density estimation as well as sample quality) – 생성된 샘플에 대해 Likelihood Function을 최대화하는 방식
- Flow based ARM (Autoregressive models)
- Implicit Model : GAN(Generative Adversarial Networks) – Divergence 와 같은 부차적 방식을 통해 분포를 학습

Two-Stage Image Synthesis

- **VQ-VAE** : Use autoregressive model to learn an expressive prior over a discretized latent space. [66] extend this approach to text-to-image generation by learning a joint distribution over discretized image and text representations
- **VQ-GAN** : Employ a first stage with an adversarial and perceptual objective to scale autoregressive transformers to larger images. However, the **high compression rates** required for feasible ARM training, which introduces **billions of trainable parameters**.
- Our work prevents such tradeoffs, as our proposed **LDMs scale more gently to higher dimensional latent spaces** due to their **convolutional backbone**.
- Thus, we are free to choose the level of compression which optimally mediates between learning a powerful first stage, without leaving too much **perceptual compression** up to the generative diffusion model while guaranteeing **high fidelity reconstructions** (see Fig. 1).

2. Related Work

- **VQ-VAE**

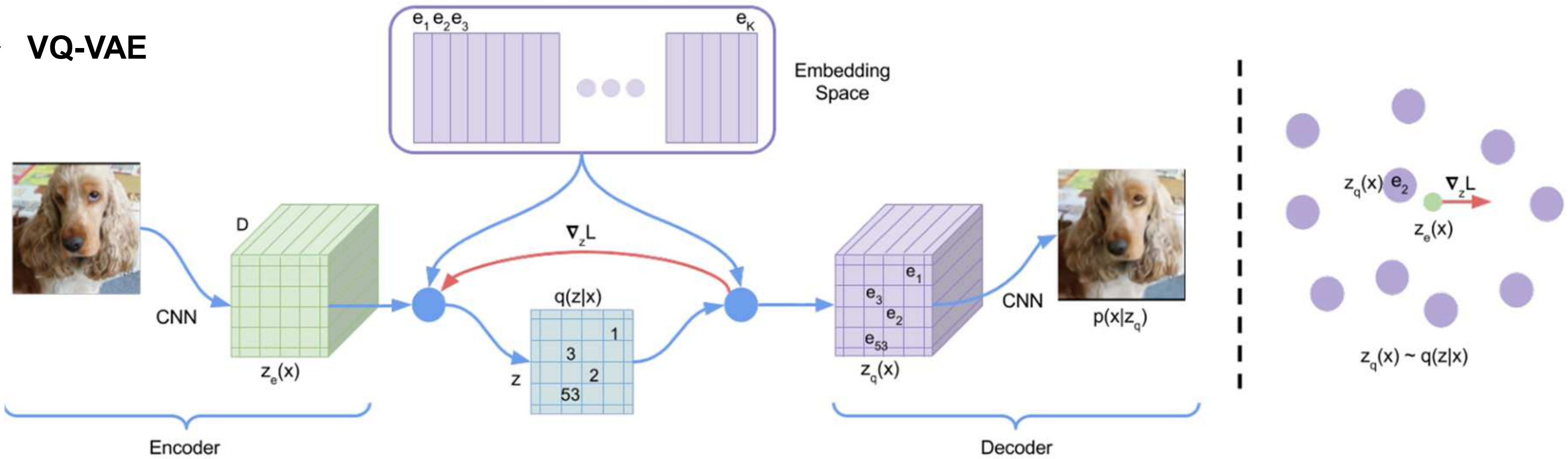


Figure 1: Left: A figure describing the VQ-VAE. Right: Visualisation of the embedding space. The output of the encoder $z(x)$ is mapped to the nearest point e_2. The gradient $\nabla_z L$ (in red) will push the encoder to change its output, which could alter the configuration in the next forward pass.

- **VQ-GAN**

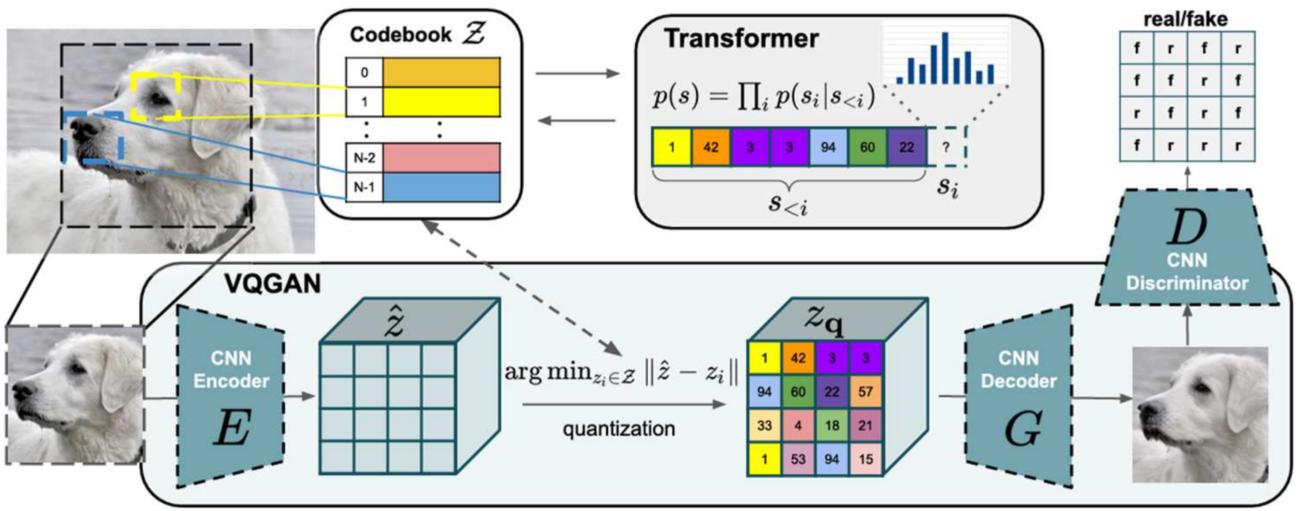


Figure 2. Our approach uses a convolutional VQ-GAN to **learn a codebook of context-rich visual parts**, whose **composition is subsequently modeled with an autoregressive transformer architecture.** A discrete codebook provides the interface between these architectures and a patch-based discriminator enables strong compression while retaining high perceptual quality. This method introduces the efficiency of convolutional approaches to transformer based high resolution image synthesis.

High-Resoluton Image Synthesis with Latent Diffusion Model

3. Method

- **To lower computational demand of training DM towards high-resolution image synthesis**
  - Although **diffusion models** allow to **ignore perceptually irrelevant details by *undersampling* the corresponding loss terms** [30], they still **require costly function evaluations in pixel space**, which causes huge demands in computation time and energy resources
  - Introduce an **explicit separation** of the **compressive** from the **generative** learning phase
  - We **utilize an autoencoding model** which **learns a space that is perceptually equivalent to the image space**, but **offers significantly reduced computational complexity**.

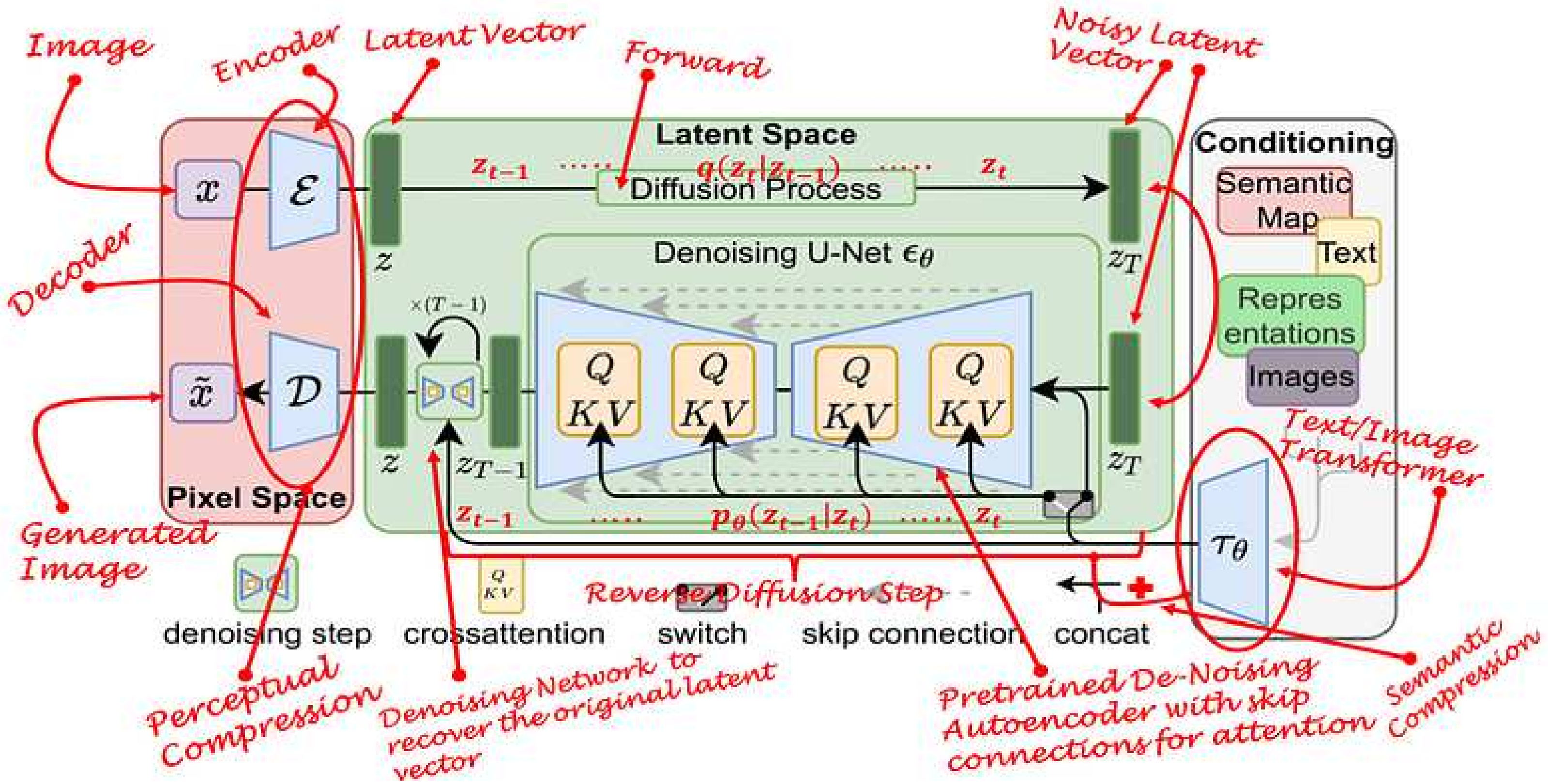
- **Several Advantages**

  I. By leaving the high-dimensional image space, we obtain DMs which are **computationally** much more **efficient** because **sampling is performed on a low-dimensional space**.

  II. **Exploit the inductive bias of DMs** inherited from **their UNet architecture** [71], which makes them particularly **effective for data with spatial structure** and therefore alleviates the need for aggressive, quality-reducing compression levels as required by previous approaches [23, 66].

  III. **Obtain general-purpose compression models** whose **latent space** can be used to train multiple generative models and which can also be utilized for other downstream applications such as single-image CLIP-guided synthesis [25].

3. Method



Latent Diffusion Model (Base Diagram:[3], Concept-Map Overlay: Author)

https://towardsdatascience.com/what-are-stable-diffusion-models-and-why-are-they-a-step-forward-for-image-generation-aa1182801d46

High-Resoluton Image Synthesis with Latent Diffusion Model

3.1 Perceptaul Image Compression

- Consists of **an autoencoder** trained by combination of **a perceptual loss** [106] and **a patch-based** [33] **adversarial objective** [20, 23, 103].
- This **ensures that the reconstructions are confined to the image manifold** by enforcing local realism and **avoids bluriness** introduced by relying solely on pixel-space losses such as L2 or L1 objectives

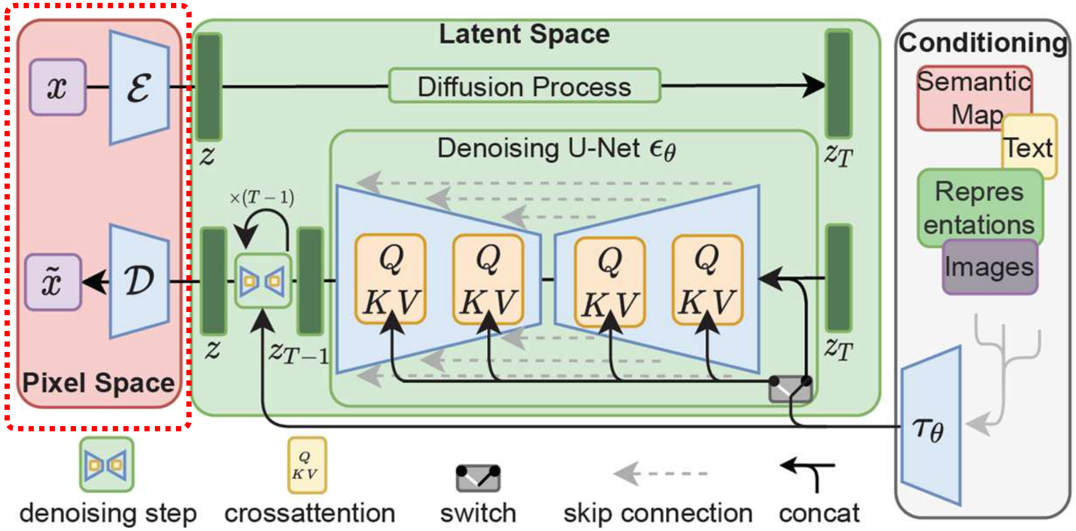


Figure 3. We condition LDMs either via concatenation or by a more general cross-attention mechanism.

- Encoder $\mathcal{E}$ encodes an image x into a latent representations z and Decoder D reconstructs the image from the latent

$$z = \mathcal{E}(x) \qquad \hat{x} = D(z) = D(\mathcal{E}(x))$$

$$x \in \mathbb{R}^{H \times W \times 3} \qquad z \in \mathbb{R}^{h \times w \times c}$$

- Encoder *downsamples* the image by a factor $f = \frac{H}{h} = \frac{W}{w}$, and different downsampling factors $f = 2^m$ with $m \in \mathbb{N}$

- **To avoid arbitrarily high-variance latent spaces**, we experiment with two different kinds of regularizations; ***KL-reg*** and ***VQ-reg***

✓ ***KL-reg*** : a slight KL-penalty towards a standard normal on the learned latent, similar to a VAE

✓ ***VQ-reg*** : Uses a vector quantization layer [96] within the decoder, which can be interpreted as a VQGAN [23] but with the quantization layer absorbed by the decoder.

3.2 Latent Diffusion Models

Diffusion Model

- Probabilistic models designed to learn a data distribution $p(x)$ by gradually denoising a normally distributed variable, which corresponds to learning the reverse process of a fixed Markov Chain of length T
- For image synthesis, **rely on a reweighted variant of the variational lower bound on $p(x)$**, which mirrors denoising score-matching
- Interpreted as an ***equally weighted sequence of denoising autoencoders***; $\epsilon_\theta(x_t, t); t = 1, \dots, T$, trained to predict a denoised variant of their input x_t, noisy version of x
- The corresponding objective

$$L_{DM} = \mathbb{E}_{x,\epsilon\sim\mathcal{N}(0,1),t}\Big[\|\epsilon - \epsilon_\theta(x_t, t)\|_2^2\Big] \tag{1}$$

with t uniformly sampled from $\{1, \dots, T\}$

Generative Modeling of Latent Representation

- Access to an efficient, **low-dimensional latent space** in which high-frequency, imperceptible details are abstracted away.
  ① **Focus on** the important, **semantic bits of the data**
  ② **Train in a lower dimensional**, computationally much more efficient **space**
- **Image-specific inductive biases**; build the underlying **UNet** primarily from 2D convolutional layers, and further focusing **the objective** on **the perceptually most relevant bits** using the reweighted bound

$$L_{LDM} := \mathbb{E}_{\mathcal{E}(x),\epsilon\sim\mathcal{N}(0,1),t}\Big[\|\epsilon - \epsilon_\theta(z_t, t)\|_2^2\Big] \tag{2}$$

- Neural backbone $\epsilon_\theta(z_t, t)$: Time-conditional UNet
- Since the forward process is fixed, z_t can be efficiently obtained from $\mathcal{E}$ during training, and samples from $p(z)$ can be decoded to image space with a single pass through $\mathcal{D}$.

3.3 Conditioning Mechanisms

- **Diffusion models** are capable of **modeling conditional distributions of the form $p(z|y)$**. Be implemented with ***a conditional denoising autoencoder*** $\epsilon_\theta(z_t, t, y)$ by controlling the synthesis process through inputs y such as text [68], semantic maps [33, 61] or other image-to-image translation tasks [34]

➢ **Turn DMs into more flexible conditional image generators**

- By augmenting their underlying **UNet backbone with the cross-attention mechanism** [97], which is effective for learning attention-based models of various input modalities [35,36].

- To pre-process y from various modalities (such as language prompts), we introduce **a domain specific encoder τ_θ** that **projects y to an intermediate representation $\tau_\theta(y) \in \mathbb{R}^{M \times d_\tau}$**, which is then **mapped to the intermediate layers of the UNet** via **a cross-attention layer** implementing

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V.$$

$$Q = W_Q^{(i)} \cdot \varphi_i(z_t), \; K = W_K^{(i)} \cdot \tau_\theta(y), \; V = W_V^{(i)} \cdot \tau_\theta(y)$$

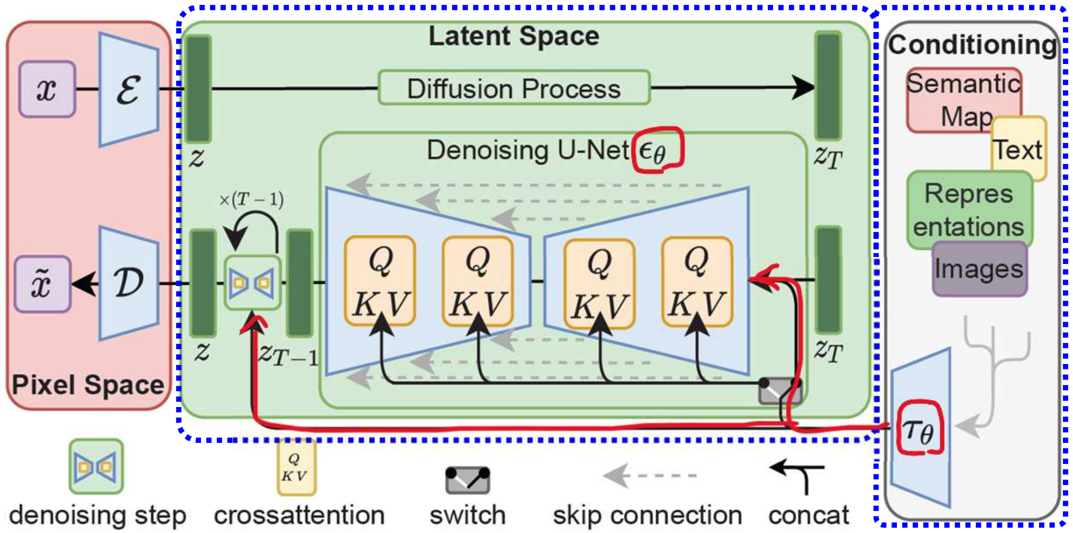


Figure 3. We condition LDMs either via concatenation or by a more general cross-attention mechanism.

✓ $\varphi_i(z_t) \in \mathbb{R}^{N \times d_\epsilon^i}$: a (flattened) intermediate representation of the UNet implementing ϵ_θ

✓ $W_V^{(i)} \in \mathbb{R}^{d \times d_\epsilon^i}$, $W_Q^{(i)} \in \mathbb{R}^{d \times d_\tau}$, $W_K^{(i)} \in \mathbb{R}^{d \times d_\tau}$: Learnable projection matrices

High-Resoluton Image Synthesis with Latent Diffusion Model

3.3 Conditioning Mechanisms

➢ **Conditional LDM**

$$L_{LDM} := \mathbb{E}_{\mathcal{E}(x), y, \epsilon \sim \mathcal{N}(0,1), t}\Big[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))\|_2^2\Big] \qquad (3)$$

$$L_{LDM} := \mathbb{E}_{\mathcal{E}(x), \epsilon \sim \mathcal{N}(0,1), t}\Big[\|\epsilon - \epsilon_\theta(z_t, t)\|_2^2\Big]$$

- Both τ_θ and ϵ_θ are jointly optimized via Eq. 3.
- This conditioning mechanism is flexible as τ_θ can be parameterized with domain-specific experts, e.g. (unmasked) *transformers* [97] when y are text prompts

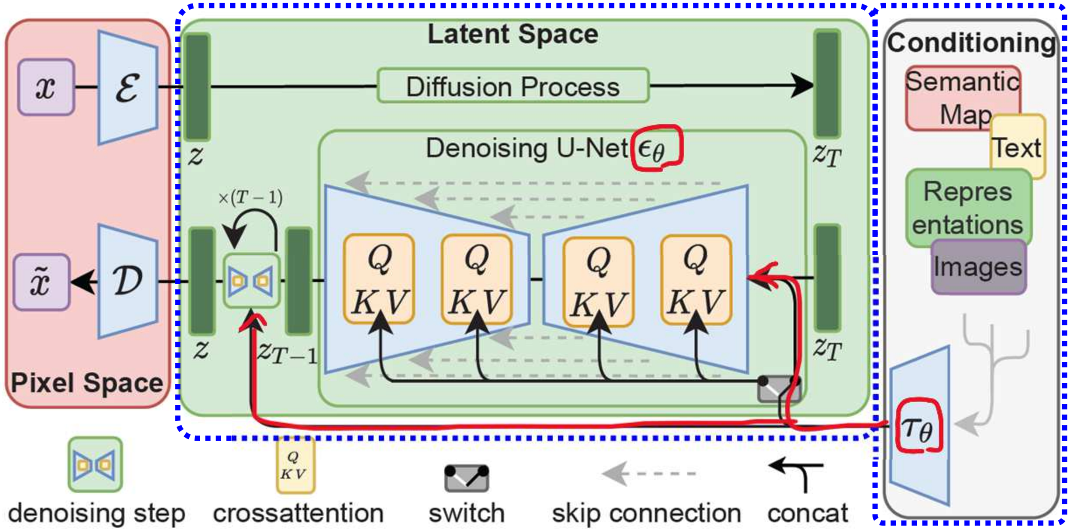


Figure 3. We condition LDMs either via concatenation or by a more general cross-attention mechanism.

$$L_{DM} = \mathbb{E}_{x, \epsilon \sim \mathcal{N}(0,1), t}\Big[\|\epsilon - \epsilon_\theta(x_t, t)\|_2^2\Big]$$

$$L_{LDM} := \mathbb{E}_{\mathcal{E}(x), \epsilon \sim \mathcal{N}(0,1), t}\Big[\|\epsilon - \epsilon_\theta(z_t, t)\|_2^2\Big]$$

Latent Vector

Latent State of the Denoising U-Net at time 't'

Domain Specific Encoder (i.e. Transformer) on a condition 'y'

Conditional Loss Function

$$L_{LDM} := \mathbb{E}_{\mathcal{E}(x), y, \epsilon \sim \mathcal{N}(0,1), t}\Big[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))\|_2^2\Big]$$

4. Experiments

4.1 On Perceptual Compression Tradeoffs

- LDMs with different downsampling factors $f = \{1,2,4,8,16,32\}$ (LDM-f) (LDM-1 correspond to pixel-based DMs)
- Perceptual compression Test : A signle NVIDIA A100, same number of steps and parameters

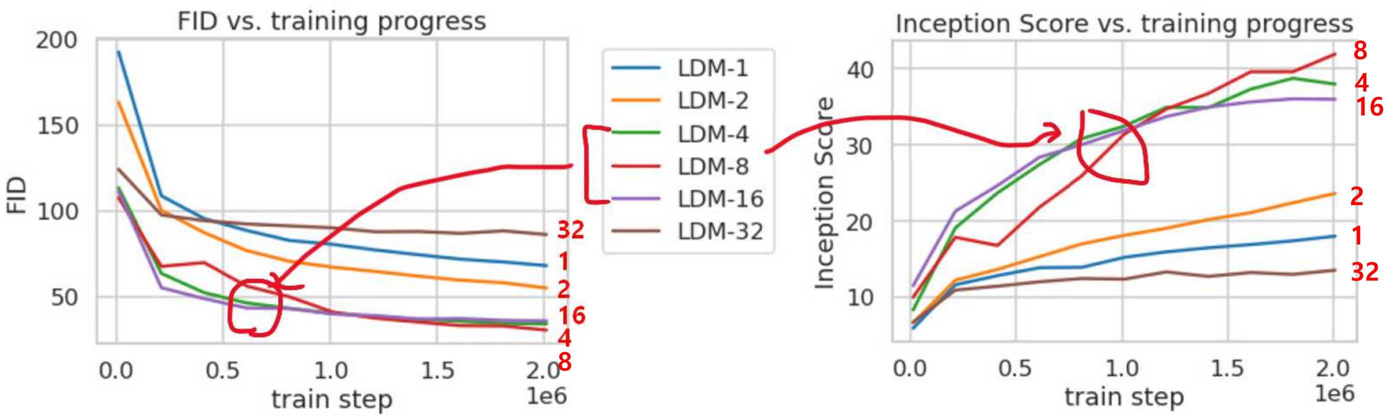


Figure 6. Analyzing the training of **class-conditional LDMs** with different downsampling factors f over $2M$ train steps on the ImageNet dataset. Pixel-based **LDM-1** requires substantially larger train times compared to models with larger downsampling factors (**LDM-[4-16]**). Too much perceptual 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 κ= 0.

- i) **Small downsampling** factors for LDM-[1,2] result in **slow training progress**, whereas ii) overly **large values of f** (LDM-32) cause **stagnating fidelity** after comparably few training steps.
- i) **Leaving most of perceptual compression to the diffusion model** and ii) **Too strong first stage compression resulting in information loss and thus limiting the achievable quality**.
- **LDM-[4-16]** strike a good balance between efficiency and perceptually faithful results, which manifests in a significant FID [29] gap of 38 between pixel-based diffusion (LDM-1) and LDM-8 after 2M training steps.

High-Resoluton Image Synthesis with Latent Diffusion Model

4. Experiments

4.1 On Perceptual Compression Tradeoffs

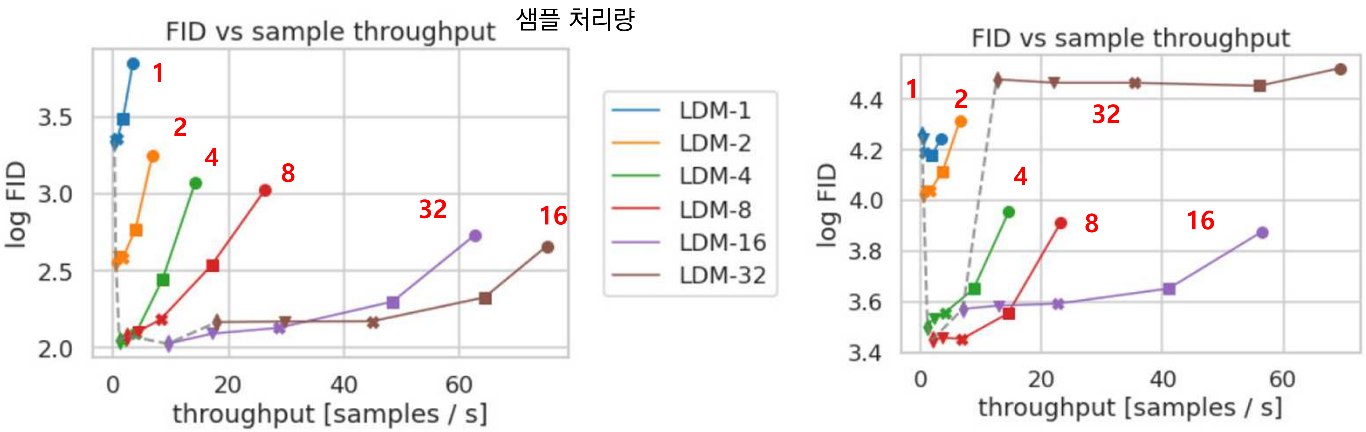


Figure 7. Comparing LDMs with varying compression on the CelebA-HQ (left) and ImageNet (right) datasets. Different markers indicate {10, 20, 50, 100, 200} sampling steps using DDIM, from right to left along each line. The dashed line shows the FID scores for 200 steps, indicating the strong performance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

- **LDM-[4-8]** outperform models with unsuitable ratios of perceptual and conceptual compression. Especially compared to pixel-based LDM-1, they achieve much lower FID scores while simultaneously significantly increasing sample throughput. **Complex datasets** such as ImageNet **require reduced compression rates to avoid reducing quality**.
- In summary, **LDM-4 and -8** offer the best conditions for achieving high-quality synthesis results.

4. Experiments

4.2 Image Generation with Latent Diffusion

- Train unconditional models of 256x256 images on CelebA-HQ [39], FFHQ [41], LSUN-Churches and -Bedrooms [102] and evaluate the i) **sample quality** and ii) their coverage of the data manifold using ii) **FID** [29] and ii) **Precision-and-Recall** [50].

CelebA-HQ 256 × 256				FFHQ 256 × 256			
Method	FID ↓	Prec. ↑	Recall ↑	**Method**	FID ↓	Prec. ↑	Recall ↑
DC-VAE [63]	15.8	-	-	ImageBART [21]	9.57	-	-
VQGAN+T. [23] (k=400)	10.2	-	-	U-Net GAN (+aug) [77]	10.9 (7.6)	-	-
PGGAN [39]	8.0	-	-	UDM [43]	5.54	-	-
LSGM [93]	7.22	-	-	StyleGAN [41]	4.16	0.71	0.46
UDM [43]	7.16	-	-	ProjectedGAN [76]	**3.08**	0.65	0.46
LDM-4 (ours, 500-s†)	**5.11**	0.72	0.49	*LDM-4* (ours, 200-s)	4.98	**0.73**	**0.50**

LSUN-Churches 256 × 256				LSUN-Bedrooms 256 × 256			
Method	FID ↓	Prec. ↑	Recall ↑	**Method**	FID ↓	Prec. ↑	Recall ↑
DDPM [30]	7.89	-	-	ImageBART [21]	5.51	-	-
ImageBART [21]	7.32	-	-	DDPM [30]	4.9	-	-
PGGAN [39]	6.42	-	-	UDM [43]	4.57	-	-
StyleGAN [41]	4.21	-	-	StyleGAN [41]	2.35	0.59	0.48
StyleGAN2 [42]	3.86	-	-	ADM [15]	1.90	**0.66**	**0.51**
ProjectedGAN [76]	**1.59**	0.61	0.44	ProjectedGAN [76]	**1.52**	0.61	0.34
*LDM-8** (ours, 200-s)	4.02	**0.64**	**0.52**	*LDM-4* (ours, 200-s)	2.95	**0.66**	0.48

Table 1. Evaluation metrics for unconditional image synthesis. CelebA-HQ results reproduced from [43,63,100], FFHQ from [42,43].
†: N-s refers to N sampling steps with the DDIM [84] sampler.
∗: trained in KL-regularized latent space

- A latent diffusion model is trained jointly together with the first stage. In contrast, we train diffusion models in a fixed space and avoid the difficulty of weighing reconstruction quality against learning the prior over the latent space.

4. Experiments

4.3 Conditional Latent Diffusion

4.3.1 Transformer Encoders for LDMs

- **Cross-attention based conditioning** into LDMs; Various conditioning modalities

➢ **Test-to-image modeling**

- Train a 1.45B parameter **KL-regularized LDM** conditioned on **language prompts** on LAION-400M [78].
- Employ the **BERT-tokenizer** [14] and implement $\boldsymbol{\tau_\theta}$ as a **transformer** [97] to **infer a latent code** which is mapped into the UNet via (multi-head) cross-attention.
- This combination of domain specific experts for **learning** a **language representation** and **visual synthesis** results in a powerful model, which generalizes well to complex, user-defined text prompts, cf . Fig. 8 and 5.
- Evaluate on MS-COCO

Text-Conditional Image Synthesis				
Method	FID ↓	IS↑	N_{params}	
CogView† [17]	27.10	18.20	4B	self-ranking, rejection rate 0.017
LAFITE† [109]	26.94	26.02	75M	
GLIDE* [59]	12.24	-	6B	277 DDIM steps, c.f.g. [32] $s = 3$
Make-A-Scene* [26]	**11.84**	-	4B	c.f.g for AR models [98] $s = 5$
LDM-KL-8	23.31	20.03±0.33	1.45B	250 DDIM steps
*LDM-KL-8-G**	12.63	**30.29**±0.42	1.45B	250 DDIM steps, c.f.g. [32] $s = 1.5$

Table 2. Evaluation of text-conditional image synthesis on the 256x256-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. †/*:Numbers from [109]/ [26]

- Applying **classifier-free diffusion guidance** [32] greatly **boosts sample quality**, such that the **guided LDM-KL-8-G** is on par with the recent state-of-the-art AR [26] and diffusion models [59] for text-to-image synthesis, while substantially reducing parameter count

High-Resoluton Image Synthesis with Latent Diffusion Model

4. Experiments

4.3 Conditional Latent Diffusion

➢ **Test-to-image modeling**

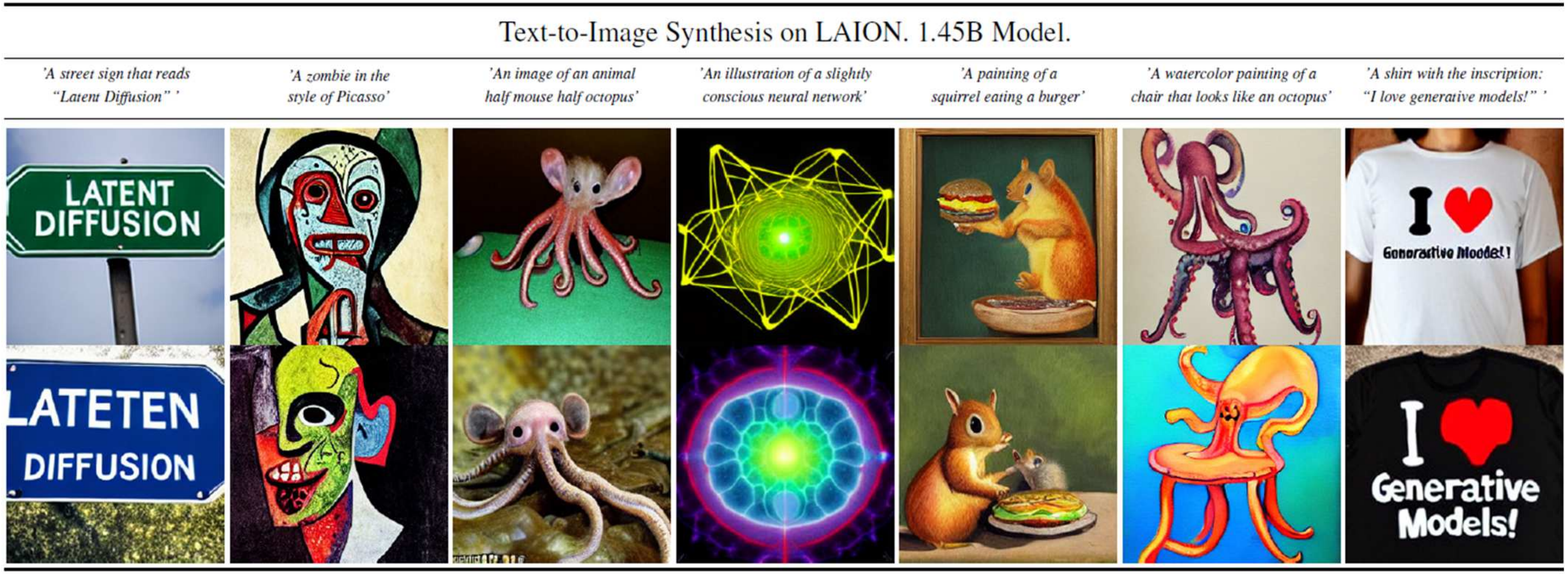


Figure 5. Samples for user-defined text prompts from our model for text-to-image synthesis, LDM-8 (KL), which was trained on the LAION [78] database. Samples generated with 200 DDIM steps and η =1.0. We use unconditional guidance [32] with s = 10.0

High-Resoluton Image Synthesis with Latent Diffusion Model

4. Experiments

4.3 Conditional Latent Diffusion

➢ **Semantic layouts**

- To further analyze **the flexibility of the cross-attention** based **conditioning mechanism**, we train models to **synthesize images based on semantic layouts on OpenImages** [49], and **finetune on COCO** [4].

Figure 8. **Layout-to-image synthesis with an LDM on COCO** [4].

➢ **Class-conditional ImageNet Model**

- Evaluate our best-performing class-conditional ImageNet models with $f = \{4,8\}$.
- Here we **outperform SOTA diffusion model ADM [15] while significantly reducing computational requirements and parameter count**, cf . Tab 18.

Table 3. **Comparison of a class-conditional ImageNet LDM with SOTA models for class-conditional image generation on ImageNet** [12].
c.f.g. denotes classifier-free guidance with a scale s as proposed in [32].

Method	FID↓	IS↑	Precision↑	Recall↑	Nparams	
BigGan-deep [3]	6.95	203.6±2.6	**0.87**	0.28	340M	-
ADM [15]	10.94	100.98	0.69	**0.63**	554M	250 DDIM steps
ADM-G [15]	4.59	186.7	0.82	0.52	608M	250 DDIM steps
LDM-4 (ours)	10.56	103.49±1.24	0.71	0.62	400M	250 DDIM steps
LDM-4-G (ours)	**3.60**	**247.67**±5.59	**0.87**	0.48	400M	250 steps, c.f.g [32], $s = 1.5$

4. Experiments

4.3 Conditional Latent Diffusion

➢ **Semantic layouts**

Figure 8. **Layout-to-image synthesis with an LDM on COCO** [4], see Sec. 4.3.1. Quantitative evaluation in the supplement D.3.

Table 3. **Comparison of a class-conditional ImageNet LDM with SOTA methods for class-conditional image generation on ImageNet** [12]. A more detailed comparison with additional baselines can be found in D.4, Tab. 10 and F. c.f.g. denotes classifier-free guidance with a scale s as proposed in [32].

Method	FID↓	IS↑	Precision↑	Recall↑	N_{params}	
BigGan-deep [3]	6.95	203.6±2.6	0.87	0.28	340M	-
ADM [15]	10.94	100.98	0.69	0.63	554M	250 DDIM steps
ADM-G [15]	4.59	186.7	0.82	0.52	608M	250 DDIM steps
LDM-4 (ours)	10.56	103.49±1.24	0.71	0.62	400M	250 DDIM steps
LDM-4-G (ours)	**3.60**	**247.67**±5.59	**0.87**	0.48	400M	250 steps, c.f.g [32], $s = 1.5$

- Evaluate our best-performing class-conditional ImageNet models with $f = \{4,8\}$ from Sec. 4.1 in Tab. 3, Fig. 4 and Sec. D.4.
- Here we **outperform SOTA diffusion model ADM [15] while significantly reducing computational requirements and parameter count**, cf . Tab 18.
- To further analyze **the flexibility of the cross-attention** based **conditioning mechanism**, we train models to **synthesize images based on semantic layouts on OpenImages** [49], and **finetune on COCO** [4], (Fig. 8. Sec. D.3) for the quantitative evaluation and implementation details.

4. Experiments

4.3 Conditional Latent Diffusion

4.3.2 Convolutional Sampling Beyond 256x256

- LDMs - General purpose image-to-image translation models by **concatenating spatially aligned conditioning information to the input of ϵ_θ**
- Train models for semantic synthesis, super-resolution (in 4.4), inpainting (in 4.5)

➢ **Semantic synthesis**

- Use images of landscapes paired with **semantic maps** [23, 61] and **concatenate downsampled versions of the semantic maps** with **the latent image representation of a f = 4 model** (VQ-reg., see Tab. 8).
- We train on an input resolution of 256x256 (crops from 384x384) but find that our model generalizes to larger resolutions and can generate images up to the megapixel regime when evaluated in a convolutional manner (see Fig. 9).

Figure 9. A LDM trained on 256x256 resolution can generalize to larger resolution (here: 512x1024) for spatially conditioned tasks such as semantic synthesis of landscape images.

4. Experiments

4.4 Super-Resolution with Latent Diffusion

- Trained for super-resolution by diretly conditioning on low-resolution images via concatenation

#1 Experiment

- Follow SR3 and fix the image degradation to a bicubic interpolation with 4x-downsampling and train on ImageNet following SR3's data processing pipeline.
- Use the f = 4 autoencoding model pretrained on OpenImages and concatenate the low-resolution conditioning y and the inputs to the UNet, i.e. ϵ_θ is the identity.

Method	FID ↓	IS ↑	PSNR ↑	SSIM ↑	N_{params}	$[\frac{\text{samples}}{s}](*)$
Image Regression [72]	15.2	121.1	**27.9**	**0.801**	625M	N/A
SR3 [72]	5.2	**180.1**	26.4	0.762	625M	N/A
LDM-4 (ours, 100 steps)	$2.8^{\dagger}/4.8^{\ddagger}$	166.3	$24.4_{\pm 3.8}$	$0.69_{\pm 0.14}$	**169M**	4.62
emphLDM-4 (ours, big, 100 steps)	$\mathbf{2.4}^{\dagger}/\mathbf{4.3}^{\ddagger}$	174.9	$24.7_{\pm 4.1}$	$0.71_{\pm 0.15}$	552M	4.5
LDM-4 (ours, 50 steps, guiding)	$4.4^{\dagger}/6.4^{\ddagger}$	153.7	$25.8_{\pm 3.7}$	$0.74_{\pm 0.12}$	184M	0.38

Table 5. x4 upscaling results on ImageNet-Val. (256x256);
†: FID features computed on validation split, ‡: FID features computed on train split;
⁎: Assessed on a NVIDIA A100

Figure 10. ImageNet 64→256 super-resolution on ImageNet-Val. **LDM-SR has advantages at rendering realistic textures** but **SR3** can synthesize **more coherent fine structures**. See appendix for additional samples and cropouts.

4. Experiments

4.4 Super-Resolution with Latent Diffusion

#2 Experiment

- Conduct a user study comparing **the pixel-baseline with LDM-SR**.
- We follow SR3 [72] where **human subjects** were shown a low-res image in between two high-res images and asked for preference.

	SR on ImageNet		Inpainting on Places	
User Study	Pixel-DM ($f1$)	*LDM-4*	LAMA [88]	*LDM-4*
Task 1: Preference vs GT ↑	16.0%	**30.4%**	13.6%	**21.0%**
Task 2: Preference Score ↑	29.4%	**70.6%**	31.9%	**68.1%**

Table 4. Task 1: Subjects were shown ground truth and generated image and asked for preference. Task 2: Subjects had to decide between two generated images.

✓ PSNR and SSIM can be pushed by using a post-hoc guiding mechanism [15] and we implement this *image-based guider* via *a perceptual loss*.

4. Experiments

4.5 Inpainting with Latent Diffusion

- *Inpainting* : Task of filling masked regions of an image with new content either because parts of the image are corrupted or to replace existing but undesired content within the image.

Model (*reg.*-type)	train throughput samples/sec.	sampling throughput† @256	sampling throughput† @512	train+val hours/epoch	FID@2k epoch 6
LDM-1 (no first stage)	0.11	0.26	0.07	20.66	24.74
LDM-4 (*KL*, w/ attn)	0.32	0.97	0.34	7.66	15.21
LDM-4 (*VQ*, w/ attn)	0.33	0.97	0.34	7.04	14.99
LDM-4 (*VQ*, w/o attn)	0.35	0.99	0.36	6.66	15.95

Table 6. Assessing inpainting efficiency. †: Deviations from Fig. 7 due to varying GPU settings/batch sizes cf . the supplement

- A speed-up of at least 2.7x between pixel- (LDM-1) and latent-based diffusion models (LDM-4) while improving FID scores by a factor of at least 1.6x

Method	40-50% masked FID ↓	40-50% masked LPIPS ↓	All samples FID ↓	All samples LPIPS ↓
LDM-4 (ours, big, w/ ft)	**9.39**	0.246± 0.042	**1.50**	0.137± 0.080
LDM-4 (ours, big, w/o ft)	12.89	0.257± 0.047	2.40	0.142± 0.085
LDM-4 (ours, w/ attn)	11.87	0.257± 0.042	2.15	0.144± 0.084
LDM-4 (ours, w/o attn)	12.60	0.259± 0.041	2.37	0.145± 0.084
LaMa [88]†	12.31	**0.243**± 0.038	2.23	**0.134**± 0.080
LaMa [88]	12.0	**0.24**	2.21	0.14
CoModGAN [107]	10.4	0.26	1.82	0.15
RegionWise [52]	21.3	0.27	4.75	0.15
DeepFill v2 [104]	22.1	0.28	5.20	0.16
EdgeConnect [58]	30.5	0.28	8.37	0.16

Table 7. Comparison of inpainting performance on 30k crops of size 512x512 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.
† recomputed on our test set, since the original test set used in [88] was not available

4. Experiments

4.5 Inpainting with Latent Diffusion

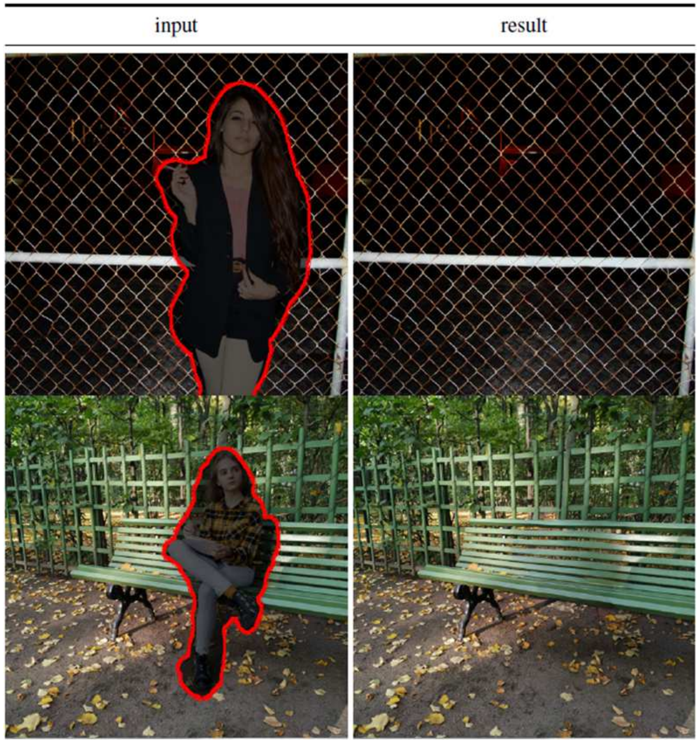


Figure 11. Qualitative results on object removal with our big, w/ft inpainting model.

5. Limitations & Societal Impact

5.1 Limitations (LDMs)

- Significantly **reduce computational** requirements compared to pixel-based approaches, their **sequential sampling process** is **still slower** than that of GANs.
- Can be **questionable** when **high precision** is required: although the loss of image quality is very small in our f = 4 autoencoding models (see Fig. 1), their **reconstruction capability** can become a **bottleneck** for tasks that require fine-grained accuracy in pixel space.

5.2 Societal Impact

- Easier to create and disseminate manipulated data or spread misinformation and spam; "**deepfake**"
- Training data including sensitive or personal information
- Deep learning modules tend to reproduce or exacerbate biases that are already present in the data [22, 38, 91].
- While diffusion models achieve better coverage of the data distribution than e.g. GAN-based approaches, the extent to which our **two-stage approach** that combines **adversarial training** and **a likelihood-based objective misrepresents the data** remains an important research question.

Appendix

B. Detailed Information on Denoising Diffusion Models

$$L_{DM} = \mathbb{E}_{x,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(x_t,t)\|_2^2\right] \quad (1)$$

$$L_{LDM} := \mathbb{E}_{\mathcal{E}(x),\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t,t)\|_2^2\right] \quad (2)$$

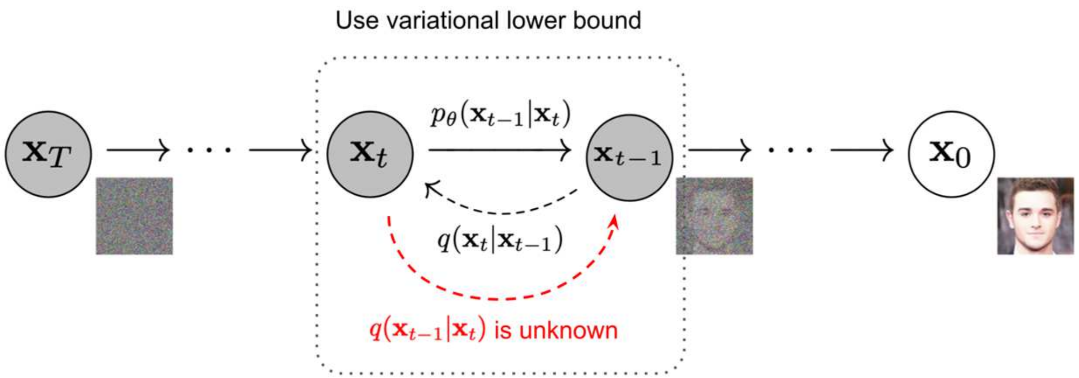
Latent Vector

Latent State of the Denoising U-Net at time 't'

Domain Specific Encoder (i.e. Transformer) on a condition 'y'

Conditional Loss Function

$$L_{LDM} := \mathbb{E}_{\mathcal{E}(x),y,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t,t,\tau_\theta(y))\|_2^2\right] \quad (3)$$



Diffusion models

- It can be specified in terms of a signal-to-noise ratio $SNR(t) = \alpha_t^2/\sigma_t^2$ consisting of sequences $(\alpha_t)_{t=1}^T$ and $(\sigma_t)_{t=1}^T$ which, starting from a data sample x_0, define a forward diffusion process q as

$$q(x_t|x_0) = \mathcal{N}(x_t|\alpha_t x_0, \sigma_t^2\mathbb{I})$$

with Markov structure for $s < t$

$$q(x_t|x_s) = \mathcal{N}(x_t|\alpha_{t|s}x_s, \sigma_{t|s}^2\mathbb{I})$$

$$\alpha_{t|s} = \frac{\alpha_t}{\alpha_s} \qquad \sigma_{t|s}^2 = \sigma_t^2 - \alpha_{t|s}^2\sigma_s^2$$

Denoising Diffusion models

- Generative models $p(x_0)$ which revert a forward diffusion process with a similar Markov structure

$$p(x_0) = \int_z p(x_T)\prod_{t=1}^{T} p(x_{t-1}|x_t)$$

Appendix

B. Detailed Information on Denoising Diffusion Models

[30] Jonathan Ho, Ajay Jain, and Pieter Abbeel. **Denoising diffusion probabilistic models**. In NeurIPS, 2020
[45] Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. **Variational diffusion models**. CoRR, abs/2107.00630, 2021

➢ **Denoising Diffusion models**

- The evidence lower bound (ELBO) associated with this model then decomposes over the discrete time steps

$$-\log p(x_0) \leq \underline{\mathbb{KL}(q(x_T|x_0)|p(x_T))} + \underline{\sum_{t=1}^{T} \mathbb{E}_{q(x_t|x_0)} \mathbb{KL}(q(x_{t-1}|x_t, x_0)|p(x_{t-1}|x_t))}$$

$$p(x_0) = \int_z p(x_T) \prod_{t=1}^{T} p(x_{t-1}|x_t)$$

✓ The prior $p(x_T)$ is typically choosen as a standard normal distribution and the first term of the ELBO then depends only on the final signal-to-noise ratio $SNR(t)$

✓ To minimize the remaining terms, a common choice to parameterize $p(x_{t-1}|x_t)$ is to specify it in terms of the true posterior $q(x_{t-1}|x_t, x_0)$ but with the unknown x_0 replaced by an estimate $x_\theta(x_t, t)$ based on the current step x_t.

$$p(x_{t-1}|x_t) := q(x_{t-1}|x_t, x_\theta(x_t, t)) \quad [45]$$

$$= \mathcal{N}(x_{t-1}|\mu_\theta(x_t, t), \sigma^2_{t|t-1}\frac{\sigma^2_{t-1}}{\sigma^2_t}\mathbb{I})$$

$$\mu_\theta(x_t, t) = \frac{\alpha_{t|t-1}\sigma^2_{t-1}}{\sigma^2_t}x_t + \frac{\alpha_{t-1}\sigma^2_{t|t-1}}{\sigma^2_t}x_\theta(x_t, t)$$

Simplify to

$$\sum_{t=1}^{T} \mathbb{E}_{\mathcal{N}(\epsilon|0,\mathbb{I})}\frac{1}{2}(\mathrm{SNR}(t-1) - \mathrm{SNR}(t))\|x_0 - x_\theta(\alpha_t x_0 + \sigma_t \epsilon, t)\|^2$$

Use the reparametrization [30]

$$\epsilon_\theta(x_t, t) = (x_t - \alpha_t x_\theta(x_t, t))/\sigma_t$$

to express the reconstruction term as a denoising objective

$$\|x_0 - x_\theta(\alpha_t x_0 + \sigma_t \epsilon, t)\|^2 = \frac{\sigma_t^2}{\alpha_t^2}\|\epsilon - \epsilon_\theta(\alpha_t x_0 + \sigma_t \epsilon, t)\|^2$$

and the reweighting, which assign each of the terms the same weight

$$L_{DM} = \mathbb{E}_{x,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|_2^2\right]$$

Appendix

C. Image Guiding Mechanisms

[15] Prafulla Dhariwal and Alex Nichol. **Diffusion models beat GANs on image synthesis**. CoRR, abs/2105.05233, 2021

- Intriguing feature of **diffusion models** is that **unconditional models can be conditioned at test-time**

- ***Image-guiding*** : Guide both unconditional and conditional models trained on the ImageNet dataset with a classifier $\log p_\Phi(y|x_t)$, trained on each x_t of the diffusion process

- For an epsilon-parameterized model with fixed variance, the guiding algorithm as introduced in [15]

$$\hat{\epsilon} \leftarrow \epsilon_\theta(z_t, t) + \sqrt{1-\alpha_t^2}\, \nabla_{z_t} \log p_\Phi(y|z_t)$$

✓ Interpret as an update correcting the "score" ϵ_θ with a conditional distribution $\log p_\Phi(y|z_t)$

✓ So far, this is only applied to a *single-class classification model*

- We re-interpret the *guiding distribution* $p_\Phi(y|T(D(z_0(z_t)))$ as a general purpose image-to-image translation task given a target image y,

- T can be differentiable transformation adopted to the image-to-image translation task, such as the identity, a *downsampling* or similar.

- Ex; We can assume a Gaussian guider with fixed variance σ^2, such that L_2 regression objective

$$\log p_\Phi(y|z_t) = -\frac{1}{2}\|y - T(\mathcal{D}(z_0(z_t)))\|_2^2$$

Appendix

C. Image Guiding Mechanisms

$$\log p_{\Phi}(y|z_t) = -\frac{1}{2}\|y - T(\mathcal{D}(z_0(z_t)))\|_2^2$$

- Fig. 14 demonstrates how this formulation can serve as an upsampling mechanism of an unconditional model trained on 256^2 images, where unconditional samples of size 256^2 guide the convolutional synthesis of 512^2 images and T is a 2x bicubic downsampling.

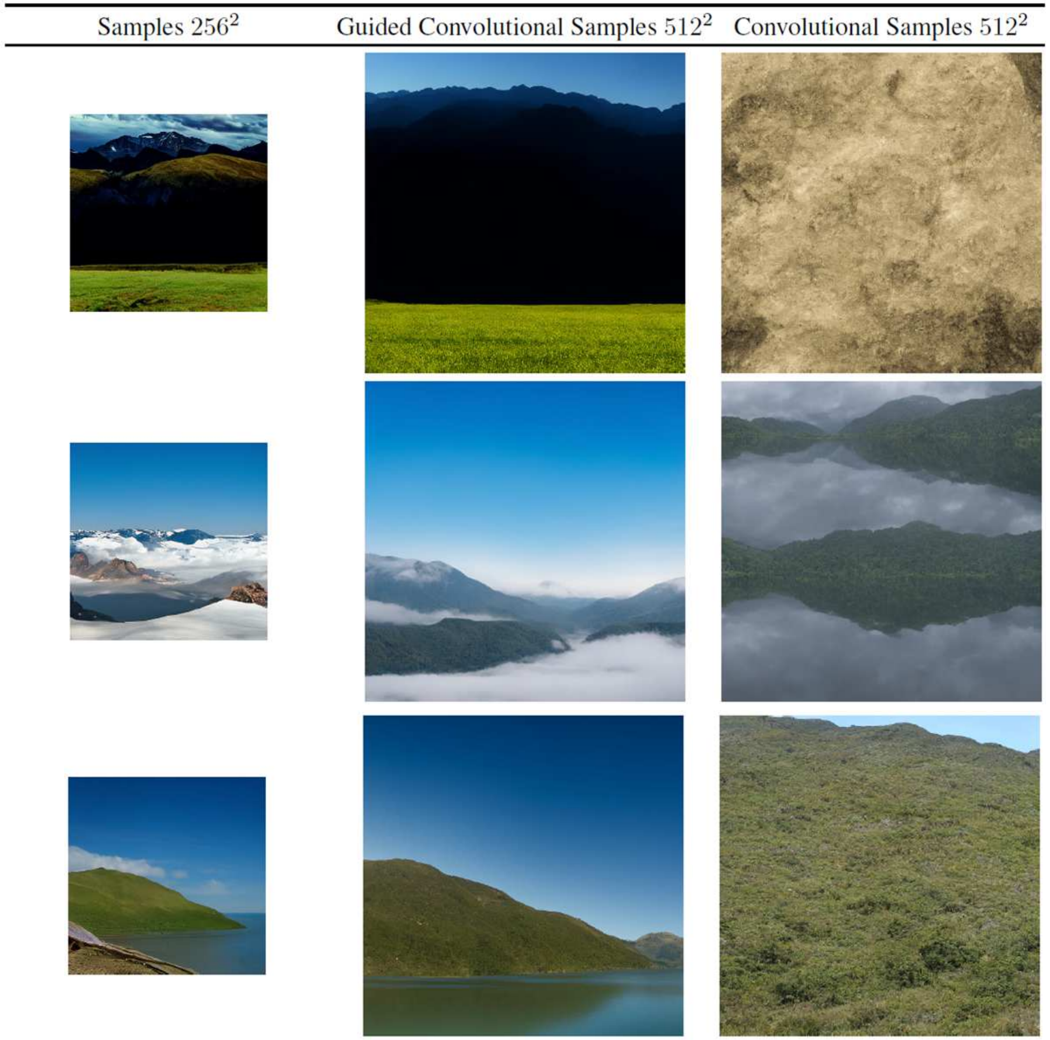


Figure 14. On landscapes, **convolutional sampling with unconditional models can lead to homogeneous and incoherent global structures** (see column 2). L_2 **-guiding with a low resolution image can help to reestablish coherent global structures**.

Appendix

D. Additional Results

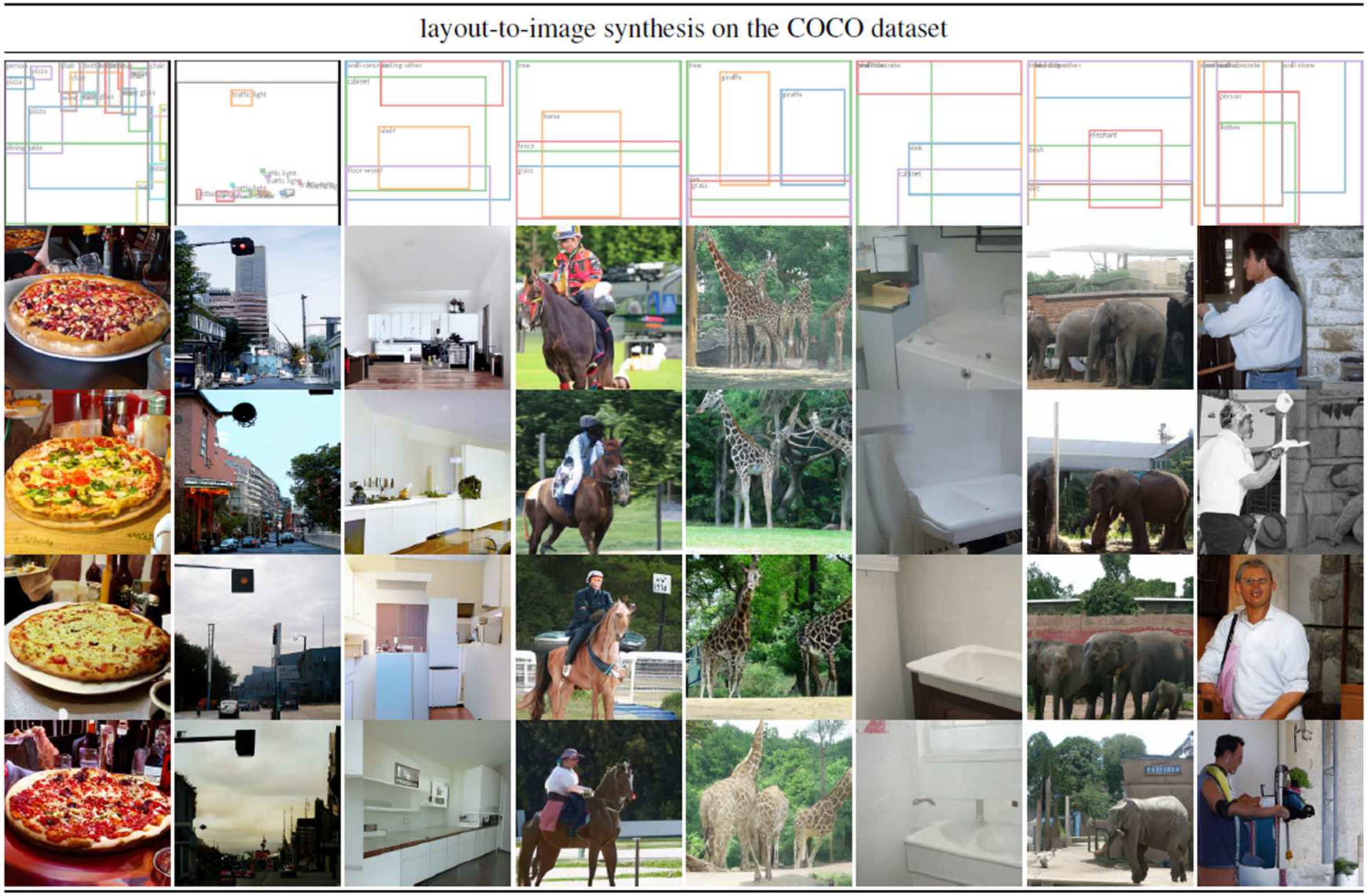


Figure 16. More samples from our best model for layout-to-image synthesis, LDM-4, which was trained on the *OpenImages* dataset and finetuned on the *COCO* dataset. Samples generated with 100 DDIM steps and $\eta = 0$. Layouts are from the COCO validation set.

Appendix

D. Additional Results

LDM-BSR: General Purpose SR Model via Diverse Image Degradation

Figure 18. LDM-BSR(*Blind Super Resolution*) generalizes to arbitrary inputs and can be used as a general-purpose upsampler, upscaling samples from a class-conditional LDM (image cf . Fig. 4) to 1024^2 resolution. In contrast, using a fixed degradation process (see Sec. 4.4) hinders generalization.

Appendix

E. Implementation Details and Hyperparameters

E.1. Hyperparmeters

Table 12. Hyperparameters for the **unconditional LDMs** producing the numbers shown in Tab. 1. All models trained on a single NVIDIA A100.

	CelebA-HQ 256×256	FFHQ 256×256	LSUN-Churches 256×256	LSUN-Bedrooms 256×256
f	4	4	8	4
z-shape	$64 \times 64 \times 3$	$64 \times 64 \times 3$	-	$64 \times 64 \times 3$
$\lvert\mathcal{Z}\rvert$	8192	8192	-	8192
Diffusion steps	1000	1000	1000	1000
Noise Schedule	linear	linear	linear	linear
N_{params}	274M	274M	294M	274M
Channels	224	224	192	224
Depth	2	2	2	2
Channel Multiplier	1,2,3,4	1,2,3,4	1,2,2,4,4	1,2,3,4
Attention resolutions	32, 16, 8	32, 16, 8	32, 16, 8, 4	32, 16, 8
Head Channels	32	32	24	32
Batch Size	48	42	96	48
Iterations*	410k	635k	500k	1.9M
Learning Rate	9.6e-5	8.4e-5	5.e-5	9.6e-5

Table 13. Hyperparameters for the **conditional LDMs** trained on the ImageNet dataset for the analysis in Sec. 4.1. All models trained on a single NVIDIA A100.

	LDM-1	*LDM-2*	*LDM-4*	*LDM-8*	*LDM-16*	*LDM-32*
z-shape	$256 \times 256 \times 3$	$128 \times 128 \times 2$	$64 \times 64 \times 3$	$32 \times 32 \times 4$	$16 \times 16 \times 8$	$88 \times 8 \times 32$
$\lvert\mathcal{Z}\rvert$	-	2048	8192	16384	16384	16384
Diffusion steps	1000	1000	1000	1000	1000	1000
Noise Schedule	linear	linear	linear	linear	linear	linear
Model Size	396M	391M	391M	395M	395M	395M
Channels	192	192	192	256	256	256
Depth	2	2	2	2	2	2
Channel Multiplier	1,1,2,2,4,4	1,2,2,4,4	1,2,3,5	1,2,4	1,2,4	1,2,4
Number of Heads	1	1	1	1	1	1
Batch Size	7	9	40	64	112	112
Iterations	2M	2M	2M	2M	2M	2M
Learning Rate	4.9e-5	6.3e-5	8e-5	6.4e-5	4.5e-5	4.5e-5
Conditioning	CA	CA	CA	CA	CA	CA
CA-resolutions	32, 16, 8	32, 16, 8	32, 16, 8	32, 16, 8	16, 8, 4	8, 4, 2
Embedding Dimension	512	512	512	512	512	512
Transformers Depth	1	1	1	1	1	1

E. Implementation Details and Hyperparameters

E.2.1 Implementations of τ_θ for conditional LDMs

- For the experiments on text-to-image and layout-to-image (Sec. 4.3.1) synthesis, we implement **the conditioner τ_θ** as an **unmasked transformer** which processes a tokenized version of the input y and produces an output $\zeta := \tau_\theta(y)$, where $\zeta \in \mathbb{R}^{M \times d_\tau}$.
- **Transformer** is implemented from ***N*** **transformer blocks** consisting of **global self-attention layers**, **layer-normalization** and **position-wise MLPs** as follows (https://github.com/lucidrains/x-transformers)

$$
\begin{aligned}
&\zeta \leftarrow \mathrm{TokEmb}(y) + \mathrm{PosEmb}(\mathrm{y}) \\
&\text{for } i = 1, \ldots, N: \\
&\quad \zeta_1 \leftarrow \mathrm{LayerNorm}(\zeta) \\
&\quad \zeta_2 \leftarrow \mathrm{MultiHeadSelfAttention}(\zeta_1) + \zeta \\
&\quad \zeta_3 \leftarrow \mathrm{LayerNorm}(\zeta_2) \\
&\quad \zeta \leftarrow \mathrm{MLP}(\zeta_3) + \zeta_2 \\
&\zeta \leftarrow \mathrm{LayerNorm}(\zeta)
\end{aligned}
$$

- With ξ available, the conditioning is mapped into the UNet via the cross-attention mechanism as depicted in Fig. 3.
- We modify the "*ablated UNet*" [15] architecture and replace **the self-attention layer** with **a shallow (unmasked) transformer** consisting of *T* blocks with alternating layers of **(i) self-attention, (ii) a position-wise MLP and (iii) a cross-attention layer**;

input	$\mathbb{R}^{h \times w \times c}$
LayerNorm	$\mathbb{R}^{h \times w \times c}$
Conv1x1	$\mathbb{R}^{h \times w \times d \cdot n_h}$
Reshape	$\mathbb{R}^{h \cdot w \times d \cdot n_h}$
$\times T$ SelfAttention	$\mathbb{R}^{h \cdot w \times d \cdot n_h}$
$\times T$ MLP	$\mathbb{R}^{h \cdot w \times d \cdot n_h}$
$\times T$ CrossAttention	$\mathbb{R}^{h \cdot w \times d \cdot n_h}$
Reshape	$\mathbb{R}^{h \times w \times d \cdot n_h}$
Conv1x1	$\mathbb{R}^{h \times w \times c}$

Table 16. Architecture of a transformer block as described in Sec. E.2.1, replacing the self-attention layer of the standard "ablated UNet" architecture [15]. Here, n_h denotes the number of attention heads and d the dimensionality per head.

Appendix

E. Implementation Details and Hyperparameters

E.2.1 Implementations of τ_θ for conditional LDMs

- For the **text-to-image model**, we rely on a publicly available ***tokenizer*** [99]. (https://huggingface.co/transformers/model_doc/bert.html#berttokenizerfast)
- The **layout-to-image model** discretizes the spatial locations of the bounding boxes and encodes each box as a (l, b, c) -tuple. where l denotes the (discrete) top-left and b the bottom-right position. Class information is contained in c.
- See Tab. 17 for the hyperparameters of τ_θ and Tab. 13 for those of the UNet for both of the above tasks.
- Note that the **class-conditional model** as described in Sec. 4.1 is also implemented via **cross-attention**, where τ_θ is a single learnable embedding layer with a dimensionality of 512, mapping classes y to $\zeta \in \mathbb{R}^{1\times 512}$.

	Text-to-Image	**Layout-to-Image**
seq-length	77	92
depth N	32	16
dim	1280	512

Table 17. Hyperparameters for the experiments with transformer encoders in Sec. 4.3

Appendix

F. Computational Requirements

Method	Generator Compute	Classifier Compute	Overall Compute	Inference Throughput*	N_{params}	FID↓	IS↑	Precision↑	Recall↑
LSUN Churches 256^2									
StyleGAN2 [42]†	64	-	64	-	59M	3.86	-	-	-
LDM-8 (ours, 100 steps, 410K)	18	-	18	6.80	256M	4.02	-	0.64	0.52
LSUN Bedrooms 256^2									
ADM [15]† (1000 steps)	232	-	232	0.03	552M	1.9	-	0.66	0.51
LDM-4 (ours, 200 steps, 1.9M)	60	-	55	1.07	274M	2.95	-	0.66	0.48
CelebA-HQ 256^2									
LDM-4 (ours, 500 steps, 410K)	14.4	-	14.4	0.43	274M	5.11	-	0.72	0.49
FFHQ 256^2									
StyleGAN2 [42]	32.13‡	-	32.13†	-	59M	3.8	-	-	-
LDM-4 (ours, 200 steps, 635K)	26	-	26	1.07	274M	4.98	-	0.73	0.50
ImageNet 256^2									
VQGAN-f-4 (ours, first stage)	29	-	29	-	55M	0.58††	-	-	-
VQGAN-f-8 (ours, first stage)	66	-	66	-	68M	1.14††	-	-	-
BigGAN-deep [3]†	128-256		128-256	-	340M	6.95	$203.6_{\pm 2.6}$	0.87	0.28
ADM [15] (250 steps) †	916	-	916	0.12	554M	10.94	100.98	0.69	0.63
ADM-G [15] (25 steps) †	916	46	962	0.7	608M	5.58	-	0.81	0.49
ADM-G [15] (250 steps)†	916	46	962	0.07	608M	4.59	186.7	0.82	0.52
ADM-G,ADM-U [15] (250 steps)†	329	30	349	n/a	n/a	3.85	221.72	0.84	0.53
LDM-8-G (ours, 100, 2.9M)	79	12	91	1.93	506M	8.11	$190.4_{\pm 2.6}$	0.83	0.36
LDM-8 (ours, 200 ddim steps 2.9M, batch size 64)	79	-	79	1.9	395M	17.41	72.92	0.65	0.62
LDM-4 (ours, 250 ddim steps 178K, batch size 1200)	271	-	271	0.7	400M	10.56	$103.49_{\pm 1.24}$	0.71	0.62
LDM-4-G (ours, 250 ddim steps 178K, batch size 1200, classifier-free guidance [32] scale 1.25)	271	-	271	0.4	400M	3.95	$178.22_{\pm 2.43}$	0.81	0.55
LDM-4-G (ours, 250 ddim steps 178K, batch size 1200, classifier-free guidance [32] scale 1.5)	271	-	271	0.4	400M	3.60	$247.67_{\pm 5.59}$	0.87	0.48

Table 18. Comparing compute requirements during training and inference throughput with state-of-the-art generative models. Compute during training in V100-days, numbers of competing methods taken from [15] unless stated differently; ∗ : Throughput measured in samples/ sec on a single NVIDIA A100; † : Numbers taken from [15]; ‡ : Assumed to be trained on 25M train examples; ††: R-FID vs. ImageNet validation set

Appendix

G. Details on Autoencoder Models

- We **train** all our **autoencoder** models in an ***adversarial manner*** following [23], such that a *patch-based discriminator* D_ψ is optimized to differentiate original images from reconstructions $D(\mathcal{E}(x))$.

- **To avoid arbitrarily scaled latent spaces**, we **regularize the latent z to be zero centered and obtain small variance** by introducing a regularizing loss term L_{reg}.

- Two different regularization methods

  ① A **low-weighted Kullback-Leibler-term** between $q_{\mathcal{E}}(z|x) = N(z; \mathcal{E}_\mu, \mathcal{E}_{\sigma^2})$ and a standard normal distribution $N(z; 0, 1)$ as in a standard variational autoencoder [46, 69]

  ② Regularizing **the latent space with a vector quantization layer** by learning a codebook of $|Z|$ different exemplars [96].

- To obtain **high-fidelity** reconstructions, we only use a very **small regularization** for both scenarios, i.e. we either weight the KL term by a factor $\sim 10^{-6}$ or choose a high codebook dimensionality $|Z|$.

- The full objective to train the autoencoding model $(\mathcal{E}, D)$

$$L_{\text{Autoencoder}} = \min_{\mathcal{E},\mathcal{D}} \max_{\psi} \Big(L_{rec}(x, \mathcal{D}(\mathcal{E}(x))) - L_{adv}(\mathcal{D}(\mathcal{E}(x))) + \log D_\psi(x) + L_{reg}(x; \mathcal{E}, \mathcal{D}) \Big)$$

Appendix

G. Details on Autoencoder Models

➢ DM Training in Latent Space

- Note that for training diffusion models on the learned latent space, we again distinguish two cases when learning $p(z)$ or $p(z|y)$ (Sec. 4.3):

[*For a **KL-regularized latent space***]

- We sample $z = \mathcal{E}_\mu(x) + \epsilon \mathcal{E}_\sigma =: \mathcal{E}(x)$, where $\epsilon = N(0,1)$.
- When rescaling the latent, we estimate the component-wise variance from the first batch in the data

$$\hat{\sigma}^2 = \frac{1}{bchw} \sum_{b,c,h,w} (z^{b,c,h,w} - \hat{\mu})^2 \quad \hat{\mu} = \frac{1}{bchw} \sum_{b,c,h,w} z^{b,c,h,w}$$

- The output of $\mathcal{E}$ is scaled such that the rescaled latent has unit standard deviation; i.e.:

$$z \leftarrow \frac{z}{\hat{\sigma}} = \frac{\mathcal{E}(x)}{\hat{\sigma}}$$

[*For a **VQ-regularized latent space***]

- We extract z before the quantization layer and absorb the quantization operation into the decoder, i.e. it can be interpreted as the first layer of D.