Text to Image Generation - Stable Diffusion and Imagen

Apoorv Garg

Diffusion Models

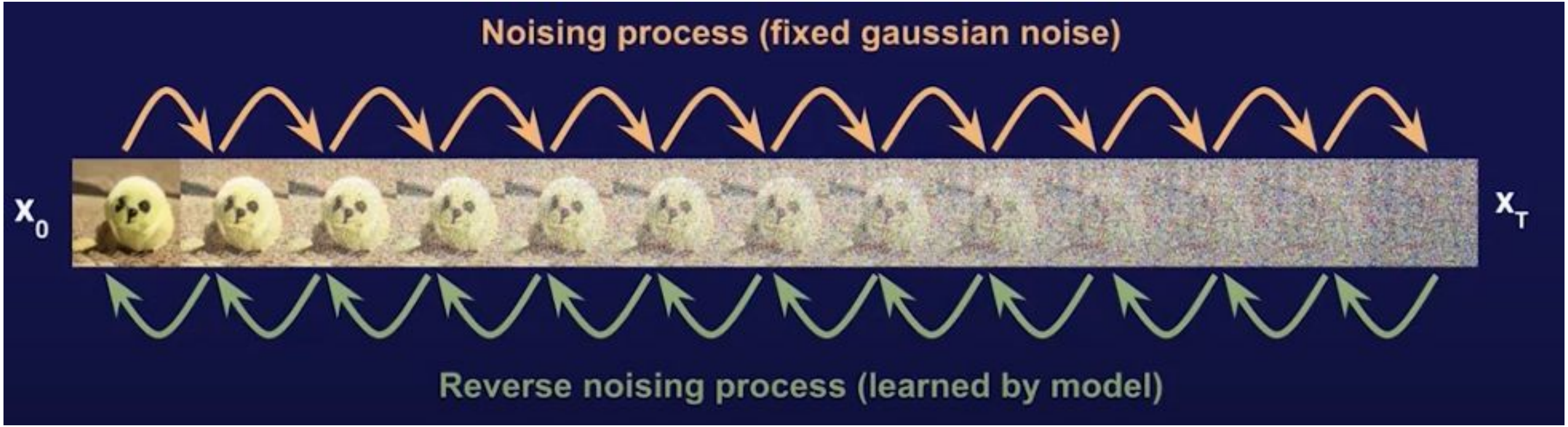

Noising process (fixed gaussian noise)
x_0
x_T
Reverse noising process (learned by model)

Figure from Prafulla Dhariwal’s Lecture

Diffusion Models

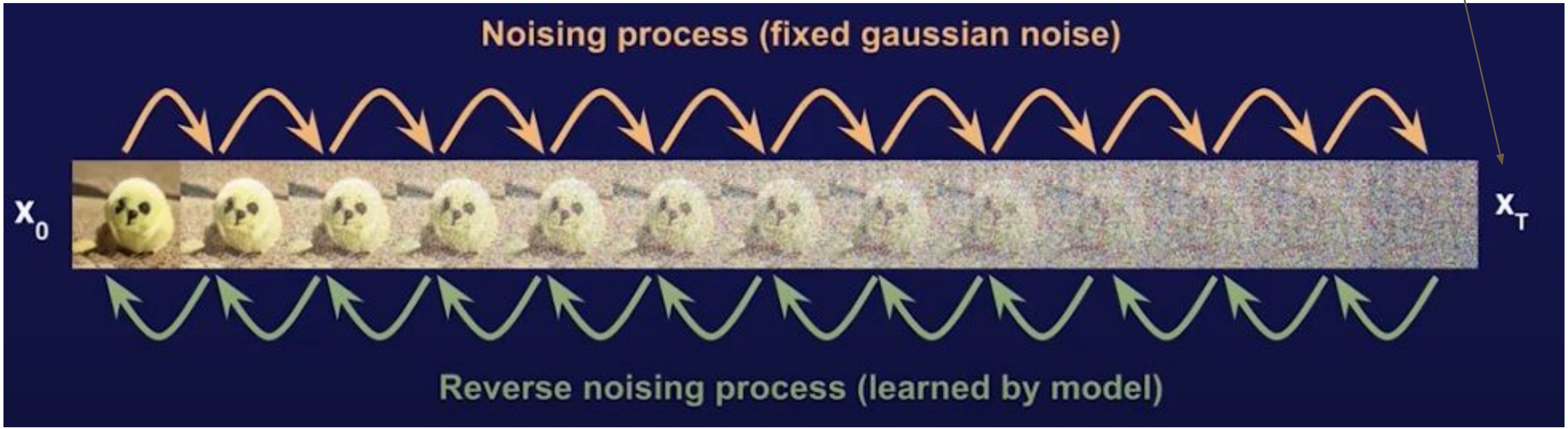

Pure Noise
Noising process (fixed gaussian noise)
x_0
x_T
Reverse noising process (learned by model)

Figure from Prafulla Dhariwal's Lecture

DDPM

We want to learn the reverse Process conditioned on the forwarded process. **For very small addition in noise in forward process, the reverse prediction is similar (also gaussian), our model predicts the mean of this reverse process**

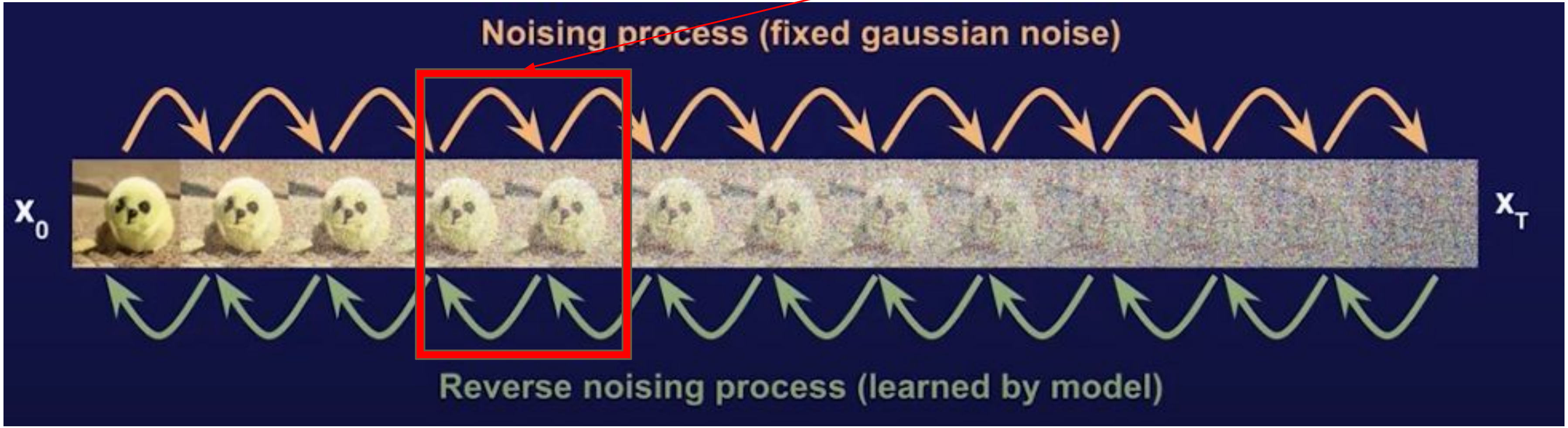

Noising process (fixed gaussian noise)
x_0
x_T
Reverse noising process (learned by model)

Figure from Prafulla Dhariwal's Lecture at MIT

DDPM

We can directly jump to any time t directly from xo in forward process

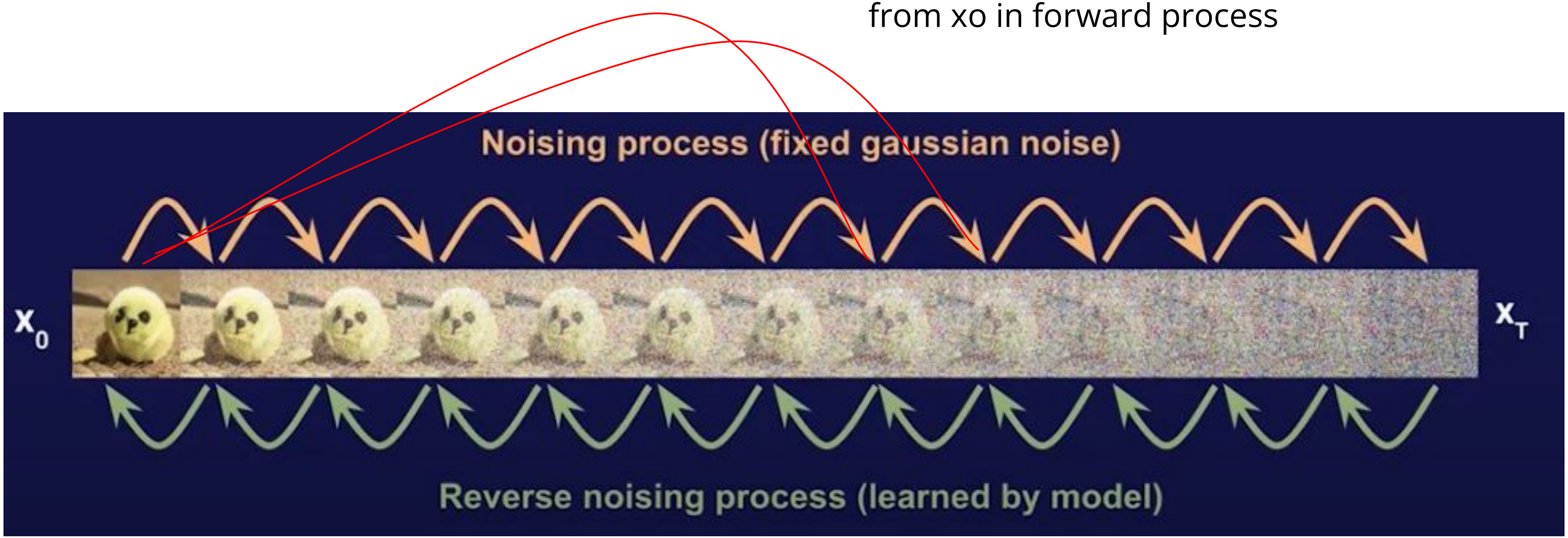
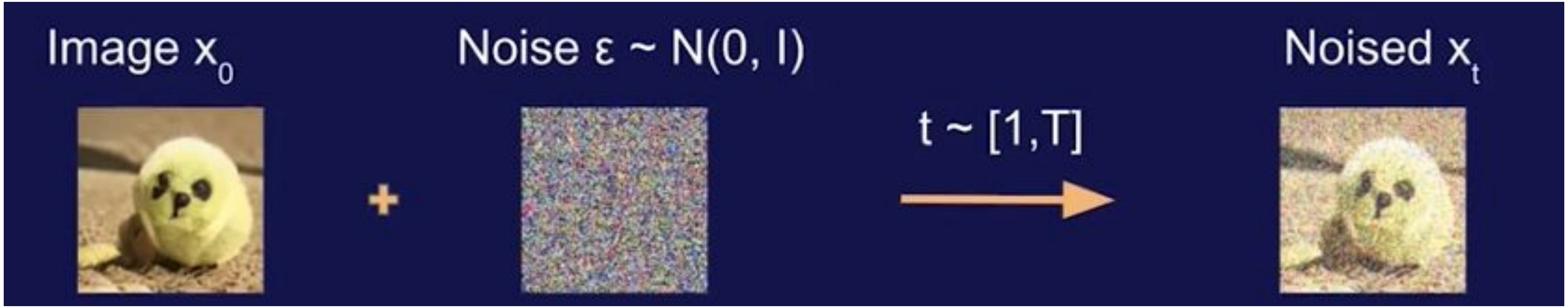
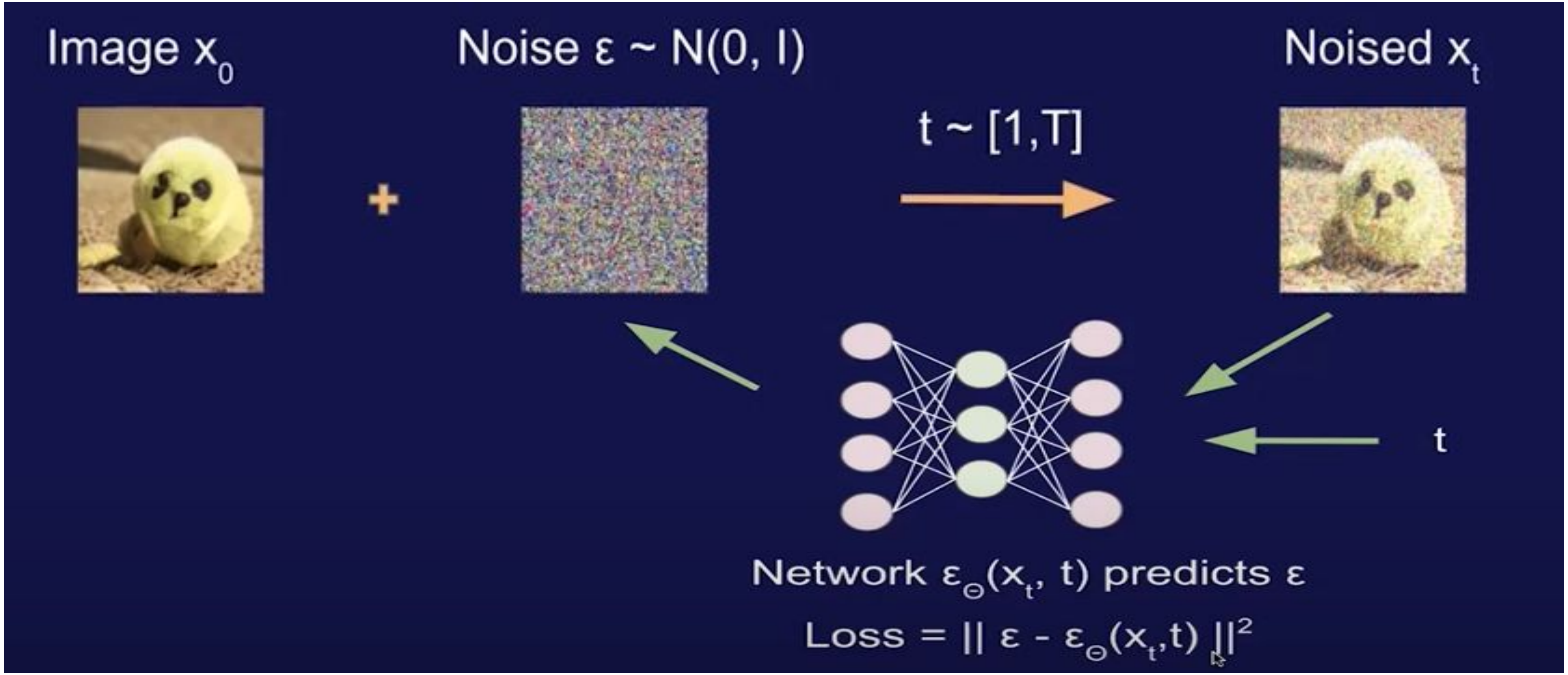

Noising process (fixed gaussian noise)
x_0
x_T
Reverse noising process (learned by model)

Figure from Prafulla Dhariwal’s Lecture at MIT

DDPM simplified into training and test terms



DDPM simplified into training and inference cycles



Network $\varepsilon_\theta(x_t, t)$ predicts ε

Loss = $|| \varepsilon - \varepsilon_\theta(x_t,t) ||^2$

DDPM simplified into training and inference cycles

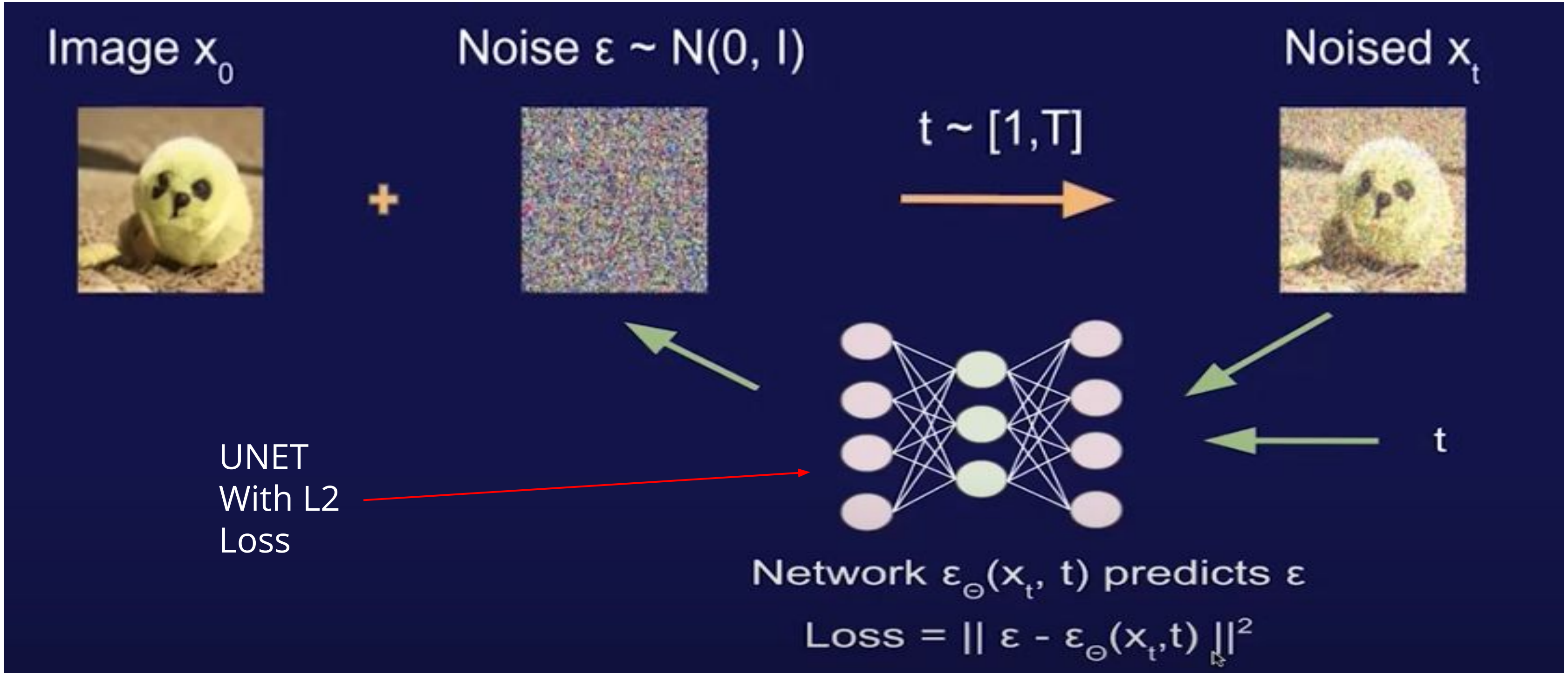
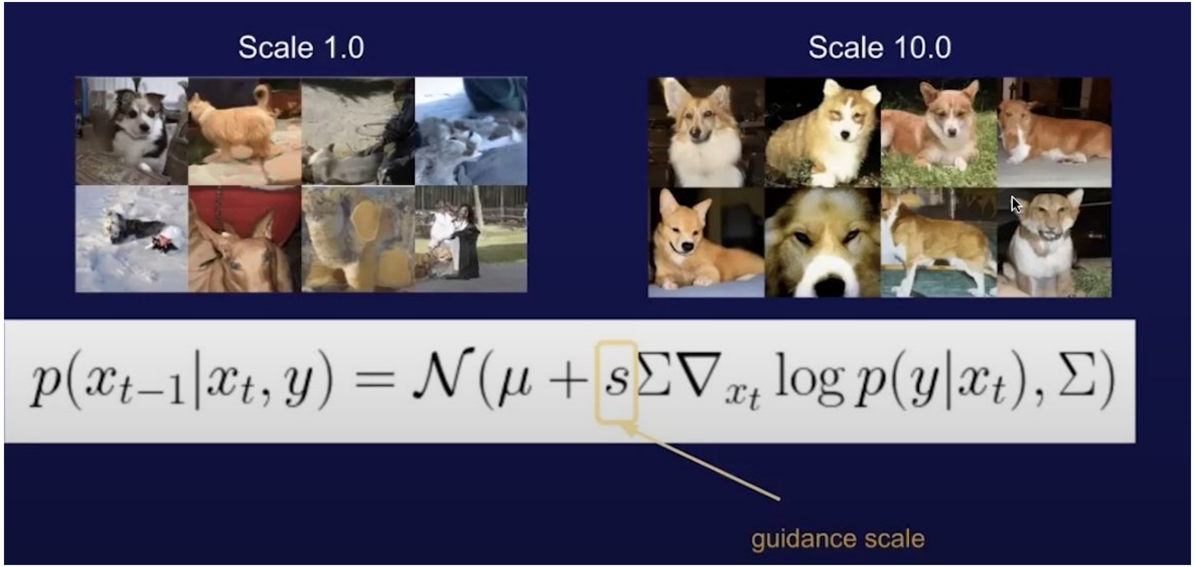


Image x_0
+
Noise $\varepsilon \sim N(0, I)$
$t \sim [1,T]$
Noised x_t
t
UNET
With L2
Loss
Network $\varepsilon_\Theta(x_t, t)$ predicts ε
Loss = $|| \varepsilon - \varepsilon_\Theta(x_t,t) ||^2$

Class Guided Conditional Generation

- We take the noisy step(x_t) generated by the UNET, run it through a classifier and add the gradient of the classifier out wrt x_t and add a scaled version of that to the generation step.
- Hence we “Guide” the model towards correctness



Scale 1.0

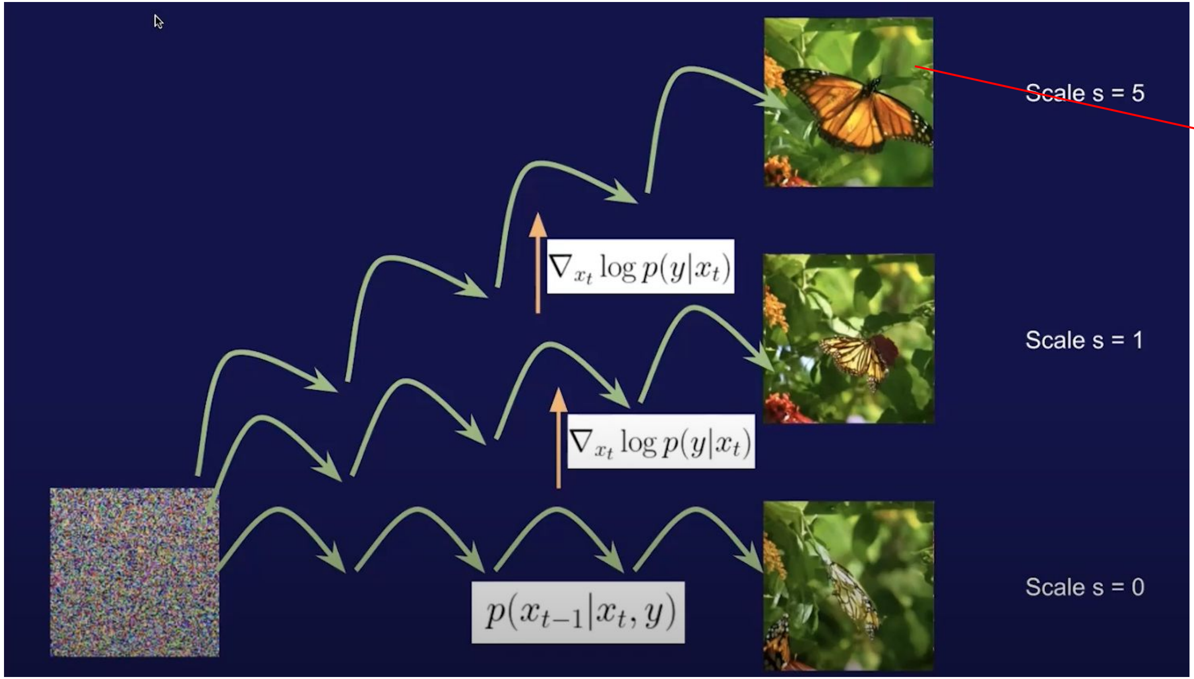
Scale 10.0

$$p(x_{t-1}|x_t, y) = \mathcal{N}(\mu + \boxed{s}\Sigma\nabla_{x_t}\log p(y|x_t), \Sigma)$$

guidance scale

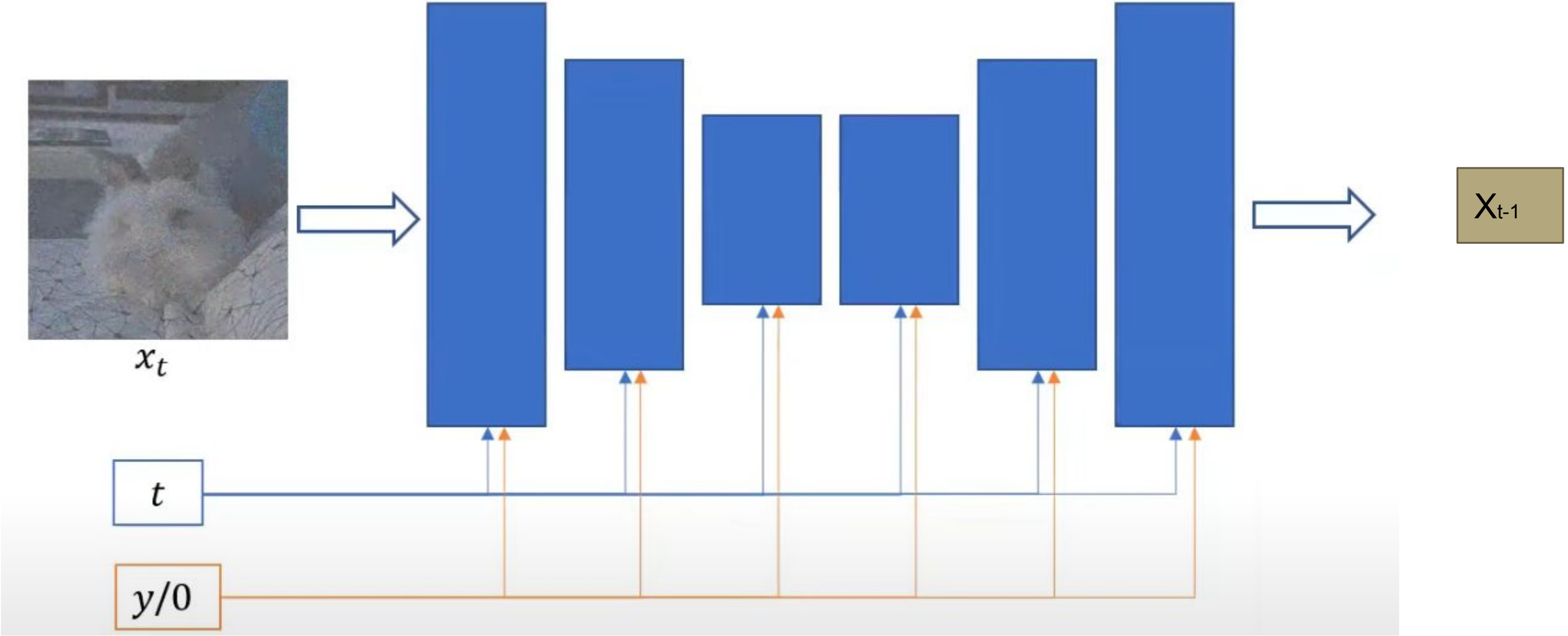
Class Guided Conditional Generation : Issues

- Training a Classifier on noisy samples is not easy

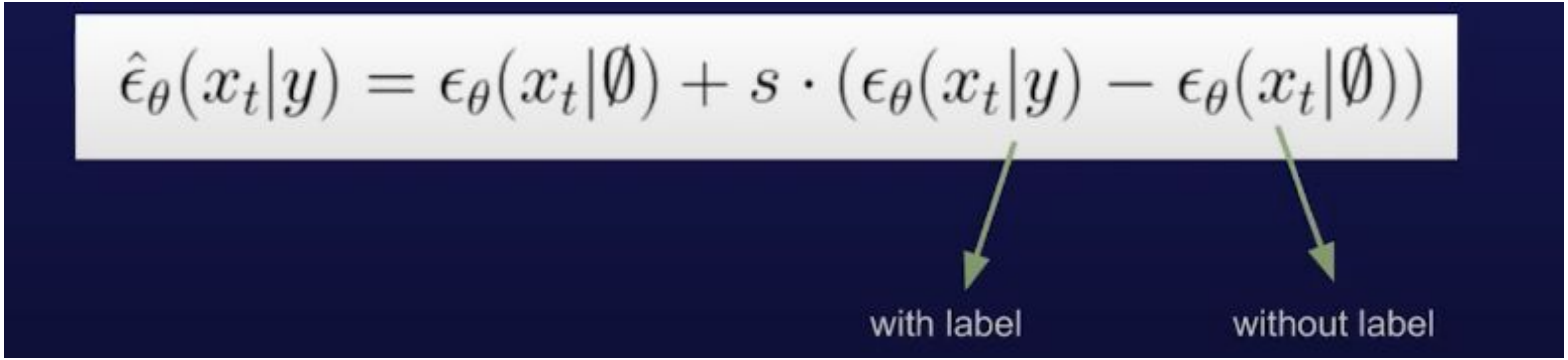


Background becomes saturated (IMAGEN solves this)

Conditioned Generation (Stable Diffusion)



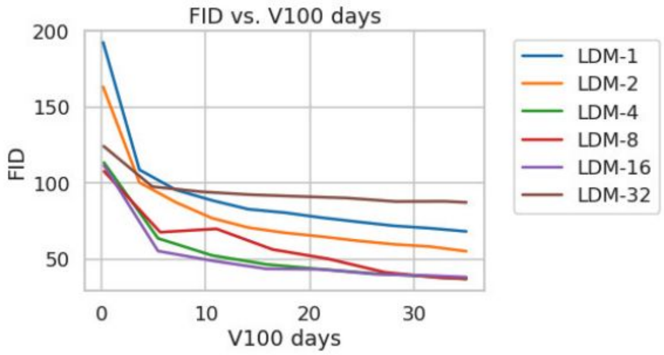
Classifier Free Guidance (Imagen)



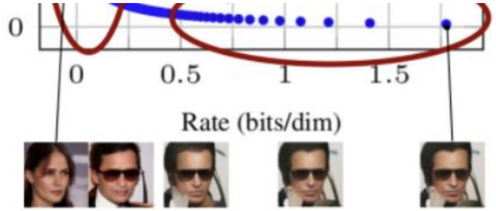
$$\hat{\epsilon}_\theta(x_t|y) = \epsilon_\theta(x_t|\emptyset) + s \cdot (\epsilon_\theta(x_t|y) - \epsilon_\theta(x_t|\emptyset))$$

Issues

- Operation in the pixel space: the sequence of denoising autoencoders always work on the high-dimension image pixel space. This is super high for high res images., leading to time consuming and GPU intensive training cycles. (150-1000 of GPU(V-100) days)



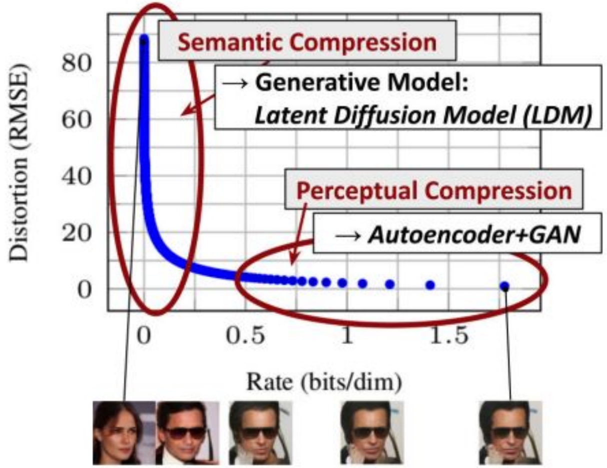
- It was also observed that performing diffusion in pixel space leads to imperceptible feature tuning which is a extra step.



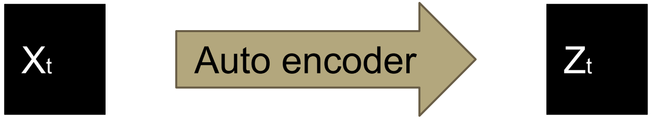
- Trade-off this extra computation in high-res pixel space for easier training still preserving image details.

Stable Diffusion

- Performs Diffusion in the latent space which is perceptually equivalent, but computationally more suitable.



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

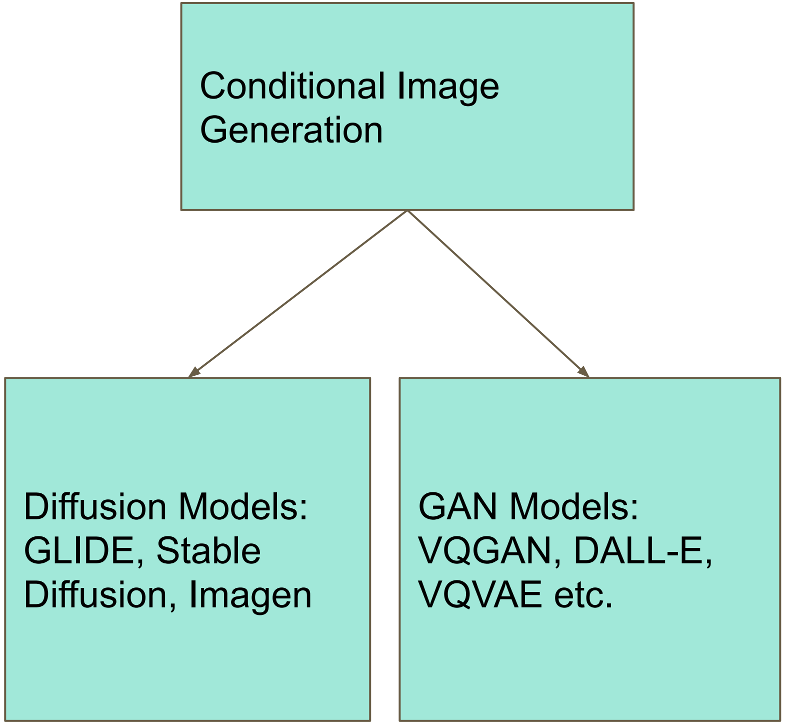


Related Work

- Autoencoders, GANS (poor in generating high resolution images)

- 2 stage training methods: VQGAN
  - Step 1: train encoder, decoder, codebook
  - STEP2: train transformer in a autoregressive manner with codebook

- Diffusion Models (DDPM)
- GLIDE : CLIP Guidance



Conditional Image Generation

Diffusion Models: GLIDE, Stable Diffusion, Imagen

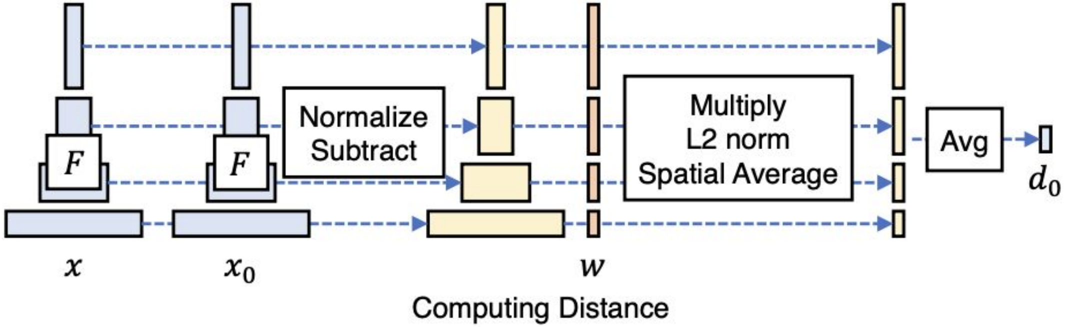
GAN Models: VQGAN, DALL-E, VQVAE etc.

Autoencoder : Perceptual Image Compression Module

- Image (H X W X 3) —-> Encoder(H/f X W/f X c) —--> Decoder (H X W X 3)
- Patch Perceptual Loss + GAN loss
- Regularization: KL on learned latent space like VAE or VQGAN style codebook learning

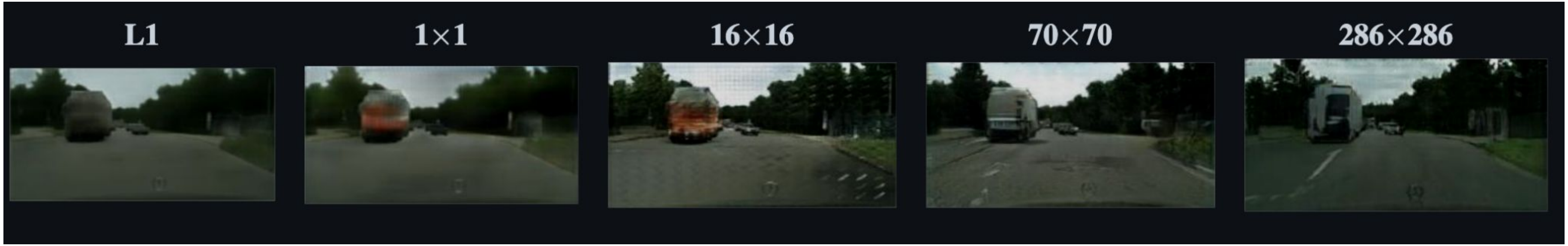
Perceptual Loss

Instead of measuring pixelwise distance, both images(real and generated) are projected to a feature space using a pretrained CNN like VGG-NET and the distance between the feature space is measured.



Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 5967–5976. IEEE Computer Society, 2017.

Patch-GAN aka pix2pix

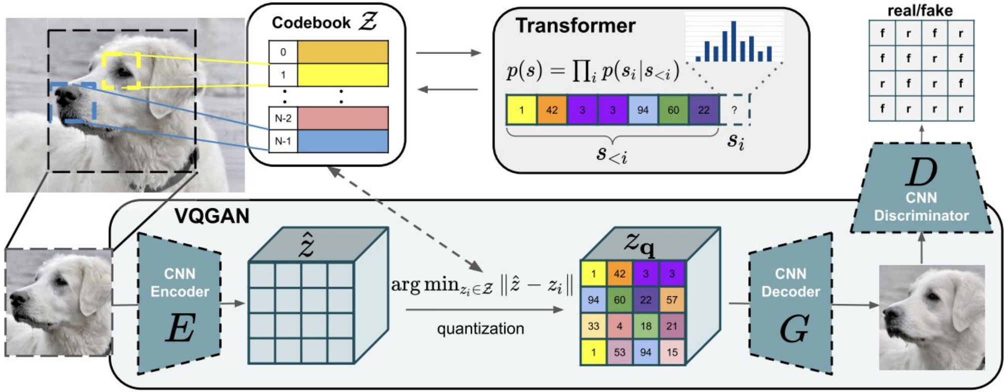


In standard GANs, the discriminator fails to capture high frequency details in image leading to blurry outputs. For the discriminator to model high frequencies, its restricted to only patches. This discriminator tries to classify if each NXN patch in an image is real or fake. Authors run this discriminator convolutionally across the image, averaging all responses to provide the ultimate output of D.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In CVPR, pages 5967–5976. IEEE Computer Society, 2017.

Regularization

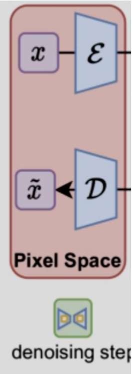
- KL to bring the latent space to be close to a prior distribution (Normal with 0 mean and unit variance).
- Discretization of the latent space by substituting with nearest neighbour using a codebook.

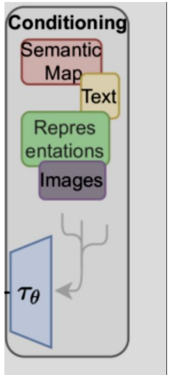


Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. *CoRR*, abs/2012.09841, 2020.

Conditioning

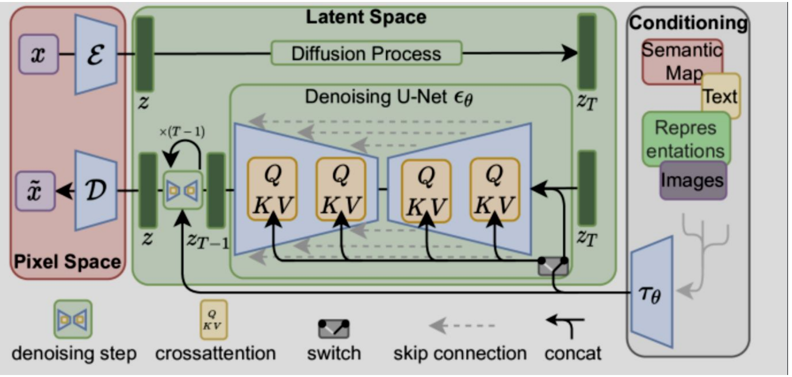
- Use a domain specific encoder to encode the conditioning.





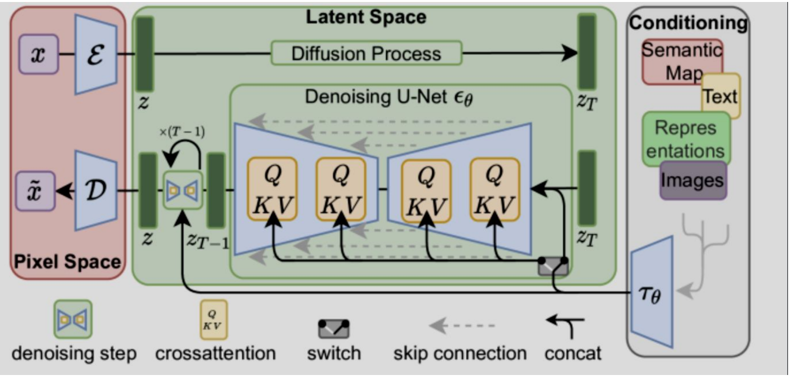
Conditioning

- Use a domain specific encoder to encode the conditioning.
- Cross attention calculated with time conditional UNET layers with Q coming from the UNET and K,V coming from the encoded conditioning.



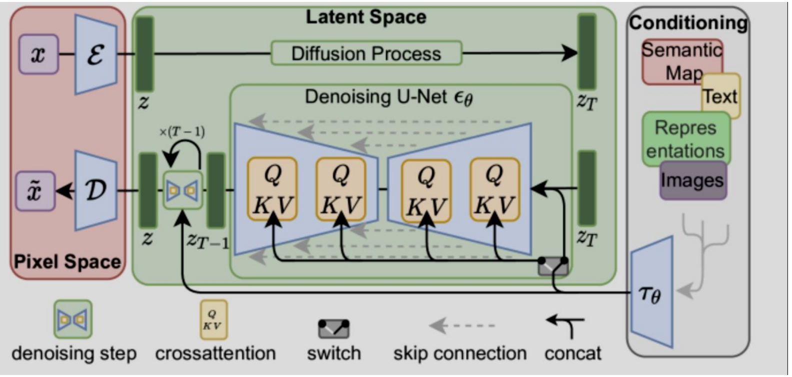
Conditioning

- Use a domain specific encoder to encode the conditioning.
- Cross attention calculated with time conditional UNET layers with Q coming from the UNET and K,V coming from the encoded conditioning.
- The conditional encoder and UNET are jointly optimized.



Conditioning

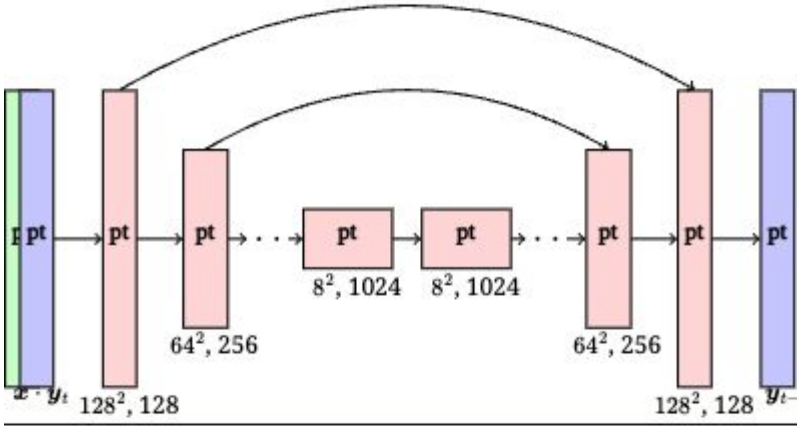
- Use a domain specific encoder to encode the conditioning.
- Cross attention calculated with time conditional UNET layers with Q coming from the UNET and K,V coming from the encoded conditioning.
- The conditional encoder and UNET are jointly optimized.



NO TIME CONDITIONING?

Super Resolution - SR3 and LDM

- The low resolution image is upsampled to the target res using bicubic interpolation
- Then it is concatenated to the conditioning y in denoising steps.
- LDM Autoencoder downsamples the high resolution into a latent space of H/4, W/4



Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. CoRR, abs/2104.07636, 2021.

Metrics

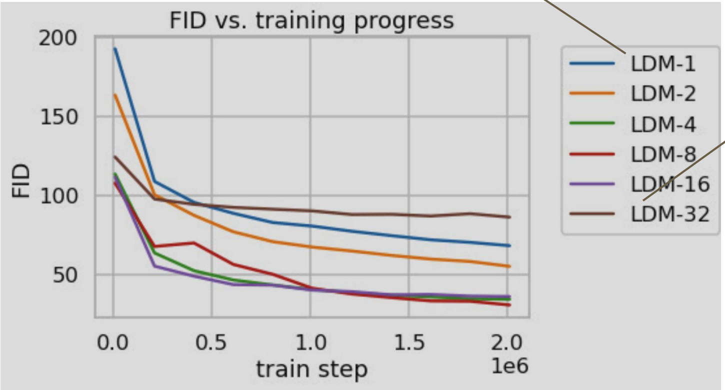
- FID Score: We use a pretrained Inception V3 on Imagenet as an encoder and calculate a measure of the difference between the real and generated image's encoding vectors. (lower is better)
- Inception Score: the KL divergence between the marginal distribution and label distribution when the image is passed through a pretrained classifier. (higher is better)

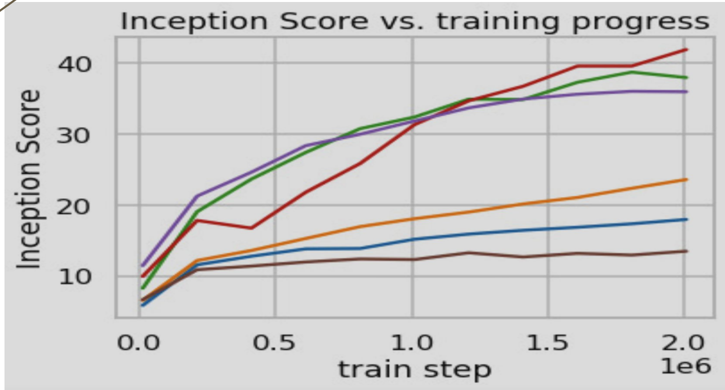
Results - Convergence wrt Diffusion

Tractable diffusion in less time
on Imagenet data

Latent space=pixel space

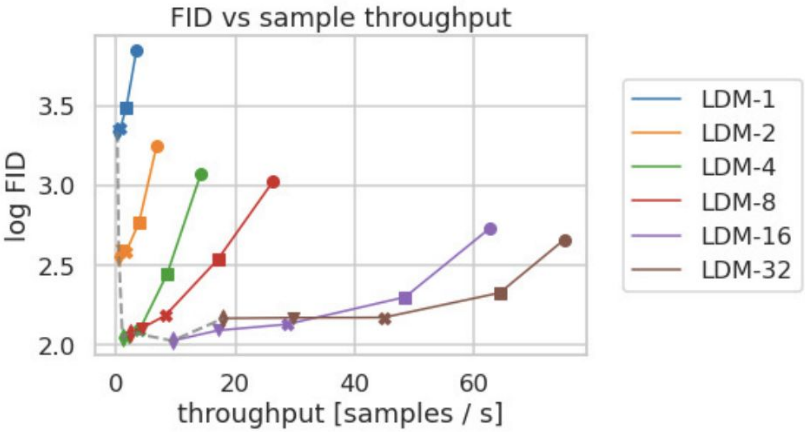
Too much compression



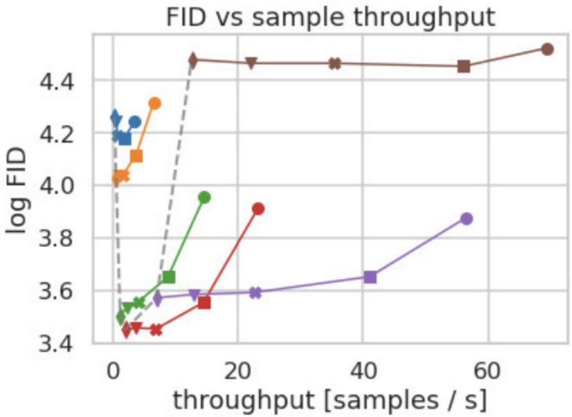


Results - Performance vs throughput

Analysis on image metrics with sampling speed. Hyperparameter: compression ratio. LDM 4-5 achieve much better(lower) FID while simultaneously improving throughput.



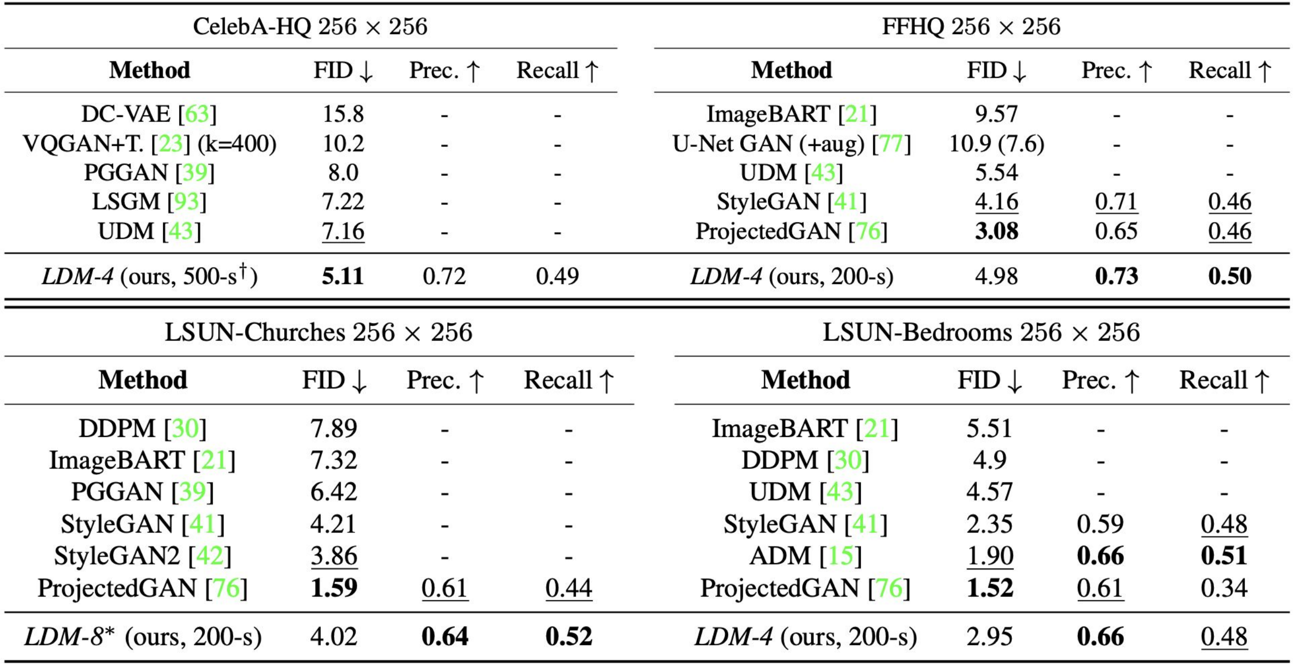
Celeb-A HQ



ImageNet

Results: Unconditional Generation

Outperform all diffusion based generation models. GAN models are worse because of low precision and recall

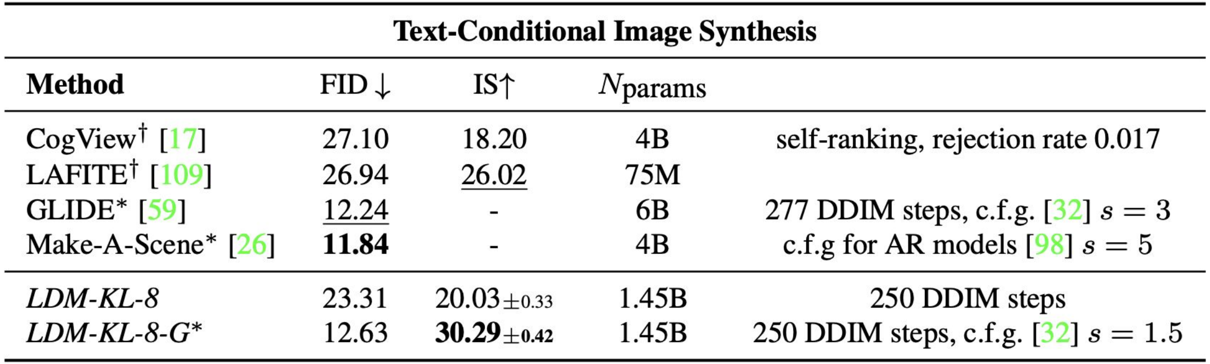


CelebA-HQ 256 × 256				FFHQ 256 × 256			
Method	FID ↓	Prec. ↑	Recall ↑	**Method**	FID ↓	Prec. ↑	Recall ↑
DC-VAE [63]	15.8	-	-	ImageBART [21]	9.57	-	-
VQGAN+T. [23] (k=400)	10.2	-	-	U-Net GAN (+aug) [77]	10.9 (7.6)	-	-
PGGAN [39]	8.0	-	-	UDM [43]	5.54	-	-
LSGM [93]	7.22	-	-	StyleGAN [41]	4.16	0.71	0.46
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

Results: Conditional Generation (text)

- Trained on LAION - 400M with BERT
- Similar performance as classifier free guidance with lesser parameter count.

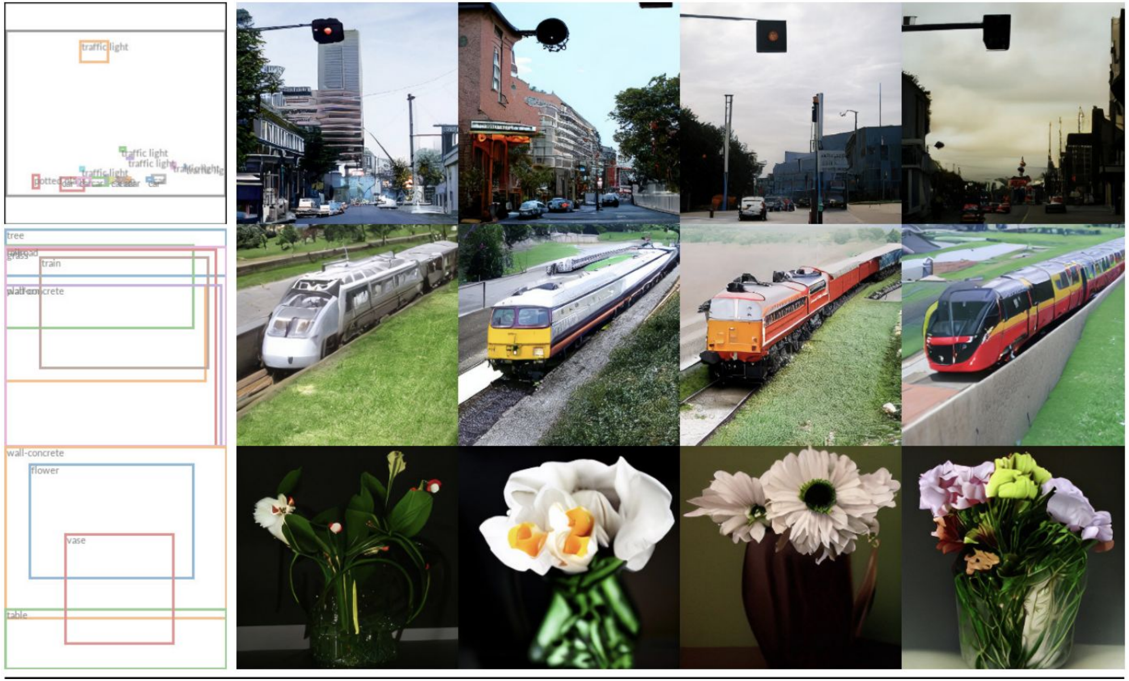


Text-Conditional Image Synthesis

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	$20.03_{\pm 0.33}$	1.45B	250 DDIM steps
*LDM-KL-8-G**	12.63	$\mathbf{30.29}_{\pm \mathbf{0.42}}$	1.45B	250 DDIM steps, c.f.g. [32] $s = 1.5$

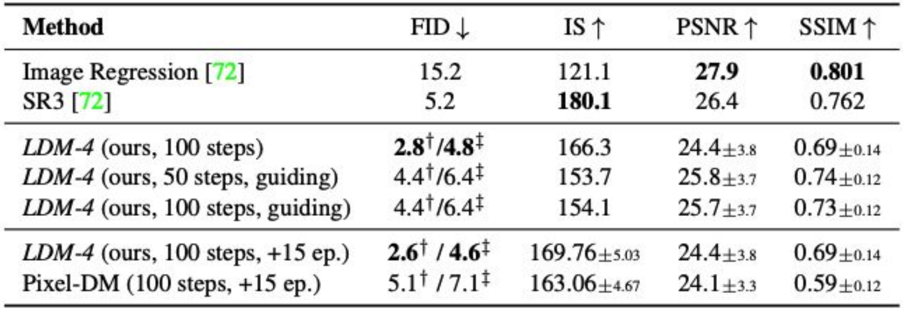
Test Performance on ImageNet

Results: Conditional Generation on layout images (COCO)

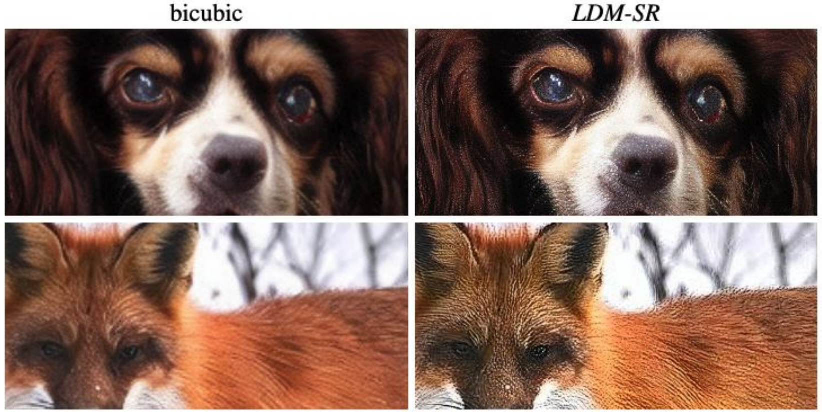


Pretrained for layouts on OpenImages and finetuned on COCO

Results: Super-Resolution



Method	FID ↓	IS ↑	PSNR ↑	SSIM ↑
Image Regression [72]	15.2	121.1	**27.9**	**0.801**
SR3 [72]	5.2	**180.1**	26.4	0.762
LDM-4 (ours, 100 steps)	$\mathbf{2.8}^{\dagger}/\mathbf{4.8}^{\ddagger}$	166.3	$24.4_{\pm 3.8}$	$0.69_{\pm 0.14}$
LDM-4 (ours, 50 steps, guiding)	$4.4^{\dagger}/6.4^{\ddagger}$	153.7	$25.8_{\pm 3.7}$	$0.74_{\pm 0.12}$
LDM-4 (ours, 100 steps, guiding)	$4.4^{\dagger}/6.4^{\ddagger}$	154.1	$25.7_{\pm 3.7}$	$0.73_{\pm 0.12}$
LDM-4 (ours, 100 steps, +15 ep.)	$\mathbf{2.6}^{\dagger}$ / $\mathbf{4.6}^{\ddagger}$	$169.76_{\pm 5.03}$	$24.4_{\pm 3.8}$	$0.69_{\pm 0.14}$
Pixel-DM (100 steps, +15 ep.)	$5.1^{\dagger}$ / $7.1^{\ddagger}$	$163.06_{\pm 4.67}$	$24.1_{\pm 3.3}$	$0.59_{\pm 0.12}$



4X Upscaling on ImageNet, trianed on OpenImages

OOD performance improves if better degradation is used

Results: Image Inpainting

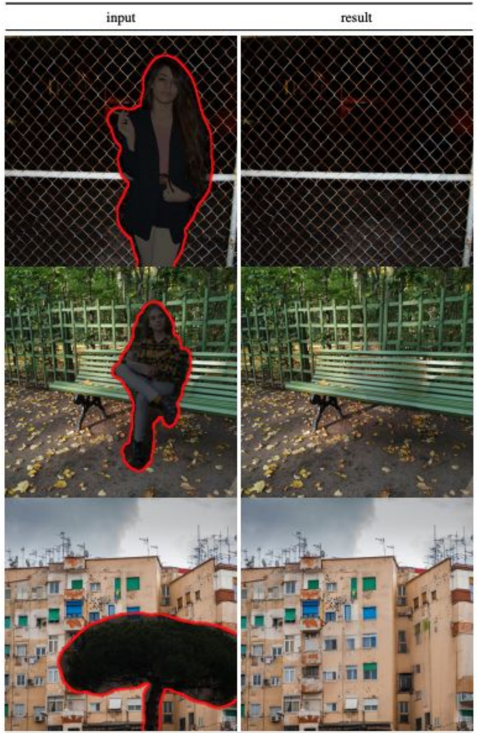
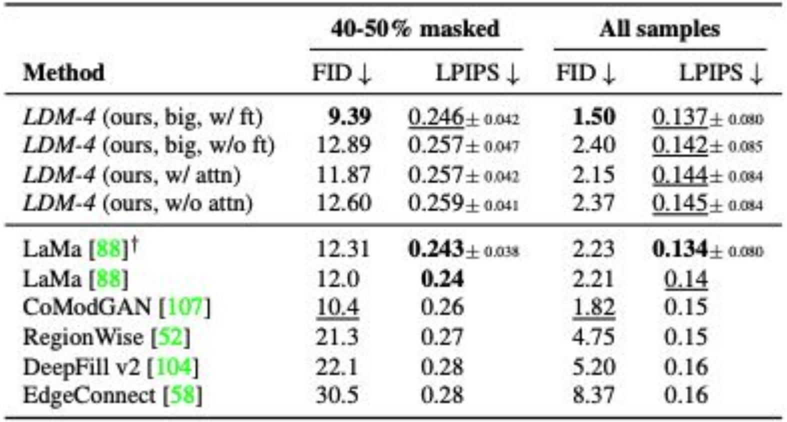


Figure 11. Qualitative results on object removal with our *big, w/ ft* inpainting model. For more results, see Fig. 22.

Inference speedup of 2.7x because of LDMs

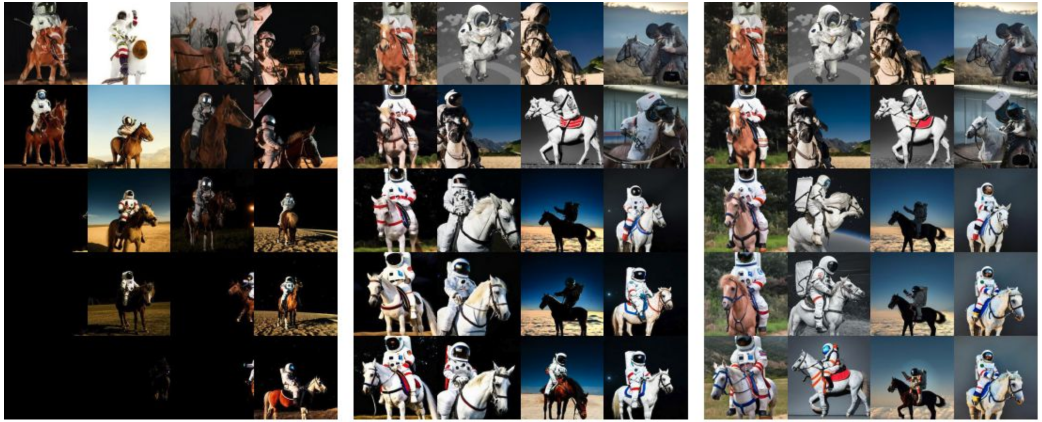


	40-50 % masked		All samples	
Method	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓
LDM-4 (ours, big, w/ ft)	**9.39**	0.246 ± 0.042	**1.50**	0.137 ± 0.080
LDM-4 (ours, big, w/o ft)	12.89	0.257 ± 0.047	2.40	0.142 ± 0.085
LDM-4 (ours, w/ attn)	11.87	0.257 ± 0.042	2.15	0.144 ± 0.084
LDM-4 (ours, w/o attn)	12.60	0.259 ± 0.041	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

SOTA FID performance

Pivot to Imagen

- Classifier - Free Guidance : High guidance weights cause image generation out of the training bounds during test cycle. This leads to unnatural images. Solution: Dynamic Thresholding



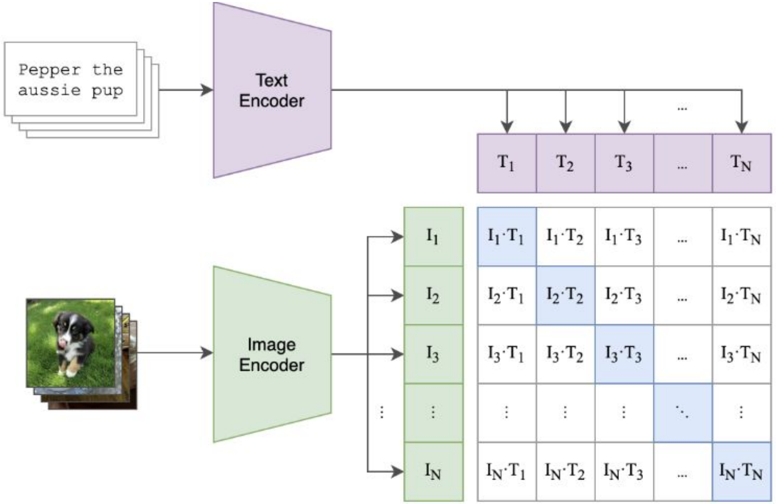
(a) No thresholding.

(b) Static thresholding.

(c) Dynamic thresholding.

Imagen : Previous Work

- GLIDE : Clip product is used to guide diffusion, but insufficient image fidelity and image text alignment
- Latent Diffusion: Needs to train an autoencoder as prior



	T_1	T_2	T_3	...	T_N
I_1	$I_1 \cdot T_1$	$I_1 \cdot T_2$	$I_1 \cdot T_3$	...	$I_1 \cdot T_N$
I_2	$I_2 \cdot T_1$	$I_2 \cdot T_2$	$I_2 \cdot T_3$	...	$I_2 \cdot T_N$
I_3	$I_3 \cdot T_1$	$I_3 \cdot T_2$	$I_3 \cdot T_3$	...	$I_3 \cdot T_N$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
I_N	$I_N \cdot T_1$	$I_N \cdot T_2$	$I_N \cdot T_3$	...	$I_N \cdot T_N$

Imagen : Introduction

- Text conditional image generation
- Uses a strong LLM’s encoder as text encoder
- No need of learning a prior for latent space generation.

Imagen : Key Contributions

- Shows that scaling text encoder is more important than scaling image diffusion model.
- Dynamic Thresholding
- Efficient UNet as underlying image model
- Drawbench : a new evaluation benchmark for text to image

Imagen Architecture Overview

Components:

1. Text Encoder
2. Image Denoiser (UNET)
3. CF Guidance : Static and Dynamic THresholding

Imagen Architecture Overview

Components:

1. Text Encoder
2. Image Denoiser (UNET)
3. CF Guidance : Static and Dynamic THresholding
4. Cascaded Diffusion Models

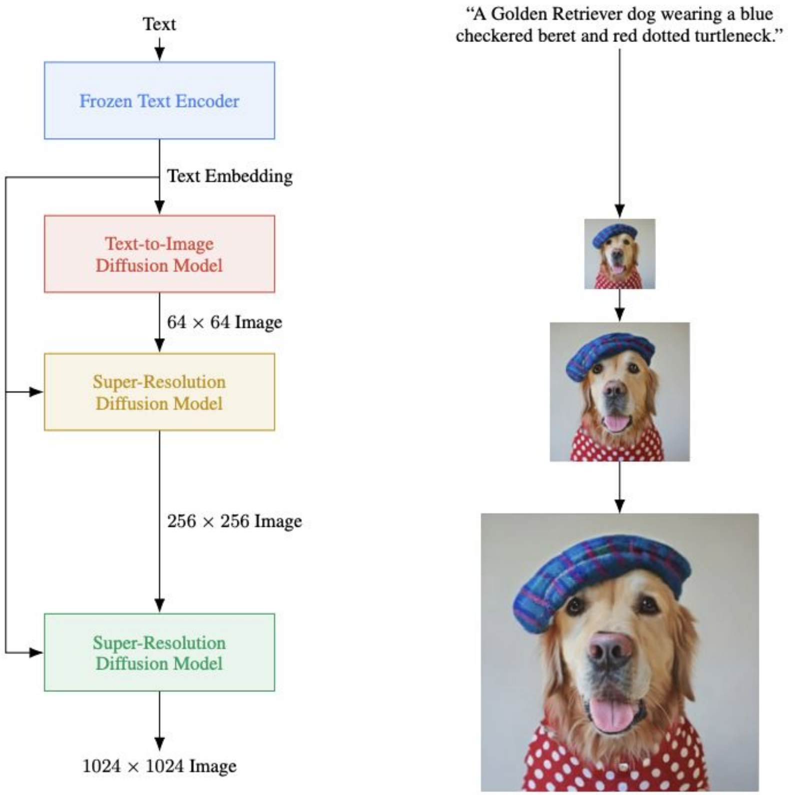
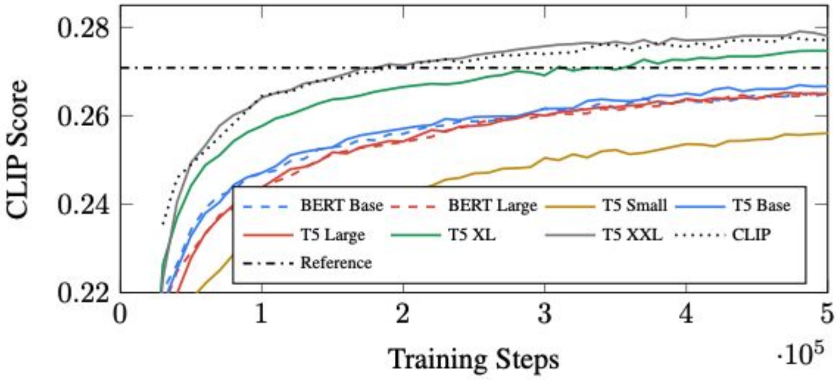
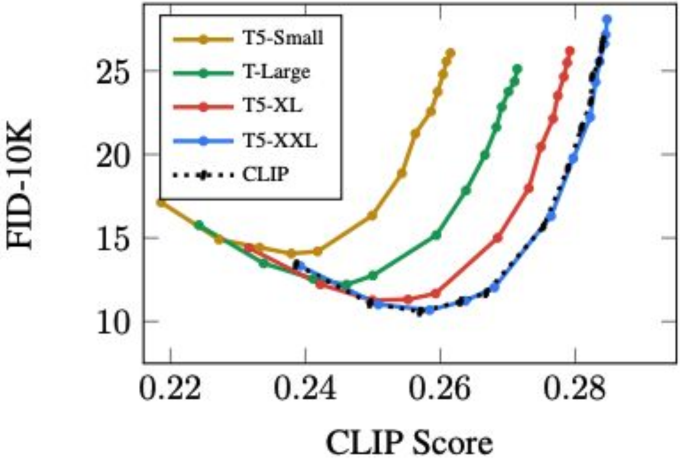

Text
Frozen Text Encoder
Text Embedding
Text-to-Image Diffusion Model
64 × 64 Image
Super-Resolution Diffusion Model
256 × 256 Image
Super-Resolution Diffusion Model
1024 × 1024 Image
“A Golden Retriever dog wearing a blue checkered beret and red dotted turtleneck.”

Imagen - Text Encoder

- Scale of text encoders is VERY important, authors choose T5-XXL Encoder
- Freeze the text encoder weights



Training convergence comparison between text encoders for text-to-image generation.



(a) Pareto curves comparing various text encoders.

Imagen : Base UNET 64X64

Text encodings Conditioning:

- Pooled Embedding Vector with time encodings added
- Cross attention similar to Latent Diffusion Model

Efficient UNET for superresolution

Resnet Block Changes

- Minor resnet changes like shifting model params from highres to lowres blocks, downsampling before conv and upsampling after convs.

256-> 1024

- UNets(64X64 -> 256X256) running on crops of 1024X1024 pixel images. No self attention

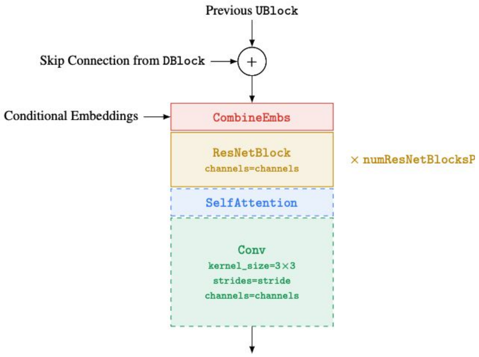
Efficient UNET for superresolution

Resnet Block Changes

- Minor resnet changes like shifting model params from highres to lowres blocks, downsampling before conv and upsampling after convs.

256-> 1024

- UNets(64X64 -> 256X256) running on crops of 1024X1024 pixel images. No self attention



Unet U Block, dotted blocks are present depending on resolution

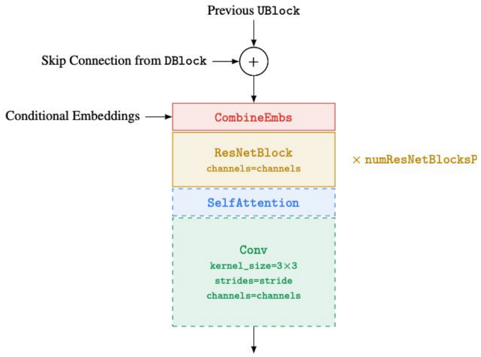
Efficient UNET for superresolution

Resnet Block Changes

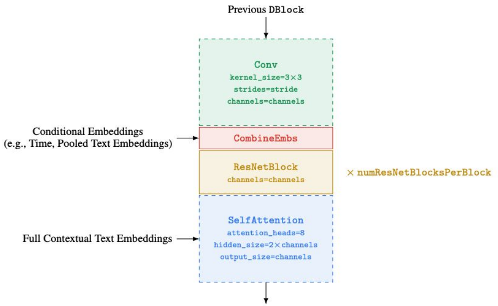
- Minor resnet changes like shifting model params from highres to lowres blocks, downsampling before conv and upsampling after convs.

256-> 1024

- UNets(64X64 -> 256X256) running on crops of 1024X1024 pixel images. No self attention



Unet U Block, dotted blocks are present depending on resolution



Unet D Block, dotted blocks are present depending on resolution

Thresholding of Generated Images To Avoid Instability

Static: Clip prediction to [-1,1] i.e. in bounds with training data , But leads to oversaturated images.

Thresholding of Generated Images To Avoid Instability

Static: Clip prediction to [-1,1] i.e. in bounds with training data , But leads to oversaturated images.

Dynamic: x - > [-s,s] where s threshold decided by a percentile of absolute pixel values , and if s>1, it is scaled to [-1,1]. Prevent pixel saturation and leads to photorealism.

Cascaded Diffusion Models

- During training, the low resolution images are gaussian Blurred with a randomly selected value and then condition the super-resolution model with this noise to denoise it i.e. superres module is aware of noise

Cascaded Diffusion Models

- During training, the low resolution images are gaussian Blurred with a randomly selected value and then condition the super-resolution model with this noise to denoise it i.e. superres module is aware of noise

- During Inference, the augmentation parameter is tuned.

- [illegible]

Cascaded Diffusion Models

- During training, the low resolution images are gaussian Blurred with a randomly selected value and then condition the super-resolution model with this noise to denoise it i.e. superres module is aware of noise

- During Inference, the augmentation parameter is tuned.

- Advantage 1: Robustness to handle artifacts generated by small scale Diffusion Models

- Advantage 2: Improves sample quality and generated high fidelity images

DrawBench - A small test set for human evaluation

- COCO has a limited set of prompts
- Drawbench is a test set of 200 prompts containing 11 categories of prompts to test if model can render different colors, numbers of objects, spatial relations, text in the scene, and unusual interactions between objects.
- The prompts are also complex textually

Training Details

- **Parameter Count :** 2B parameters for 64X64 image gen + 600M for 256X256 + 400M for 1024X1024
- **Optimizer :** Adam for superres and Adafactor for 64X64
- **Classifier Free**: Zeroed text encodings
- **Training Set:** Internal and Unreleased (460M Image-Text pairs) + Laion(400M pairs)

Results: Text-to-Image FID on COCO

Dataset: COCO Eval set

SOTA results using FID metric on Zero Shot

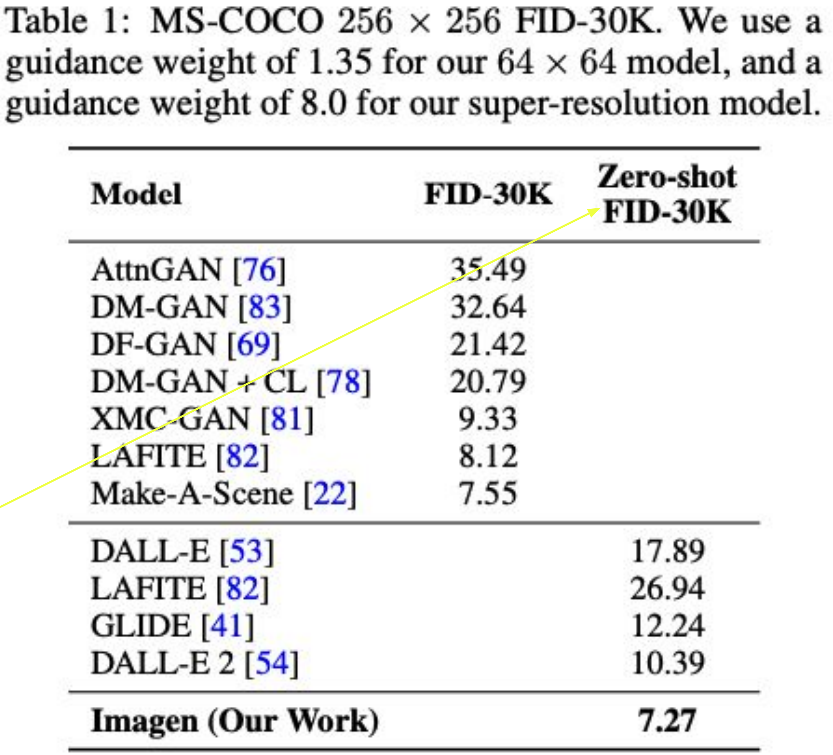


Table 1: MS-COCO 256 × 256 FID-30K. We use a guidance weight of 1.35 for our 64 × 64 model, and a guidance weight of 8.0 for our super-resolution model.

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

Data on which Imagen is trained is not released. Neither is any alignment of that with COCO

Results: User Study metrics on COCO

- The authors found issues that FID is not aligned with image quality.

Results: User Study metrics on COCO

- The authors found issues that FID is not aligned with image quality.
- CLIP score’s weakness at counting object illustrating alignment issues with the metric.

Results: User Study metrics on COCO

- The authors found issues that FID is not aligned with image quality.
- CLIP score's weakness at counting object illustrating alignment issues with the metric.
- **Solution:** User study to rank photorealistic nature and alignment with text.

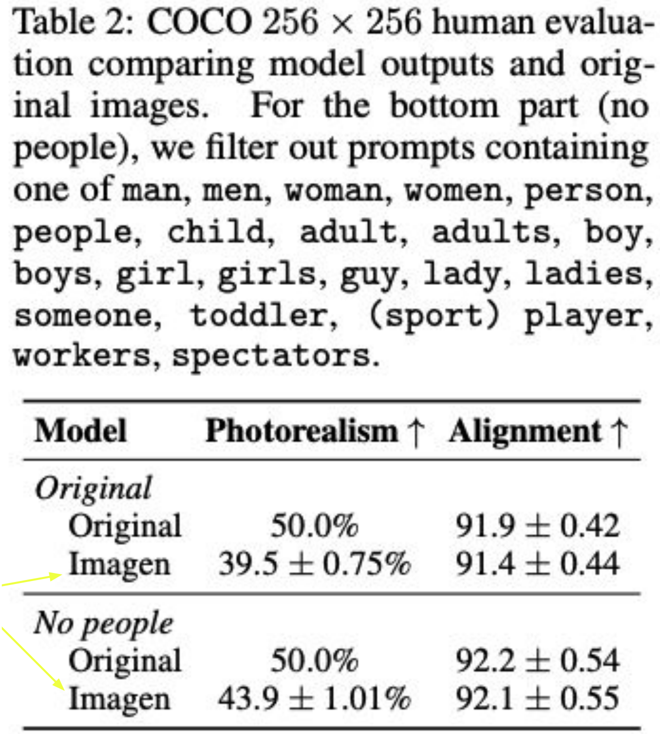


Table 2: COCO 256 × 256 human evaluation comparing model outputs and original images. For the bottom part (no people), we filter out prompts containing one of man, men, woman, women, person, people, child, adult, adults, boy, boys, girl, girls, guy, lady, ladies, someone, toddler, (sport) player, workers, spectators.

Model	Photorealism ↑	Alignment ↑
Original		
Original	50.0%	91.9 ± 0.42
Imagen	$39.5 \pm 0.75\%$	91.4 ± 0.44
No people		
Original	50.0%	92.2 ± 0.54
Imagen	$43.9 \pm 1.01\%$	92.1 ± 0.55

Results: User Study metrics on COCO

- The authors found issues that FID is not aligned with image quality.
- CLIP score's weakness at counting object illustrating alignment issues with the metric.
- **Solution:** User study to rank photorealistic nature and alignment with text.

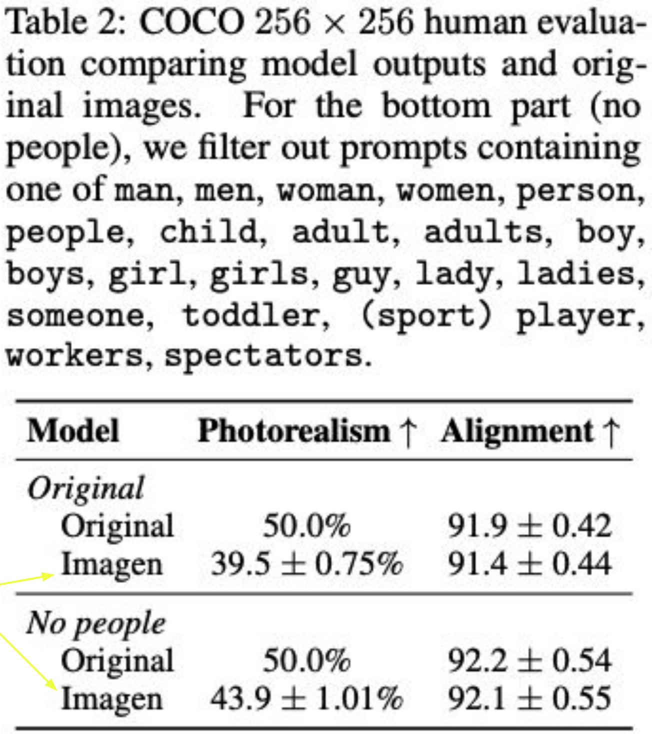
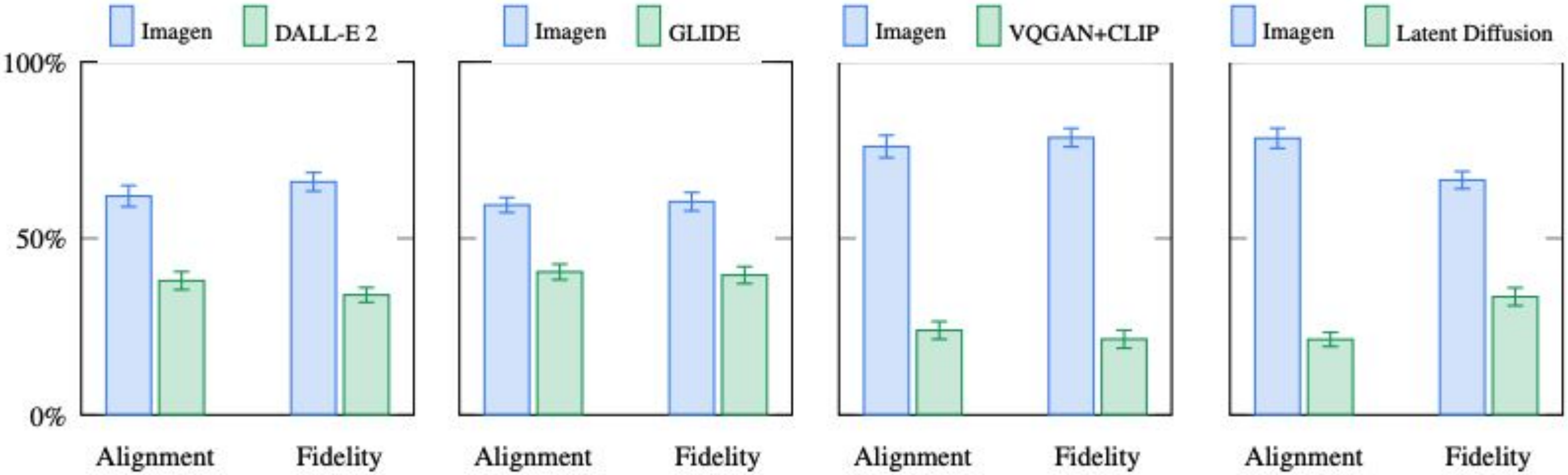


Table 2: COCO 256 × 256 human evaluation comparing model outputs and original images. For the bottom part (no people), we filter out prompts containing one of man, men, woman, women, person, people, child, adult, adults, boy, boys, girl, girls, guy, lady, ladies, someone, toddler, (sport) player, workers, spectators.

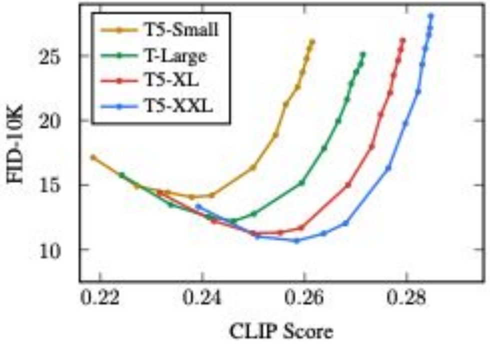
Model	Photorealism ↑	Alignment ↑
Original		
Original	50.0%	91.9 ± 0.42
Imagen	39.5 ± 0.75%	91.4 ± 0.44
No people		
Original	50.0%	92.2 ± 0.54
Imagen	43.9 ± 1.01%	92.1 ± 0.55

Poor at generating People

Results: User Study metrics on DrawBench

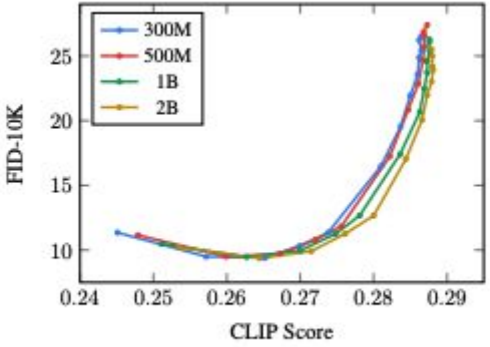


Ablations



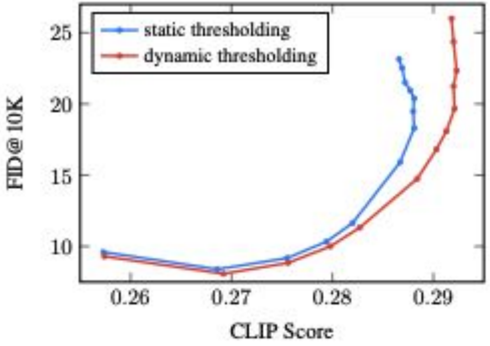
(a) Impact of encoder size.

Text Encoder Size is Important (Human study confirms this)



(b) Impact of U-Net size.

Image Encoder Size is not Important



(c) Impact of thresholding.

Dynamic Thresholding importance

Ablations (Cont)

- Noise Conditioning is important for superresolution and shows in CLIP, FID scores

Ablations (Cont)

- Noise Conditioning is important for superresolution and shows in CLIP, FID scores
- Efficient UNET causes faster inference and converges faster during training

Strengths

- Authors find limitations in COCO dataset and introduce a new DrawBench benchmark set to evaluate text conditional image generation models.
- SOTA FID results on COCO
- Importance of text encoders.

Weaknesses

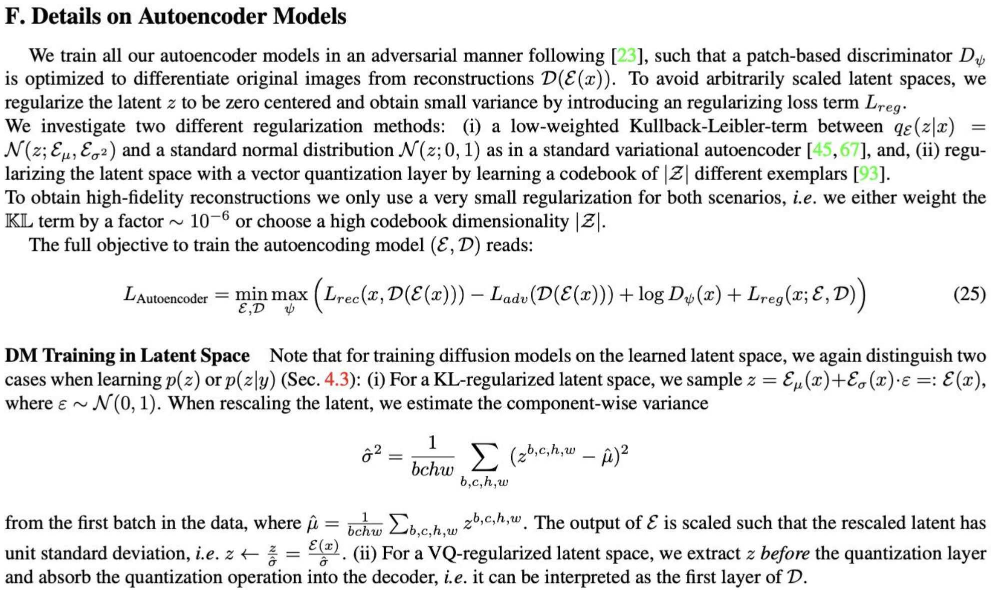
- The training data and model is not released (Even with a non commercial licence)
- Bias concerns due to social and cultural exclusions in generations have not been handled
- The training parameter count is 3B which is great, but inference cycle cintainf T5XXL encoding which has 11 B additional parameters i.e. 14B on inference

Discussion Points

- Extension of this from text guided to image guided can be studied.
- Why can’t we learn diffusion in a latent space instead of pixel space

Thank You!

BAK : Stable Diffusion autoencoder training objective



F. 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 $\mathcal{D}(\mathcal{E}(x))$. To avoid arbitrarily scaled latent spaces, we regularize the latent z to be zero centered and obtain small variance by introducing an regularizing loss term L_{reg}.
We investigate two different regularization methods: (i) a low-weighted Kullback-Leibler-term between $q_{\mathcal{E}}(z|x) = \mathcal{N}(z; \mathcal{E}_\mu, \mathcal{E}_{\sigma^2})$ and a standard normal distribution $\mathcal{N}(z; 0, 1)$ as in a standard variational autoencoder [45, 67], and, (ii) regularizing the latent space with a vector quantization layer by learning a codebook of $|\mathcal{Z}|$ different exemplars [93].
To obtain high-fidelity reconstructions we only use a very small regularization for both scenarios, *i.e.* we either weight the $\mathbb{KL}$ term by a factor $\sim 10^{-6}$ or choose a high codebook dimensionality $|\mathcal{Z}|$.

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

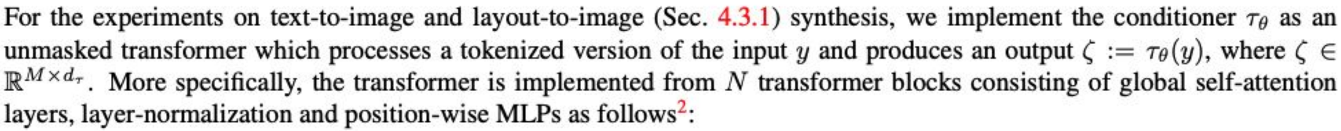
$$L_{\text{Autoencoder}} = \min_{\mathcal{E},\mathcal{D}} \max_{\psi} \Big(L_{rec}(x, \mathcal{D}(\mathcal{E}(x))) - L_{adv}(\mathcal{D}(\mathcal{E}(x))) + \log D_\psi(x) + L_{reg}(x; \mathcal{E}, \mathcal{D}) \Big) \tag{25}$$

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): (i) For a KL-regularized latent space, we sample $z = \mathcal{E}_\mu(x) + \mathcal{E}_\sigma(x) \cdot \varepsilon =: \mathcal{E}(x)$, where $\varepsilon \sim \mathcal{N}(0, 1)$. When rescaling the latent, we estimate the component-wise variance

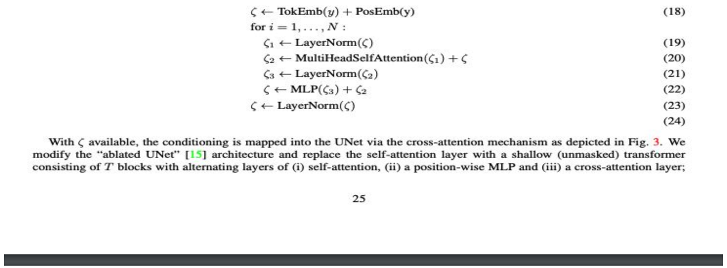
$$\hat{\sigma}^2 = \frac{1}{bchw} \sum_{b,c,h,w} (z^{b,c,h,w} - \hat{\mu})^2$$

from the first batch in the data, where $\hat{\mu} = \frac{1}{bchw} \sum_{b,c,h,w} z^{b,c,h,w}$. The output of $\mathcal{E}$ is scaled such that the rescaled latent has unit standard deviation, *i.e.* $z \leftarrow \frac{z}{\hat{\sigma}} = \frac{\mathcal{E}(x)}{\hat{\sigma}}$. (ii) 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 $\mathcal{D}$.

BAK : Stable Diff denoiser UNET and condition model

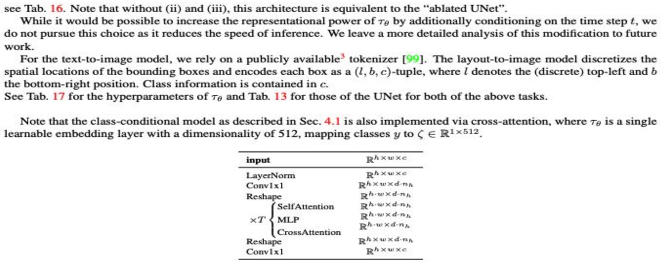


For the experiments on text-to-image and layout-to-image (Sec. 4.3.1) synthesis, we implement the conditioner τ_θ as an unmasked transformer which processes a tokenized version of the input y and produces an output $\zeta := \tau_\theta(y)$, where $\zeta \in \mathbb{R}^{M \times d_\tau}$. More specifically, the transformer is implemented from N transformer blocks consisting of global self-attention layers, layer-normalization and position-wise MLPs as follows[2]:



$$
\begin{aligned}
&\zeta \leftarrow \text{TokEmb}(y) + \text{PosEmb}(y) && (18)\\
&\text{for } i = 1, \ldots, N: \\
&\quad \zeta_1 \leftarrow \text{LayerNorm}(\zeta) && (19)\\
&\quad \zeta_2 \leftarrow \text{MultiHeadSelfAttention}(\zeta_1) + \zeta && (20)\\
&\quad \zeta_3 \leftarrow \text{LayerNorm}(\zeta_2) && (21)\\
&\quad \zeta \leftarrow \text{MLP}(\zeta_3) + \zeta_2 && (22)\\
&\zeta \leftarrow \text{LayerNorm}(\zeta) && (23)\\
& && (24)
\end{aligned}
$$

With ζ 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;



see Tab. 16. Note that without (ii) and (iii), this architecture is equivalent to the "ablated UNet".

While it would be possible to increase the representational power of τ_θ by additionally conditioning on the time step t, we do not pursue this choice as it reduces the speed of inference. We leave a more detailed analysis of this modification to future work.

For the text-to-image model, we rely on a publicly available[3] tokenizer [99]. 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 $\zeta \in \mathbb{R}^{1 \times 512}$.

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