NEW FRONTIERS IN IMAGE CREATION WITH DIFFUSION GENERATIVE AI

Advanced Machine Learning (AL_KSAIM_9_1)

MSc in Software Design with Artificial Intelligence

Technological University of the Shannon

A00315339: Vasyl Dykun

wasyldykun@gmail.com

2024

Contents

Introduction

Image generative AI models have rapidly gained prominence in machine learning. Implementations like DALL-E 2 and Stable Diffusion, among others, have garnered widespread attention from both online and research communities due to their ability to generate images from text prompts. Their capability to produce high-quality, realistic images demonstrates significant potential in this area of study. Despite variations in their approaches, these models fundamentally rely on the diffusion algorithm.

This paper focuses on descriptive research into recently developed diffusion algorithms—Latent Space Diffusion, Adversarial Diffusion Distillation, and ControlNet—and their image generation capabilities.

Diffusion model

This section aims to offer a brief description of the diffusion model architecture, which is crucial for understanding subsequent architectures. Figure 1 illustrates the core concept behind diffusion models:

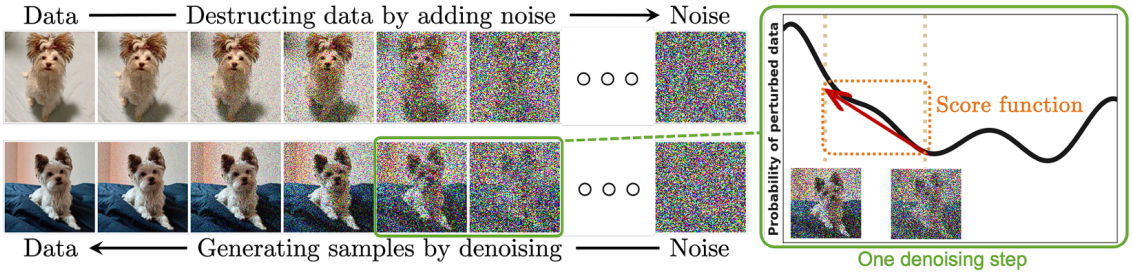

Figure 1. Diffusion Models training process [1]

Diffusion Models are a family of probabilistic generative models that iteratively add noise to data and subsequently learn to reverse this process to generate samples [1]. The generic pipeline of diffusion models comprises three main components:

The forward process (diffusion) - perturbs a training sample x_0 into its noise representation x_T by gradually adding a small amount of noise ϵ_t, across timestep t [2].

Figure 2. The forward process incrementally introduces noise to the original distribution [2]

The reverse process (denoising) - trains a neural network with trainable parameters θ to transform x_T back to x_0 [2]. The denoising process is recursive and is not one step like in GANs [2].

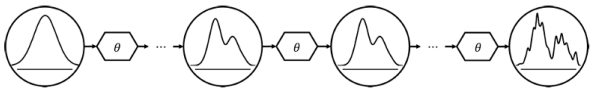

Figure 3. The reverse process trains a neural network to iteratively remove noise that was previously added [2]

The sampling procedure - uses the optimised neural network $\theta^*$to generate new data x_0^* by gradually removing noise from the noise sample [2].

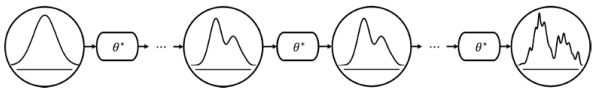

Figure 4. The sampling procedure uses a noise sample and creates new original distributions by gradually removing noise [2]

By integrating these three components, diffusion models can be trained on image datasets and then used to synthesize high-resolution, photorealistic images with high variance [3]. Below, we will explore in detail three recently developed models that employ this approach: Latent Space Diffusion, Adversarial Diffusion Distillation, and ControlNet.

Latent space diffusion

A concise overview of the architecture

The latent diffusion model represents a modification of the traditional diffusion model tailored for images. This architecture was introduced by Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer in the paper 'High-resolution image synthesis with latent diffusion models' [3]. The architecture is illustrated in Figure 5.

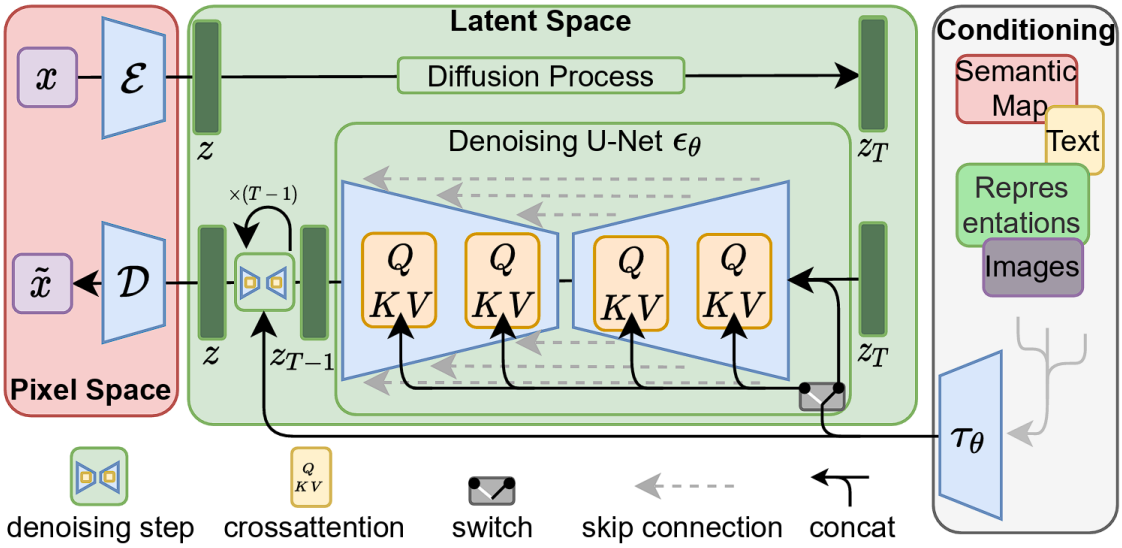

Figure 5. Architecture of the latent diffusion model [3]

The architecture aims to address one of the significant challenges associated with diffusion models: the extensive computational requirements [3]. To overcome this, their model leverages a pretrained perceptual image compression model that transforms the pixel space into a perceptually equivalent, yet far more compact, latent space [3].

The main components of the architecture include:

Encoding: The image $x \in \mathbb{R}^{H \times W \times 3}$ is processed by the encoder $\mathcal{E}$ producing a latent representation $z = \mathcal{E}(x)$ [3].

Decoding: The decoder $\mathcal{D}$ reconstructs the image from the latent representation $\tilde{x} = \mathcal{D}(z) = \mathcal{D}(\mathcal{E}(x))$, where $z \in \mathbb{R}^{H \times W \times 3}$ [3].

Encoding and decoding allow the diffusion and denoising processes to be applied in the more compact and abstract latent space, rather than the pixel space. This approach enables the model to concentrate on the most semantically significant data, allowing for training in a computationally more efficient space [3].

The diffusion process transforms latent representation $z = \mathcal{E}(x)$ into noise representation $z_T \sim \mathcal{N}(0, 1)$ [3].

For the **denoising process** the model employs U-Net architecture $\epsilon_\theta(\circ, t)$ [3]. Its loss function:

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

An additional feature implemented by the authors in the U-Net is **cross-attention mechanisms**. These mechanisms allow for the synthetic process to be controlled using y inputs such as text and representations [3]. To integrate y inputs into the U-Net, the authors utilize domain-specific encoders, τ_θ, which transform inputs into intermediate representations $\tau_\theta(\mathrm{y}) \in \mathbb{R}^{H \times W \times 3}$ [3]. These representations are then connected to intermediate layers of the U-Net via cross-attention layers [3].

Key contributions and innovations

The Latent Diffusion model introduces two key innovations:

- It utilizes pretrained **perceptual image compression** models to transform pixel space into more compact latent representations. Whereas previous diffusion models demanded extensive computational resources to generate high-resolution, photorealistic images, Latent Diffusion models mitigate this by operating within a significantly reduced, lower-dimensional latent space.
- It incorporates cross-attention layers within the U-Net architecture, enabling the inclusion of conditions such as text or bounding boxes in the generation process. The authors note that this research area, particularly the application beyond class labels or blurred image variants, was previously underexplored [3].

Strengths and limitations

The latent diffusion model has the following strengths and limitations:

Strengths

- The model shows great results, providing photorealistic images.
- The model works well with conditional and unconditional generation.
- Low computation cost; capable to generate images on consumer computers.
- The algorithm is open source and free.
- The implication of the algorithm, “Stable Diffusion”, provides trained public models.

Limitations

- Requiring much less computation resources that other diffusion approaches, is still much slower in sampling process than GAN [3].
- Image quality loss because uses autoencoder; can’t be used for tasks that require fine-graded accuracy in pixel space [3].

Performance and benchmark

Image generation

The latent diffusion model is capable of creating various types of photorealistic images. Figure 6 presents samples from models trained on datasets such as CelebAHQ, FFHQ, LSUN-Churches, LSUN-Bedrooms, and class-conditional ImageNet. Figure 7 displays samples generated from user-defined prompts after training the model on the LAION database [3].

- The model shows great results with generating faces, most of them look natural, and the observer need pay attention to spot anomalies.
- The model shows good results with generating architecture and rooms, although anomalies can be observed.
- The model exhibits impressive capabilities in abstraction and semantic understanding; It can generate text when the prompt specifies; It can apply styles to generated images.

Figure 6. Samples from latent space diffusion model [3]

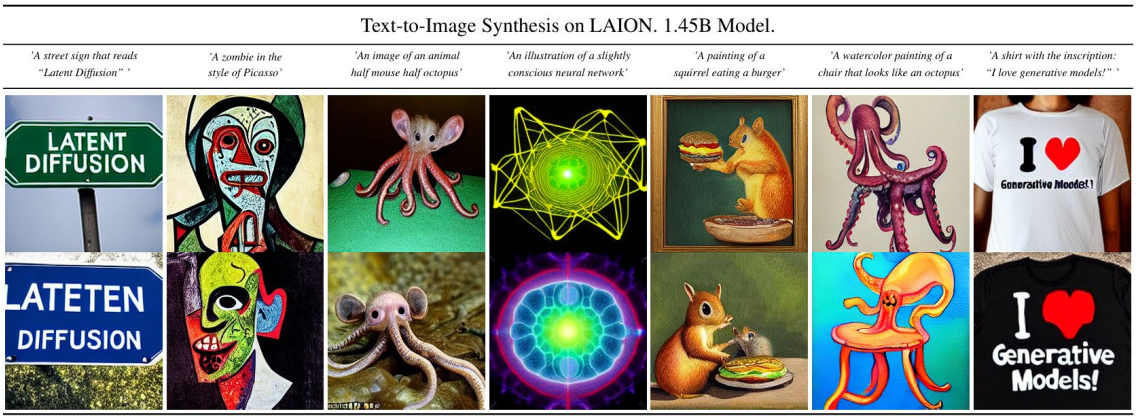

Figure 7. Samples for user-defined text prompts [3]

Performance

Table 1 and Table 2 show performance metrics of latent diffusion model (LDM) in comparison to other methods [3]. The metrics show that LDM outperform methods such as DC-VAE, ImageBART, DDPM, and several other in unconditional generation. GAN based variation overall still show better results than LDM. For text-conditional generation LDM surpass CogView, LAFITE methods while being on par with GLIDE and Make-A-Scene methods. At the same time LDM uses only 1.45B *N*params in comparison to other methods.

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 [3]

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	<u>26.02</u>	75M	
GLIDE* [59]	<u>12.24</u>	-	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] $s = 1.5$

Table 2. Evaluation of text-conditional image synthesis on the 256 x 256-sized MS-COCO dataset [3]

Applications

One of the most widely known implementation of latent diffusion model is "**Stable Diffusion**", created by Stability AI.

Link to GitHub: https://github.com/Stability-AI/StableDiffusion

Link to models: https://huggingface.co/stabilityai

Link to demo: https://huggingface.co/spaces/stabilityai/stable-diffusion

Adversarial Diffusion Distillation

A concise overview of the architecture

Adversarial Diffusion Distillation (ADD) is a novel training approach for diffusion models, introduced by Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach in 'Adversarial Diffusion Distillation,' aimed at addressing a key limitation of diffusion models: the slow sampling process. Figure 8 showcases the architecture of this approach.

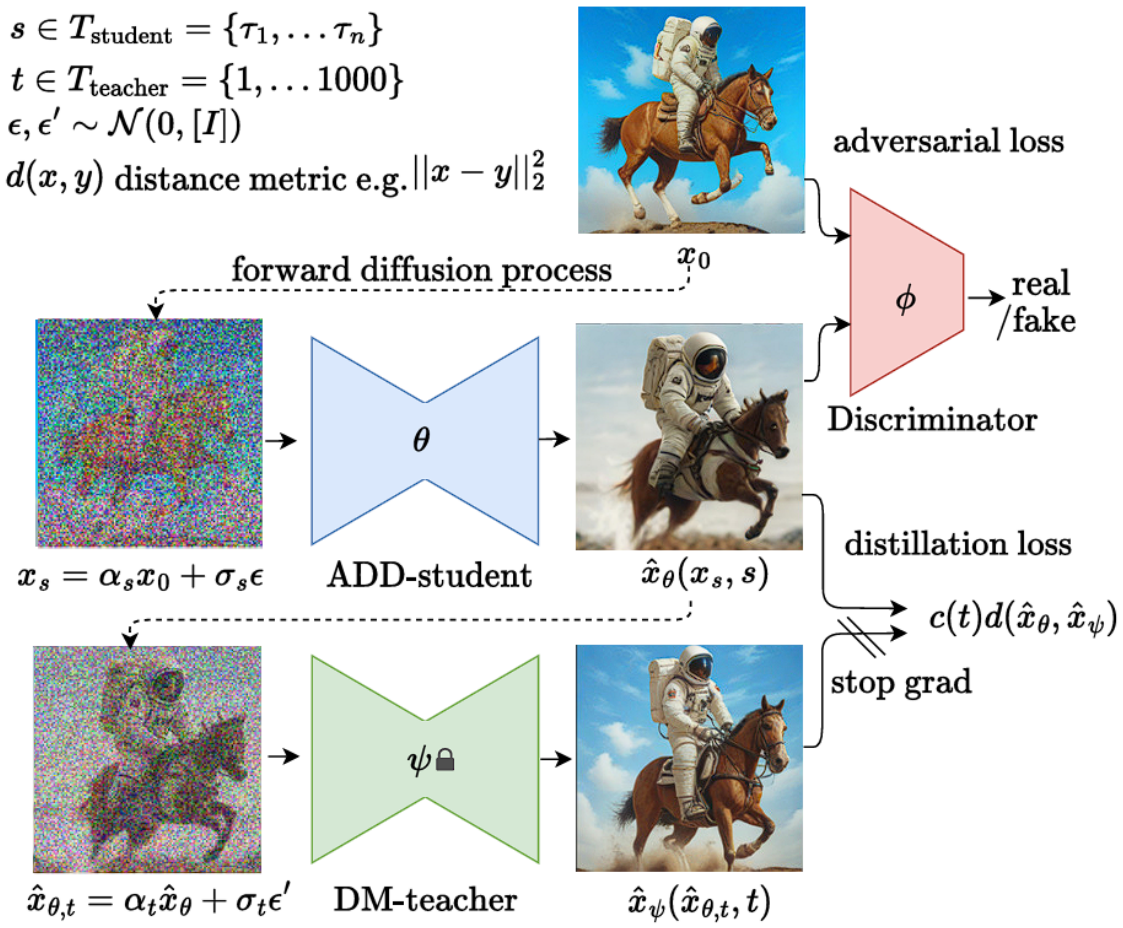

Figure 8. Adversarial Diffusion Distillation approach [3]

In their work, the authors present an algorithm that enables diffusion models to generate images with only 1-4 sampling steps [4]. The training process mirrors that of traditional diffusion models: real images x_0 undergo a forward diffusion process $x_s = a_s x_0 + \sigma_s \epsilon$ to produce noisy data x_s which is then utilized to train U-Net networks with weights θ to reverse the process and generate images $\tilde{x}_\theta = x_0$.

Key distinctions are:

- The student model initiates with weights θ from an already pretrained U-Net diffusion model [4].
- The student model employs a small number of denoising steps $N = 4$ [4].
- A pretrained U-Net diffusion model with frozen weights ψ serves as a 'teacher' for guiding the student model, which produces samples $\tilde{x}_\theta$ that are further processed to the teacher model to generate $\tilde{x}_\psi$ samples [4].
- The student model's objective function is a combination of distillation loss (between student and teacher model samples) and adversarial loss (between student and real samples) [4]

$$\mathcal{L} = \mathcal{L}_{\text{adv}}^{\text{G}}(\hat{x}_\theta(x_s, s), \phi) + \lambda \mathcal{L}_{\text{distill}}(\hat{x}_\theta(x_s, s), \psi)$$

In summary, this algorithm equips the student model to accurately emulate both real samples and those generated by the teacher model, but with significantly fewer denoising steps.

Key contributions and innovations

The approach introduces two key innovations:

- The trained model requires only a small number of sampling steps, enabling the generation of high-quality images at near-real-time speeds.
- A combination of two loss functions, distillation loss and adversarial loss, trains the model using both real images and the teacher model. This strategy allows the student model to effectively learn from the teacher model while achieving the high-speed performance characteristic of GAN models.

Strengths and limitations.

Strengths

- The model can generate images in real-time.
- It is capable of producing high-quality images; for instance, the 4-sample ADD-XL outperforms its teacher, SDXL, at a resolution of 512 x 512 [4].
- ADD significantly outperforms other baseline approaches, such as one-step GAN, LCM, and others [4].

Disadvantages

- The authors note a slight decrease in sample diversity with ADD-XL [4].

Performance and benchmark

Image generation

Samples of the ADD-XL model, trained on Stable Diffusion XL, are displayed in Figure 9. The analysis of the samples reveals:

- With just 1 sampling step, ADD-XL achieves acceptable results. Although some samples contain anomalies (e.g., an eagle with two beaks), the overall image quality is impressive.
- The samples with 2 and 4 sampling steps demonstrate excellent outcomes. Anomalies are not observed, and the model exhibits strong capabilities in abstraction.
- The model encounters some difficulties in accurately generating images for the first prompt.

Figure 9. Samples for ADD-XL with 1, 2, and 4 sampling steps [4]

Performance

The model's performance compared to baseline image generators is depicted in Figure 10. Generation speed, as well as FID and CLIP scores for ADD and other distillation models, are presented in Table 3. Analysis of both the figure and the table indicates:

- 1 step ADD-XL outperforms all other models except it's teacher SDXL with 50 steps.
- 4 steps ADD-XL provides great boost for both model image quality and prompt alignment.1 step ADD has great generation speed of 0.09 seconds.
- Being one of the fastest, ADD also shows the best results in FID and CLIP in relation to other distillation models.

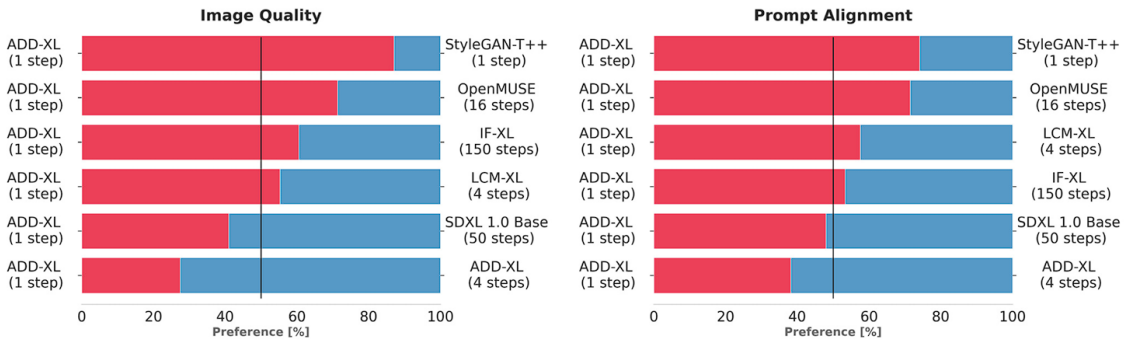

Figure 10. Performance metrics for ADD-XL comparing to baseline models [4]

Method	#Steps	Time (s)	FID ↓	CLIP ↑
DPM Solver [37]	25	0.88	20.1	0.318
	8	0.34	31.7	0.320
Progressive Distillation [43]	1	0.09	37.2	0.275
	2	0.13	26.0	0.297
	4	0.21	26.4	0.300
CFG-Aware Distillation [31]	8	0.34	24.2	0.300
InstaFlow-0.9B [36]	1	0.09	23.4	0.304
InstaFlow-1.7B [36]	1	0.12	22.4	0.309
UFOGen [71]	1	0.09	22.5	0.311
ADD-M	1	0.09	**19.7**	**0.326**

Table 3. Distillation models comparison [4]

Applications

Stability AI has made SDXL-Turbo publicly available, an ADD model that has been trained on the Stable Diffusion XL dataset:

Link to GitHub: https://github.com/Stability-AI/generative-models

Link to models: https://huggingface.co/stabilityai/sdxl-turbo

Link to demo: https://clipdrop.co/stable-diffusion-turbo

Control Net

A concise overview of the architecture

ControlNet is an architecture designed to introduce spatial conditioning into pre-existing text-to-image diffusion models [5]. It supports various conditions, including sketches, normal maps, depth maps, canny edges, and several others [5].

This architecture was developed by Lvmin Zhang, Anyi Rao, and Maneesh Agrawala, as detailed in their work 'Adding Conditional Control to Text-to-Image Diffusion Models'. Figure 11 illustrates the abstract representation of the ControlNet algorithm.

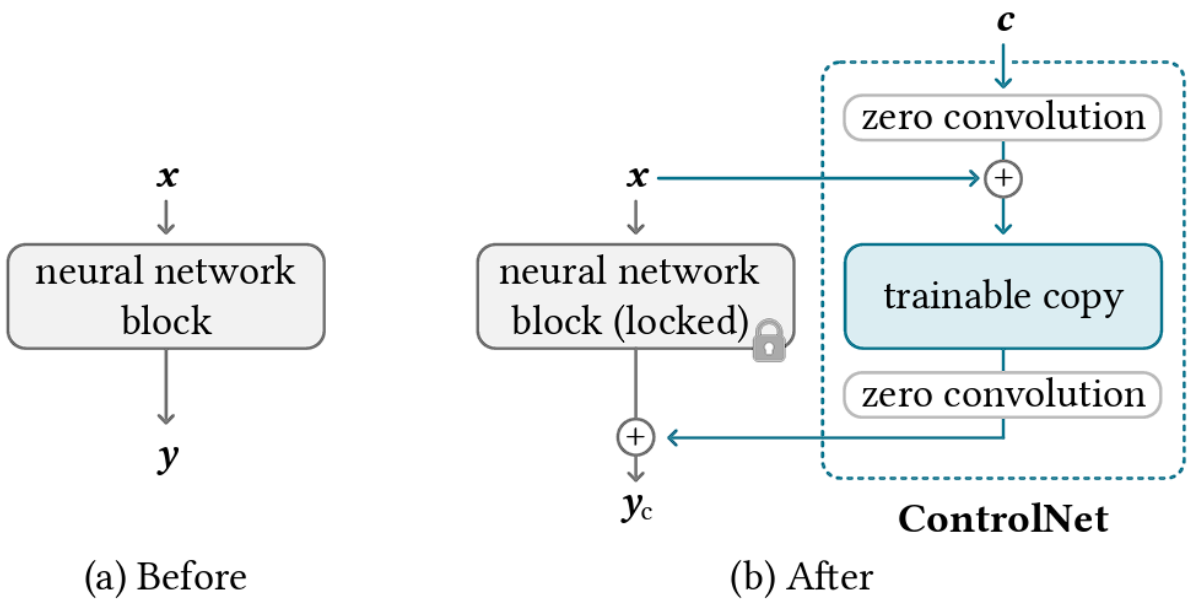

Figure 11. ControlNet algorithm [5]

The algorithm integrates additional conditional information into a pre-trained neural network block by creating a trainable duplicate of the original [5]. The process is as follows [5]:

- The original neural network block, $y = f(x; \Theta)$, with weights Θ, processes the feature map $x \in \mathbb{R}^{h \times w \times c}$, transforming it into feature map y;
- ControlNet freezes the weights Θ and generates an exact, trainable copy with weights Θ_c;
- This copy accepts a vector c as the condition input, which is processed alongside the feature map x through a zero-convolution layer;
- The output from the copy, after passing through another zero-convolution layer, is merged with the output from the original block;
- The resulting y_c is the new feature map, enhanced with the applied conditions;

The equation for the full process is following [5]:

$$\boldsymbol{y}_c = \mathcal{F}(\boldsymbol{x}; \Theta) + \mathcal{Z}(\mathcal{F}(\boldsymbol{x} + \mathcal{Z}(\boldsymbol{c}; \Theta_{z1}); \Theta_c); \Theta_{z2})$$

The application of this algorithm to the U-Net layers of Stable Diffusion is depicted in Fig. 12.

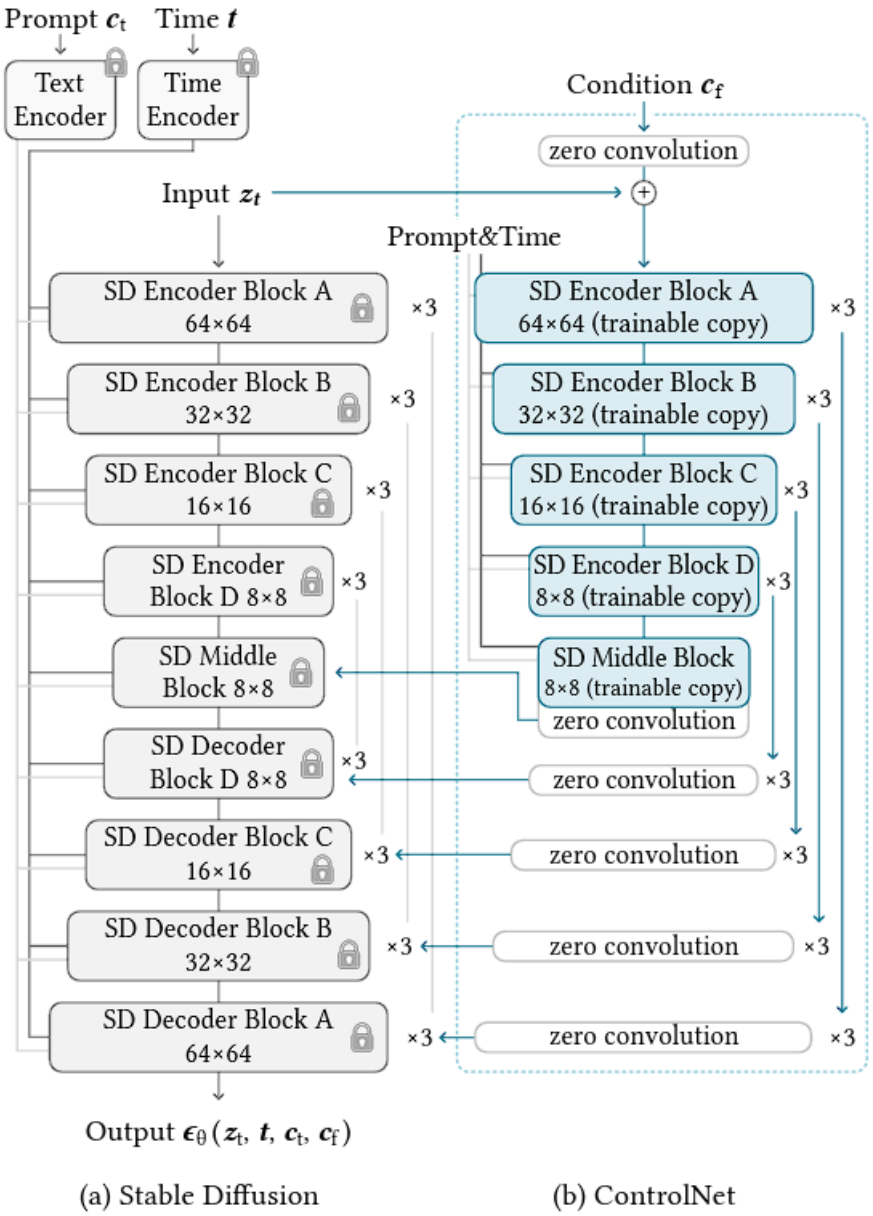

Figure 12. Stable Diffusion U-Net architecture with ControlNet [5]

Key contributions and innovations

The algorithm enables the retraining of state-of-the-art diffusion models to work with inputs beyond text prompts. Consequently, users gain significantly more control over the generated outputs, employing conditions such as sketches, human poses, depth maps, and others to achieve the precise results they desire.

Another innovation is the incorporation of zero-convolution layers in the trainable copies. According to the authors' research, these layers allow the copies to effectively utilize all the knowledge from their originals while filtering out harmful noise, as demonstrated in Figure 14(a) [5].

Strengths and limitations.

Strengths

- Capable of interpreting various content types, as demonstrated in Figure 13.
- Utilizes pre-trained text-to-image models without altering the original model's structure.

- Exhibits strong transferability to community-developed models, showing excellent compatibility with Stable Diffusion, Comic Diffusion, and others [5].

Limitations

- Slight decrease in the FID scores of original models after retraining.

Performance and benchmark

Image generation

The samples from the ControlNet model, integrated with Stable Diffusion, are displayed in Figure 13.

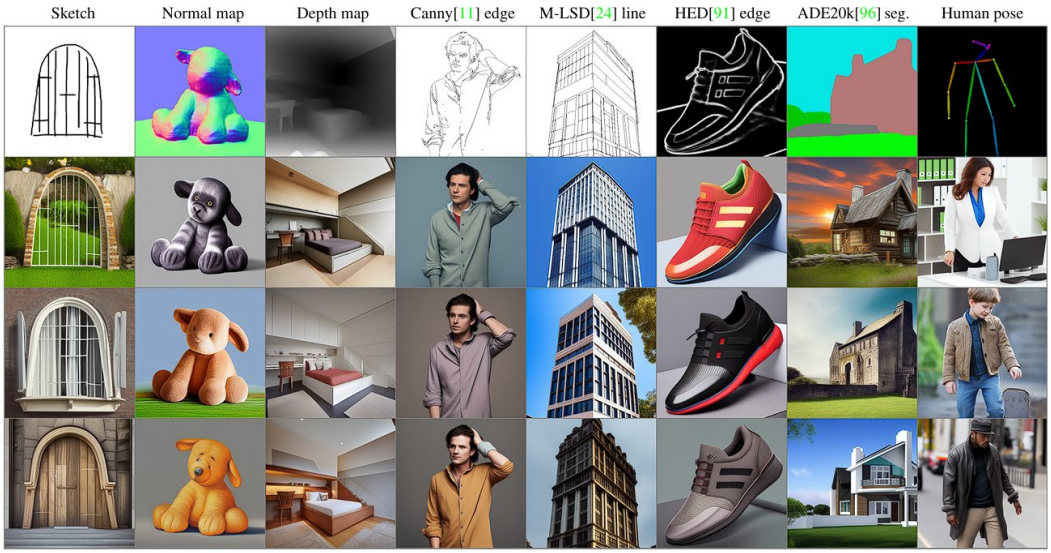

Figure 13. Stable Diffusion with ControlNet: Samples with various conditions, without text prompts [5]

- The model excels at creating comprehensive images from abstract sketches.
- It is capable of seamlessly merging different segments into a single, harmonious image.
- The model effectively transforms human poses into realistic images.
- It also demonstrates strong performance with other conditions.

Fig. 14 illustrates the model's performance across different prompts and architecture implementations, clearly indicating that the inclusion of zero convolution layers significantly enhances model performance.

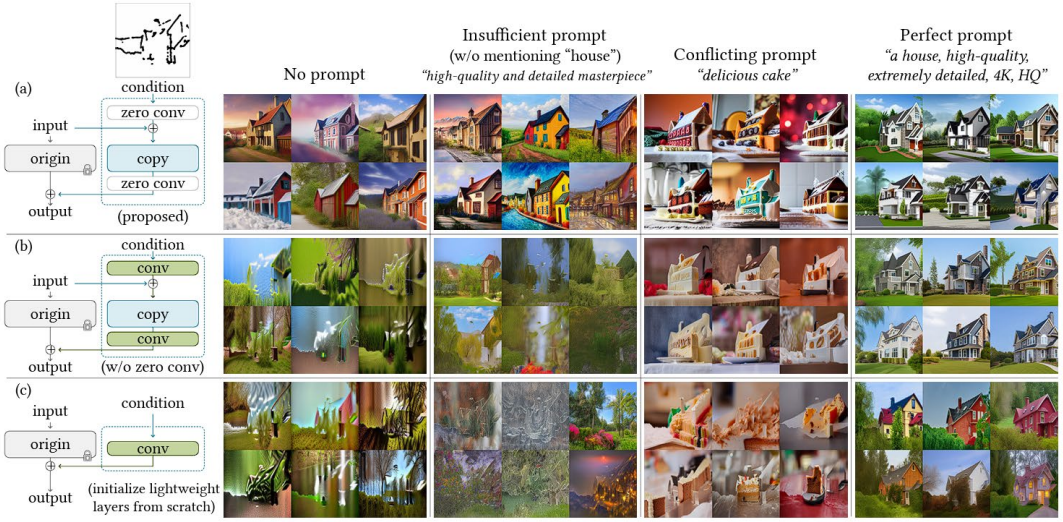

Figure 14. ControlNet with and without prompts and different algorithms applications [5]

Performance

The model's performance, in comparison to other baselines, is presented in Table 4:

Method	FID ↓	CLIP-score ↑	CLIP-aes. ↑
Stable Diffusion	6.09	0.26	6.32
VQGAN [19](seg.)*	26.28	0.17	5.14
LDM [72](seg.)*	25.35	0.18	5.15
PITI [89](seg.)	19.74	0.20	5.77
ControlNet-lite	17.92	0.26	6.30
ControlNet	15.27	0.26	6.31

Table 4. ControlNet performance [5]

ControlNet exhibits superior performance, significantly outperforming other baseline approaches. The only exception is Stable Diffusion without conditions, which achieves better FID scores.

Applications

The authors have developed and provided several ControlNet models that are based on Stable Diffusion version 1.5:

Link to GitHub: https://github.com/lllyasviel/ControlNet

Link to models: https://huggingface.co/lllyasviel/ControlNet

Link to demo: https://huggingface.co/spaces/hysts/ControlNet

Conclusions

The exploration of these diffusion models represents a substantial advancement in the field of image generative AI. While the first generation of diffusion models was characterized by slow processing speeds and significant computational resource requirements, newer versions have shown remarkable improvements in efficiency. The Latent Space Diffusion model dramatically reduces computational demands, and Adversarial Diffusion Distillation enables the creation of models capable of operating in real-time. Further advancements in output control have been realized through ControlNet, which allows for precise manipulation of diffusion model outputs.

The future of image generative AI appears bright and promising. With the current pace of development, it is only a matter of time before these technologies are integrated into every aspect of our lives.

REFERENCES

[1] Yang, Ling, et al. "Diffusion models: A comprehensive survey of methods and applications." ACM Computing Surveys 56.4 (2023): 1-39.

[2] Chang, Ziyi, George A. Koulieris, and Hubert PH Shum. "On the design fundamentals of diffusion models: A survey." arXiv preprint arXiv:2306.04542 (2023).

[3] Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.

[4] Sauer, Axel, et al. "Adversarial diffusion distillation." arXiv preprint arXiv:2311.17042 (2023).

[5] Zhang, Lvmin, Anyi Rao, and Maneesh Agrawala. "Adding conditional control to text-to-image diffusion models." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023.