1. What is the difference between an auto-encoder, a generative adversarial network (GAN), and a diffusion model? Consider: (1) model structure; (2) optimized objective functions, and (3) how different components of each models are trained. (9 % of CW2) (Renee)

Aspect	**Auto-Encoder**	**Generative Adversarial Network (GAN)**	**Diffusion Model**
Model Structure	Consists of a decoder and an encoder. While the decoder reconstructs the input from this representation, the encoder compresses the input data into a latent representation.[3][4] [Fig 1.]	Consists of a discriminator and a generator. While the discriminator attempts to discern between produced and genuine data samples, the generator creates data samples from random noise.[1] [Fig 1.]	Entails simulating how noise levels change over time. In order to get to the next state in the diffusion process—which eventually produces realistic data—each step can be understood as adding noise to the preceding state.[6] [Fig 1.]
Optimized Objective Functions	Reconstruction loss is usually minimised between the input and the reconstructed output, for example, by minimising Mean Squared Error (MSE) or Binary Cross-Entropy (BCE).[3][4]	Reduces an objective function with a min-max. The discriminator's goal is to accurately discriminate between actual and created samples, while the generator's goal is to produce samples that are identical to real data in order to trick the discriminator. [1]	Reduces the observed data's negative log-likelihood under the diffusion process. Modelling the probability distribution of data points given the noise levels is the aim.[6]
Training Components	Gradient descent and backpropagation are used in tandem to train the encoder and decoder. To update the parameters, the reconstruction error is backpropagated throughout the network.[3][4]	An adversarial training approach is used for the discriminator and generator. Based on the discriminator's feedback, the generator is adjusted to produce more realistic samples, and vice versa. [1]	During training, data points and noise levels are conditioned on one another. Data points are produced by repeating the diffusion process with noise levels sampled from an earlier distribution.[7][6]

2. The UNet is one of the most important components of a diffusion system because it facilitates the actual diffusion process. Consider an unconditional diffusion model that has a UNet model with the following parameters: (Mouneer)
   - Input size: 512 * 512

- Number of channels (n_channel): 128 (This is the number of channels in the initial feature map that we transform the image into)
- Channel multipliers (ch_mults): 1, 2, 4 (This is the list of channel numbers at each resolution. The number of channels is (ch_mults[i] * n_channels at layer i)
- Up/Down sampling factors: 2
- Number of up/down blocks: 3

(1) Please write down the dimensions of the intermediate feature maps after each of the following UNet blocks in turn: the 3 downsample blocks, the 1 middle block, the 3 upsample blocks.

(2) Within UNet, there are attention modules and time-step embeddings. Briefly describe how they integrate with UNet.

Hint:Input dimension of UNet: [512,512,128]

Output of the first Downsample Block : [256,256,128]

You can use the UNet image in the slides to help you solve the problem. (14% of CW2)

Answer:

Part A:

Input dimension of Unet : [512,512,128]

Downsampling path::

Output of the first Downsample Block : [256,256,128]

Output of the second Downsample Block : [128,128,256]

Input of the Third Downsample Block : [128,128,256]

Middle blocks:

Output of the Downsample Middle Block : [64,64,512]

Input of the Upsample Middle Block : [512,64,64]

Upsample path:

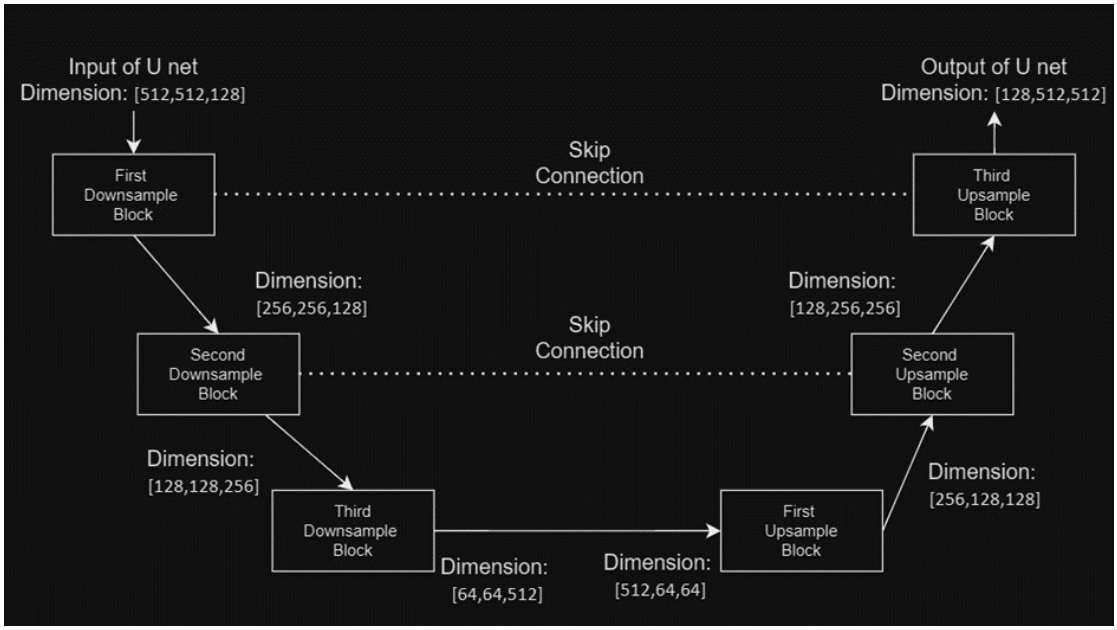
Output of the first Upsample Block: [256,128,128]

Output of the second Upsample Block: [128,256,256]

Output of the third Upsample Block: [128,512,512]

Final output of the Unet: [128,512,512]

For better visualisation, please refer to the following diagram:



Part B:

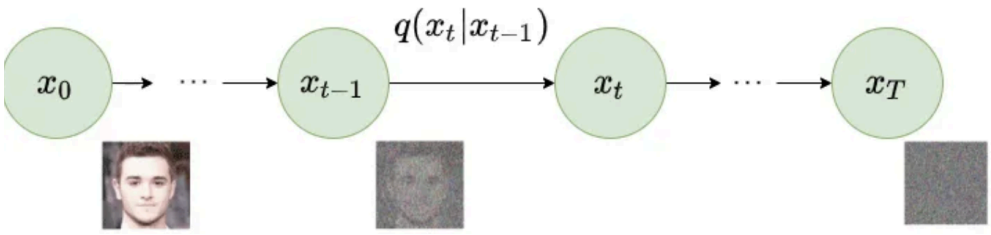
Proposed by Oktay et al, the concept of attention was introduced in the U-Net to focus on or segment of target structures of any size and shape. The model known as Attention Gate (AG), makes use of soft attention in which image features are assigned continuous weights reflecting their relevance. These weights are learned during training, enabling the model to dynamically prioritise informative regions while still processing the entire image. While upsampling, the skip connections present in the U-Net merge the latent space from downsampling to reconstruct more precise spatial information. One issue that remains is that redundant features are brought through the skip connection. The soft attention is hence implemented in the skip connection of the UNet which enhances activation in only regions of interest and thereby lowers the amount of low-level features that are added. [10]

Another issue of the U-Net is that it has no inherent mechanism for positioning. That is, it cannot evaluate which noise step is being decoded. However, for a diffusion model to produce accurate results, the amount of noise predicted depend on the timestep. For example, predicted noise will have lower intensity as the end result is being produced. To solve this issue, time step embeddings are defined within the U-Net itself which was inspired from the positional embedding from the Transformer model [11].

3. If an image undergoes a forward process of noise addition and then a sampling process of denoising, (1) would the resultant image remain identical to the original? (2) Why or why not? (3) How would the results differ with more noising and denoising steps?(5 % of CW2) (Tian Xiu)

Answer: The resulting image will not be identical to the original image. Even advanced denoising techniques struggle to completely restore the original image after noise has been added. Because the noise addition process introduces randomness into the original image, which may lead to the loss of some information. Although the denoising process aims to restore the original image, due to the loss of information and the imperfections of the denoising model, the denoising process may not be able to precisely recover all details[6].
We can see it from the figure below. The forward process of diffusion accumulates Gaussian noise T times. As t increases, x_t tends to approach pure noise. The time step is designed to simulate a gradually increasing perturbation process over time. Each time step represents one

perturbation process, starting from an initial state and gradually altering the distribution of the image through multiple applications of noise. Therefore, smaller steps represent weaker noise disturbances, while larger ones represent stronger noise disturbances. During the training process, loss will gradually decrease, and the change in loss will become smaller later in the training process. So increasing the number of noise addition and denoising steps may lead to greater differences between the result and the original image. Each noise addition and subsequent denoising process can result in the accumulation of information loss and estimation errors. If the noise level is particularly high, certain details of the original image may be completely obscured, making it difficult for the denoising process to recover these details. Therefore, a complex model architecture and a large amount of training data are required.[7]



4. Samplers are one of the key components in Diffusion models. Answer the following question about the sampler: (1) What sampler was used for training in this coursework? (2) What is the main difference between DDPM and DDIM (from the aspect of stochastic) and what are the benefits of DDIM over DDPM? (7 % of CW2) (Renee)
   Answer: The sampler used for training in this coursework is the CustomDDPMScheduler which uses the Langevin dynamics sampler. [7]

When it comes to stochastic aspects, the primary distinction between the diffusion implicit model (DDIM) and the diffusion probabilistic model (DDPM) is in how they handle sampling during training.

By first adding Gaussian noise to the data and then using Langevin dynamics to progressively diffuse the noise away, samples are generated using Langevin dynamics sampling, which is used in DDPM. This entails applying the discretized Euler-Maruyama method to solve a stochastic differential equation (SDE) [7].
On the other hand, DDIM employs a less complex sampling process. DDIM uses a sequence of invertible transformations to create samples from noise by directly sampling it from a fixed distribution (usually a Gaussian distribution) as opposed to Langevin dynamics. As a result, training is quicker and more scalable because the computational expense of solving SDEs is avoided [8].

The benefits of DDIM over DDPM include:

- Efficiency: Compared to DDPM, which uses Langevin dynamics, DDIM's sampling process is computationally more efficient. This improves DDIM's scalability to huge datasets and speeds up training durations [8].
- Simplicity: Compared to Langevin dynamics, which requires solving stochastic differential equations, the sampling process used in DDIM is more straightforward and easier to put into practice. Because of its simplicity, academics and practitioners may find DDIM easier to use [8].
- Scalability: DDIM may be more suited for scaling up to high-dimensional data or large-scale generative modelling tasks because of its simplicity and computing efficiency [8].

In general, DDIM is a more effective and scalable substitute for DDPM, which makes it a desirable option for generative modelling projects when scalability is an issue or computational resources are scarce.

References

1. I. Goodfellow et al., “Generative Adversarial Nets,” Jun. 2014. Available: https://arxiv.org/pdf/1406.2661.pdf
2. A. Gainetdinov, “GAN Mode Collapse explanation,” Medium, Mar. 07, 2023. https://medium.com/towards-artificial-intelligence/gan-mode-collapse-explanation-fa5f9124ee73 (accessed Mar. 15, 2024).
3. D. Kingma and M. Welling, “Auto-Encoding Variational Bayes,” 2014. Available: https://arxiv.org/pdf/1312.6114.pdf
4. J. Rocca, “Understanding Variational Autoencoders (VAEs),” Medium, Mar. 15, 2020. https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73
5. J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, S. Ganguli, and S. Edu, “Deep Unsupervised Learning using Nonequilibrium Thermodynamics,” Nov. 2015. Available: https://arxiv.org/pdf/1503.03585.pdf
6. L. Weng, “What are Diffusion Models?,” lilianweng.github.io, Jul. 11, 2021. https://lilianweng.github.io/posts/2021-07-11-diffusion-models
7. J. Ho, A. Jain, and P. Abbeel, “Denoising Diffusion Probabilistic Models,” Dec. 2020. Available: https://arxiv.org/pdf/2006.11239.pdf
8. J. Song, C. Meng, and S. Ermon, “Published as a conference paper at ICLR 2021 DENOISING DIFFUSION IMPLICIT MODELS.” Available: https://arxiv.org/pdf/2010.02502.pdf
9. M. Suginoo, “Mini-Max Optimization Design of Generative Adversarial Networks (GAN),” Medium, Jan. 14, 2024. https://towardsdatascience.com/mini-max-optimization-design-of-generative-adversarial-networks-gan-dc1b9ea44a02#:~:text=Therefore%2C%20GAN%20has%20two%20objective (accessed Mar. 15, 2024).
10. Oktay, O. *et al.* (2018) *Attention U-net: Learning where to look for the pancreas*, *arXiv.org*. Available at: https://arxiv.org/abs/1804.03999 (Accessed: 18 March 2024).
11. Young, B. (2023) *A gentle introduction to diffusion*, *W&B*. Available at: https://wandb.ai/byyoung3/ml-news/reports/A-Gentle-Introduction-to-Diffusion---Vmlldzo2MzgxNjc3 (Accessed: 20 March 2024).

APPENDIX

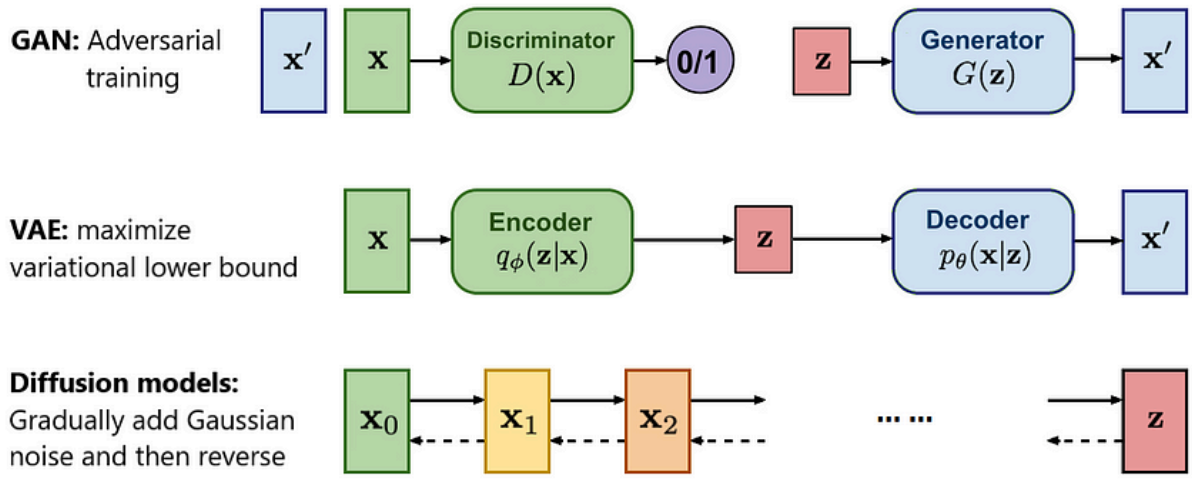


Fig.1.Overview of different types of generative models.