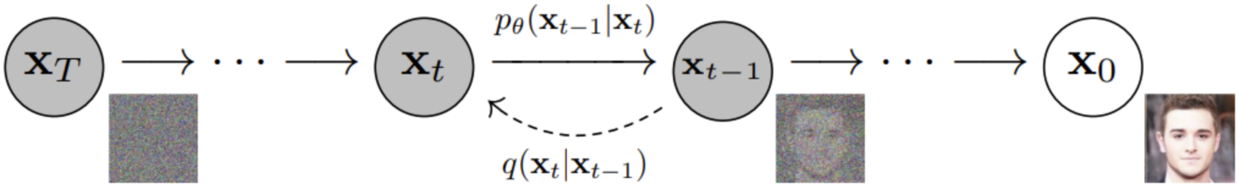
Image Synthesis with Diffusion Model

Paper: **High-Resolution Image Synthesis with Latent Diffusion Models** (**arXiv:2112.10752**)
Group Name: Berkeley Goggles
Members: Jun Tian, Jihoon Yang, Aayush Gupta, Cathie Lin
SID: 3038738146, 3038270198, 3036609229, 3036879142

0. Abstract

In the past year, AI has made significant breakthroughs and developments in the field of AIGC, which could not have been possible without the renowned stable diffusion model's contributions to the open-source community. Stable diffusion is a text conditional latent diffusion model, created by researchers and engineers from CompVis, Stability AI, and LAION. As it utilizes latent space, it greatly reduces the computational resource requirements for training, making generative models widely studied. The latent diffusion model not only speeds up training time but also performs competitively against state-of-the-art models for tasks such as inpainting, super-resolution, text-to-image synthesis, and more. Since stable diffusion is essentially a form of diffusion model, our project will mainly explore how to use diffusion models to generate images.

The key idea of the diffusion model is to break down the generation process into iterative denoising steps. They work by taking as input an image x_0, successively adding Gaussian noise to it, and then learning how to recover the original input by reversing this noising process. Once trained, a diffusion model can then generate new images by running this learned denoising process on randomly sampled noise.


$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$
$\mathbf{x}_{t-1}$
$\mathbf{x}_0$

Our main goal with this project is to have students become familiar with diffusion models. This includes becoming familiar with what the diffusion process actually is, how we add noise to our images (with a scheduler), how we denoise images (through Unets), and how the Unet itself is built. More specifically, we can divide our project into three parts:

- Part 1: Build different modules to facilitate the construction of Unet models, this will help students understand the architecture of the denoising Unet model.
- Part 2: Implement a q-sample algorithm and apply it on a single image. This will help students understand the forward noise-adding process in the diffusion algorithm.
- Part 3: Train a simple diffusion model on the MNIST dataset, which will give students a closer perspective of how the diffusion model is trained

1. Review Comments

Here is the copy of reviewer comments.

1.1 Reviewer 1

Question 1: Content and Correctness (Option 1). If this paper is not Option 1, write NA.

Answer: This paper very thorough with making students implement the different UNet components as well as the actual training process. The code runs smoothly and efficiently.

Question 1 - grade

Answer: Excellent work, no actions needed.

Question 2: Scaffolding (Option 1). If this paper is not Option 1, write NA.

Answer: This paper includes very detailed descriptions, diagrams, and equations to provide adequate context for each step. This makes the homework self-contained.

Question 2 - grade

Answer: Excellent work, no actions needed.

Question 3: Readability/Clarity (Option 1). If this paper is not Option 1, write NA.

Answer: The code and accompanying descriptions are very well structured. The accompanying code runs without any bugs.

Question 3 - grade

Answer: Excellent work, no actions needed.

Question 4: Commentary on HW (Option 1). If this paper is not Option 1, write NA.

Answer: The commentary provides adequate context to the paper that this homework is based off of in a very concise and structured way. This commentary is very detailed, but I do wish that some of the terminology could be better explained.

Question 4 - grade

Answer: Small improvement needed

Question 5: Going above and beyond (Option 1). If this paper is not Option 1, write NA.

Answer: This paper has very solid visualization and equations that help me understand what the goal of the model is.

Question 5 - grade

Answer: Excellent work, no actions needed.

1.2 Reviewer 2

Question1: Content and Correctness (Option 1). If this paper is not Option 1, write NA.

Answer: The notebook successfully manages to put together a lot of content and engages with them in a systematic way. Each block individually has proper scaffolding and can be solved in a reasonable amount of time. There was a good amount of einops rearrange which is not really the point of the course, more of a code issue, but its pretty useful for practical applications of ML so its good to get practice with that. Since it's not really taught how to use it though, they take more time to solve. The first autograder question didn't work, the resnet one. I copied the code from the solutions and ran it and didn't pass. Overall content is good though.

Question 1 - grade

Answer: Small improvement needed

Question 2: Scaffolding (Option 1). If this paper is not Option 1, write NA.

Answer: The scaffolding is good, they provide a good amount of hints which helps a lot. Without them, the problem would be quite difficult. Also, the text above those cells with the algorithms and explanations of whats happening is good for the most part. There is one large code cell with titled "Build Your Spatial Transformer" which I think is too long, its hard to fill out around 8 "todos" without any breaks and any explicit text explanations of whats happening.

Question 2 - grade

Answer: Small improvement needed

Question 3: Readability/Clarity (Option 1). If this paper is not Option 1, write NA.

Answer: Yes, clear and easy to follow. Again, that one code block "Build Your Spatial Transformer" is quite long so the readability goes down for that part. Splitting that up would be good.

Question 3 - grade

Answer: Small improvement needed

Question 4: Commentary on HW (Option 1). If this paper is not Option 1, write NA.

Answer: Commentary is good, I have sufficient background knowledge of what's going on with the commentary and information from the class.

Question 4 - grade

Answer: Excellent work, no actions needed.

Question 5: Going above and beyond (Option 1). If this paper is not Option 1, write NA.

Answer: Pretty similar to the paper, not much above and beyond. The last part where they modified the training set to use MNIST was good and helped provide visualizations of whats happening.

Question 5 - grade

Answer: Excellent work, no actions needed.

1.3 Reviewer 3

Question 1: Content and Correctness (Option 1). If this paper is not Option 1, write NA.

Answer: The coding part can smoothly run.
However, the written part is obvious and trivial; diffusion model is highly related to the statistics.
Therefore, if there are more analytical problems it will be better.

Question 1 - grade

Answer: Small improvement needed

Question 2: Scaffolding (Option 1). If this paper is not Option 1, write NA.

Answer: This project does an excellent job of providing a solid foundation for students to understand and engage with the material. All code that is required for the project is explained in a clear and concise manner through text cells in Jupyter and/or code comments, ensuring that students can understand what is expected of them.

Additionally, the project provides all necessary external packages or clear instructions on how to download them before starting the assignment. This level of support ensures that students can focus on the task at hand without being hindered by technical difficulties.

Furthermore, the availability of autograding tests and sanity checks ensures that students can test their code and receive immediate feedback on its correctness. This helps to reinforce their learning and promotes a deeper understanding of the material.

Overall, this project is self-contained and provides students with all necessary materials, making it an excellent resource for anyone looking to learn and engage with the material.

Question 2 - grade

Answer: Excellent work, no actions needed.

Question 3: Readability/Clarity (Option 1). If this paper is not Option 1, write NA.

Answer: I must say, the readability and clarity of your HW assignment and commentary are impressive. The content is easy to comprehend and follow, and any mathematical notations used are well-defined and understandable. Your use of non-standard notations is also commendable, as you have taken the time to explain them clearly. Furthermore, I did not come across any spelling or grammar errors while reviewing the assignment and commentary.

As for the code, I did not detect any bugs during my review. The code runs smoothly, and I did not encounter any issues while testing it.

Question 3 - grade

Answer: Excellent work, no actions needed.

Question 4: Commentary on HW (Option 1). If this paper is not Option 1, write NA.

Answer: The commentary on the HW is well-written and very helpful for understanding the key concepts in the paper. The 3 page commentary provides a clear and concise explanation of how the assignment engages with these concepts, which is useful for students who may not have a strong background in the subject matter.

In addition, the commentary also highlights the most important points from the paper and provides context for the assignment, which helps students to see the bigger picture and understand why the concepts are important.

Overall, I believe that these commentaries are very helpful and achieve the learning goals of the HW. They provide students with a deeper understanding of the material and help them to apply the concepts they have learned to real-world problems.

Question 4 - grade

Answer: Excellent work, no actions needed.

Question 5: Going above and beyond (Option 1). If this paper is not Option 1, write NA.

Answer: The task is MNIST, a rather small dataset, and the trained model doesn't perform well. You can further think of how to train a better task with more data and high training speed. This is difficult for a big model, but this also test your understanding of the model if you can simplify and get really good answer.

Question 5 - grade

Answer: Medium improvement needed

1.4 Reviewer 4

Question 1: Content and Correctness (Option 1). If this paper is not Option 1, write NA.

Answer: Coding question concepts are useful in expressing the ideas from the paper, any additional context and background or equations that are needed are expressed well. However, there are a few small things to note however

- I think its too long for 2 hours of work given the sheer number of coding todos
- In one of the coding cells, when ran it shows "The relative error for resnet is 0.02200488932430744, should be smaller than 0.001", which may not be accurate
- A small typo "timesepts 1 to T."

However, this is a very well prepared homework apart from the length.

Question 1 - grade

Answer: Small improvement needed

Question 2: Scaffolding (Option 1). If this paper is not Option 1, write NA.

Answer: This homework is self-contained and well supported by diagrams and mathematical explanations. A small comment I would make is that it would be helpful to have additional assert statements in order to show what values are expected rather than just printing it out. The coding sections could also be a bit more split up, rather than continuous todos having to be implemented so students can checkpoint their work halfway and verify that it is correct

Question 2 - grade

Answer: Small improvement needed

Question 3: Readability/Clarity (Option 1). If this paper is not Option 1, write NA.

Answer: The notebook runs smoothly, and has excellent grammar and exposition. I don't believe improvement is needed

Question 3 - grade

Answer: Excellent work, no actions needed.

Question 4: Commentary on HW (Option 1). If this paper is not Option 1, write NA.

Answer: Small latex style problems with the attention equation. I like the constant comparison with the paper and reference to the exercises that the students are supposed to implement as well as rationale given for the architectural choices made in light of the constraints

Question 4 - grade

Answer: Excellent work, no actions needed.

Question 5: Going above and beyond (Option 1). If this paper is not Option 1, write NA.

Answer: It was a great idea to include the visualization gif! I think it really hammers home the idea of what the model is actually doing and how it learns.

Question 5 - grade

Answer: Excellent work, no actions needed.

3. Response to the Review

3.1 To Reviewer 1

Content and Correctness, *No actions needed.*

Scaffolding, *No actions needed.*

Readability/Clarity, *No actions needed.*

Commentary, *Small Improvements Needed*: They remarked that some of the commentary in the paper could be better explained but did not explain which terminology they thought could be better explained. Therefore, we polished the report as a whole, making its description more academic and easier to understand.

Going Above and Beyond, *No actions needed.*

3.2 To Reviewer 2

Content and Correctness, *Small Improvements Needed*: The reviewer reported that the first autograder question did not work which was fixed in our final submission.

Scaffolding, *Small Improvements Needed*: The reviewer reported that one large code cell titled "Build Your Spatial Transformer" was very long and hard to get through as there were little to no breaks or text explanation accompanying that section of code. We addressed this by splitting the coding questions into four classes and only requiring students to implement the FeedForward Layer and CrossAttention Layer in the final submission, which will be much easier to figure out. What's more, we also reduced the difficulty of the rest of the code questions, keeping the overall completion time of the task within 1-2 hours.

Readability/Clarity, *Small Improvements Needed:* Again, reviewer reported that the "Build Your Spatial Transformer" was too long and reduced readability. We split up that section to increase readability.

Commentary, *No actions needed.*

Going Above and Beyond, *No actions needed.*

3.3 To Reviewer 3

Content and Correctness, *Small Improvements Needed:* The reviewer reported that the written part was too trivial and wished to see more analytical problems. We added a few analytical questions asking students to identify the performance of various LDM-f models and also to write the loss of conditional LDM models. We also added another question that asked students about ELBO loss to show how maximizing the negative ELBO minimizes KL divergence. Since we reduced the difficulty and length of the coding portion, we believe that the additional written questions will still be completable within the 2 hour timeframe.

Scaffolding, *No actions needed.*

Readability/Clarity, *No actions needed.*

Commentary, *No actions needed.*

Going Above and Beyond, *Medium Improvement Needed*: The reviewer reported that the trained model does not perform well. Indeed, this is a problem faced by current models, which we also address in the questions in the writing section. On the one hand, we can make further optimizations, but that will be beyond the scope of this paper, and the time for students to complete their homework will also be greatly increased. On the other hand, additional datasets and larger networks increase the memory requirements, which is not realistic since we need every student to be able to complete the assignment. Therefore, after comprehensive consideration, we did not try to train a better model, but tried to let students have a more vivid understanding of the diffusion model through better visualization schemes and code questions.

3.4 To Reviewer 4

Content and Correctness, *Small Improvements Needed:* The reviewer reported that the coding sections were too long to be completed within two hours, so we removed a few of the TODOs in the "Build Your Spatial Transformer" portion (See response to Reviewer 2). One of the coding cells incorrectly reported the relative error as being less than 0.001 when it was not so we fixed this issue. We also fixed minor spelling typos in the Jupyter Notebook.

Scaffolding, *Small Improvements Needed:* The reviewer wished to see additional assert statements to show what values are expected in the code. This is a good suggestion, but since our tests are mainly on network modules, the input and output are mainly tensors. Printing tensors directly in the code is not very beautiful, so we save all the data we need to judge in github. If students wanted, we advise them to print the y values in the test to see what to expect.
Again, the reviewer wished to see some of the longer coding sections being split up. As addressed above, we split up these sections into smaller sections by providing some additional code in some of the TODOs and removed some of the TODOs.

Readability/Clarity, *No actions needed.*

Commentary, *No actions needed:* The reviewer noted that there are some minor issues with the LaTeX of the attention equation used in the commentary so we fixed these issues.

Going Above and Beyond, *No actions needed.*

4. Final Version of Our Project

In this section, We'll present our final submission. Due to limited space, we only added the revised commentary and writing assignment here. For coding questions, please see section 6.

4.1 Project commentary

As described in the abstract, our main goal with this project is to have students become familiar with diffusion models. This includes becoming familiar with what the diffusion process actually is, how we add noise to our images (with a scheduler), how we denoise images (through Unets), and how the Unet itself is built.

Our project can be divided into two assignments, including the writing assignment and coding assignment. In the writing part, we asked students to briefly read the paper and ask questions about the ELBO loss function of the LDM model and the comparative experiments of different scaling factors. Through the VAE ELBO loss function, we have students gain better intuition for what maximizing the ELBO does and how it is useful. We believe that through these problems, students can first have a rough understanding of the latent diffusion model.

Then, in the coding assignment, we ask the students to implement different parts of the diffusion model, and train a small network for visualization. More specifically, we can divide our project into three parts:

Part 1: Implement Modules in the Diffusion Model

The denoising Unet is foundational for a diffusion model, so we guide the students through modules necessary to understand and build the Unet, which includes the ResNets Blocks, Spatial Transformer Blocks, and Unet itself.

ResNet Module

The Unet model that we present in our homework is built off of ResNet blocks which are used to help with the downsampling and upsampling process. The skip connections in ResNets help us deal with vanishing gradients when we increase the number of convolutional layers in our model. It is worth noting that the ResNet we used here is not the same as the ResNet we learned in lecture before because we combine timestep information. To do this, we need to scale-shift the parameters. What’s more, in order to facilitate students' implementation, we also drew a model architecture diagram.

Text Embeddings

Since the latent diffusion model introduces new forms of input including text inputs which gives us the power to perform text-to-image synthesis, we need a way to turn the text input into a representation that the diffusion model can work with. The authors use a domain-specific encoder, $\tau\theta$, to transform inputs into a representation that the cross-attention layers use for the keys and values. For text inputs, the authors use the BERT tokenizer to help prepare the text-to-image inputs and produce the representation. The text is read and turned into a vector representation per token. For the homework we use the CLIP transformer encoder instead, specifically the CLIPTokenizer as CLIP has been used for prior diffusion models. We have this implemented for the students, but explain the usage of CLIP for text embeddings and provide the code so they can see how the process works.

Attention Module

In the paper's diffusion model, the author uses spatial transformers to combine text information with image features. This is achieved by using cross-attention. The cross-attention layer is very similar to the self-attention we learned in class before, which uses the following equation with softmax.

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}}) \cdot V$$

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

Where φ(z) is the latent representation. Thus we can see that for cross attention, Q is a projection of the latent representation of the image, and K, V are projections of the input text representation. For the coding questions, we have the students implement a feed-forward layer to prepare the inputs for each attention layer. We then ask students to implement the multi-head cross-attention layer, including the forward pass which is implemented with the same attention that the paper uses. Cross-attention is used in the paper to incorporate the latent representation with the input conditioning.

The feed-forward layer and cross-attention layer are both used in a BasicTransformerBlock which is then used to implement the final Spatial Transformer. For students to come to an understanding of how the Spatial Transformer works, we ask them to implement the layers and then the forward pass.

Position Embeddings

Since the authors use a time-conditional Unet in their diffusion model, we need to add position embeddings for the model to know which of the noise levels we are operating at for each step. We provide the code for Sinusoidal Position Embedding for the students and encourage them to read through it so they can understand that position embeddings are necessary for the Unet.

Denoising Unet

The paper uses an architecture that connects transformers to a denoising Unet (which was introduced in 2015 by a previous paper) as the backbone of the latent diffusion model. The Unet is helpful by downsampling an image into a lower dimensional space and then reconstructing that representation, with the goal of denoising the image. Each time images are passed through the Unet, the Unet removes more noise until our image becomes "clear" again. In the paper, the authors specify that the Unet is primarily built from 2D convolutional layers that perform the downsampling and upsampling.

We provide an image to the students to show how the UNet works and how its architecture gives it the name "UNet." Implementing the Unet includes implementing the time embeddings, the convolution layer, downsampling, and upsampling. To implement the downsampling and upsampling, students make use of the ResNet blocks and transformer blocks they implemented before. We also have the students think about how to include skip connections in the Unet, which like in ResNets, are used to help deal with vanishing gradients. Finally, we include one written question for the students to think more deeply about why a Unet might be used for the diffusion model architecture.

Part 2: Implementing the Forward Diffusion Process On A Single Image
Scheduler

To help students understand how noise relates to diffusion models, we introduce students to the diffusion process itself which includes iteratively adding Gaussian noise until the image input is very noisy. The process of adding noise can be presented as a Markov model where the amount of noise at each time step depends on the previous time step. We present images of the Markov model showing the forward and backward diffusion processes. The diffusion model then reverts the process of diffusion by denoising the image with the use of Unets. The authors use a linear scheduler (i.e. noise added increases at a linear rate at each time step) in their model to specify the variance of the Gaussian noise that should be added at each time step. We have the students implement a linear schedule akin to the one used in the paper after presenting them with four classic schedulers.

To fully cement the students' understanding of the diffusion process, we have them witness the step-by-step diffusion process on an image and see how with the use of the scheduler, the image gradually becomes noisier and noisier at each timestep.

Part 3: Training Diffusion Model

Diffusion Model, Put Together

The main improvement with latent diffusion is the ability to work with text inputs in the latent space instead of having to work on a large image which greatly speeds up the training of the model. This is done with a variational autoencoder with vector quantization (VQ-VAE) to compress the representation of an image into a latent space. However, we do not have the students implement this part of the latent diffusion model, but ensure that they understand this is a key part of the latent diffusion model and the paper's main breakthrough. Because of computation restrictions and the fact that the MNIST dataset on which students train their diffusion models has small-sized images (28 x 28), it is small enough that we do not need to run the images through an autoencoder.

We then put all the components that we built before into one piece, and have the students write parts of the loss function and p-sampling. We also provide a training structure diagram to make it easier for students to understand the training process. Finally, students will be able to train a simple diffusion model themselves and visualize its effects.

1 Paper Introduction

This homework explores Latent Diffusion Models as presented in the paper High Resolution Image Synthesis with Latent Diffusion. Diffusion models learn how to de-noise images that have had noise added to them during the diffusion process, which will be explored in this homework. Stable diffusion is a text conditional latent diffusion model, created by researchers and engineers from CompVis, Stability AI, and LAION. Diffusion models are incredibly expensive to the train so with the use of latent space, we can greatly reduce the computational resource requirements for training, and can also result in very competitive performance on tasks such as inpainting, text-to-image synthesis, super-resolution and more. We encourage you to read over the paper on your own as it is quite fascinating!

1.1 Auto-encoders and Downsampling

Latent diffusion models use pre-trained auto-encoders to produce a lower dimensional space representation of the input. Auto-encoders downsample the input images by a factor of $f = 2^m, m \epsilon \mathbb{R}$. The paper experiments with 6 different sampling factors $f = [1, 2, 4, 8, 16, 32]$ abbrieviated as LDM-f. LDM-1 represents pixel-based diffusion models with no downsampling. The paper tested the performance on these 6 models and measured performance with two different scores: FID (Frechet Inception Distance) which compares the distributions of images generated with the model and real images and IS (Inception Score) which quantifies the quality of images generated by the model.

(a) Given two graphs that show the FID scores and IS scores against training progress, we can see that three of the LDM-f models perform well and three have weaker performance. Identify the top three performing LDM-f models and the bottom three performing LDM-f models. Provide a brief explanation for your answer.

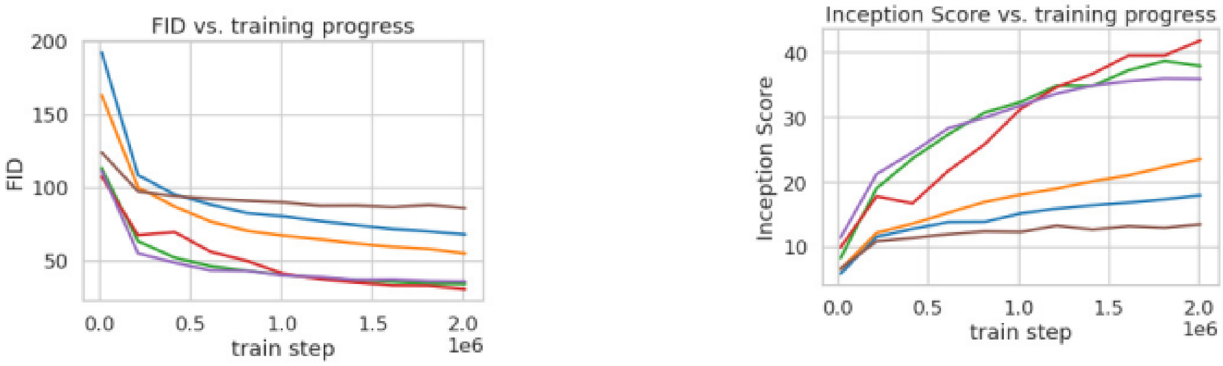


Abbildung 1: IS Scores vs Training Progress

Solution: For FID, the lower the FID score, the greater the similarity between the distributions of the generated images and real images. For the inception score, the higher the score, the better the quality of the model. The top three performing models are LDM-4, LDM-8, LDM-16. The worst of the three models are LDM-1, LDM-2, and LDM-32. With LDM-1 and LDM-2, the performance is too slow and we see the same issues that occur with the large training time and computation requirements of regular diffusion models. LDM-32 results in information loss due to too much compression. With a downsampling factor of $f = [4, 16]$ we can reap the benefits of faster performance without having losing too much information about the image in the process.

1.2 Conditional LDM Loss

For a deeper explanation of the diffusion process, feel free to complete the code in Problem 2 first where an explanation of the diffusion process is provided in the Google Colab. We can think of diffusion models as a sequence of denoising auto-encoders $\epsilon_\theta(x_t, t)$; $t = 1...T$ where x_t is a noisy version of the input and t is sampled uniformly. We get the following loss for diffusion models:

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

For a conditional LDM, we use a conditional denoising auto-encoder which takes in the latent representation, z_t and a domain specific encoder τ_θ that projects a text input y into an intermediate representation.

(b) Write the loss function for a conditional LDM.

Solution: Instead of taking in the noisy input x_t, we take in the latent representation z_t and the representation of a text input y after being passed through the domain specific encoder, τ_θ. The representation of y is then $\tau_\theta(y))$. The output of the conditional denoising auto-encoder can be written as $\varepsilon_\theta(z_t, t, \tau_\theta(y))$. Thus the loss can be written as

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

2 Image Synthesis with Diffusion Model

Please follow the instructions in this notebook. You will implement a text conditioned denoising unet and the sampling algorithm. Then you'll see the diffusion process on a single image and try to train your own diffusion mode. Once you finished with the notebook,

- download submission.zip and submit it to Gradescope.
- Answer the following questions in your submission of the written assignment:

(a) Why might a U-Net by a good choice for the model backbone? List two reasons.

Solution: (This is an open question, if the explanation of the answer is reasonable, then it can be considered correct)

- The model must output the amount of noise in the input, which is of the same shape as the input, meaning both the input and output are of identical shape. This is exactly what a U-Net does.
- UNet has an inductive bias geared towards image-like data. This is due to using convolutions and the downsampling followed by upsampling with skip connections between them which gives it the ability to pick out hierarchical and spatial structures in data.

(b) Screenshot your visualization for get noisy image and include it in your submission of the written assignment. And describe the picture you observed briefly. What kind of process is this?

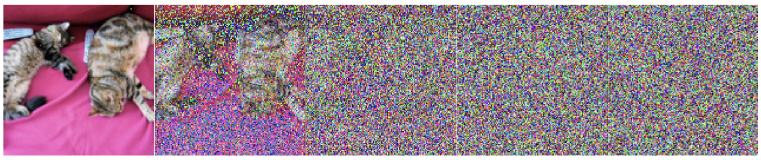


Abbildung 2: Visualization for Sampling

Solution: Shown as Abbildung 2. This should be a process of gradually adding noise to the original picture. Corresponding to the q sample in diffusion model algorithm.

(c) Screenshot one of your visualizations about the sampled images and include it in your submission of the written assignment. Then answer the following question. How does the model perform and does it meet your expectations? If not, what do you think are the directions for improvement?

Solution: The performance of the model was not so great and did not meet expectations. As can be seen from the visualization picture, it only learns the style of the dataset, but does not learn the real semantic information. The main reason is that the datasets and models we use are too small. At the same time, MNIST is not a standard image-text pair dataset, so our text definition is not very accurate. In addition, clip does not perform particularly well with handwritten digits. In order to improve the performance of the model, we also need to consider using a larger and more professional image-text pair dataset while increasing the scale of the model. This further underscores the current trend towards larger models.

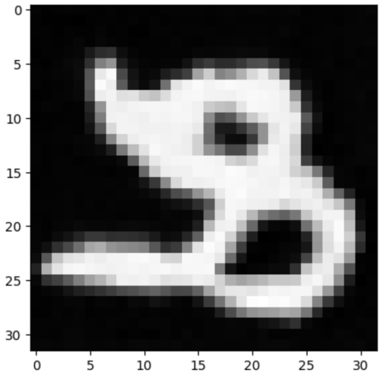


Abbildung 3: Visualization for Model Performance

3 Understanding the VAE Loss

Latent diffusion models use an autoencoder model similar to a VAE to encode images from pixel space to some compressed latent space, perform the diffusion process in this latent space, and finally decode the denoised latent back into pixel space.

To have our model learn the correct latent space, we would like the estimated posterior by the encoder $q_\phi(z|x_i)$ to be very close to the real posterior $p_\theta(z|x_i)$. We can quantify this using the Kullback-Leibler (KL) divergence to get a measure of the difference between the two distributions, defined as

$$D_{KL}(P||Q) = \mathbb{E}_{x\sim P(x)}[log\frac{P(x)}{Q(x)}]$$

However, $p_\theta(z|x_i)$ is intractable in practice. Another goal of VAE-like models is to maximize the probability of generating real samples, or $p_\theta(x)$. During training, rather than directly maximizing this quantity, we maximize a lower-bound called the evidence lower bound (ELBO):

$$log(p_\theta(x_i)) \geq \mathcal{L}(x_i,\theta,\phi) = \mathbb{E}_{z\sim q_\phi(z|x_i)}[log(p_\theta(x_i|z)] - D_{KL}[q_\phi(z|x_i)||p_\theta(z)]$$

At a high-level, this loss can be seen as doing two things:

- **Reconstruction loss**: The first term corresponds to the likelihood of generating images from the true data distribution given a sampled latent
- **Matching Prior**: The second term acts as a regularizer to minimize the difference between the estimated posterior and the prior (generally a standard normal).

Note: In the latent diffusion paper, the autoencoder loss is actually comprised of 3 terms: a patch-based discriminator loss for reconstruction, the standard KL-divergence term between $q_\phi(z|x)$ and a prior $p(z)$, and a regularizing term L_{reg} for the latent z to be zero centered and with small variance. For simplicity, however, we will just be analyzing the traditional ELBO loss here.

(a) **Show** that maximizing the ELBO loss also minimizes $D_{KL}(q_\phi(z|x_i)||p_\theta(z|x_i))$, the KL-divergence of the estimated posterior and the real posterior.
Hint: Start with the KL-divergence formula and manipulate it to get ELBO loss plus some other term.
Solution:

$$\begin{aligned}
D_{KL}(q_\phi(z|x_i)||p_\theta(z|x_i)) &= \mathbb{E}_{z\sim q_\phi(z|x_i)}[log\frac{q_\phi(z|x_i)}{p_\theta(z|x_i)}] &(1)\\
&= \mathbb{E}_{z\sim q_\phi(z|x_i)}[log\frac{q_\phi(z|x_i)p_\theta(x_i)}{p_\theta(z,x_i)}] &(2)\\
&= \mathbb{E}_{z\sim q_\phi(z|x_i)}[log\frac{q_\phi(z|x_i)p_\theta(x_i)}{p_\theta(x_i|z)p_\theta(z)}] &(3)\\
&= -\mathbb{E}_{z\sim q_\phi(z|x_i)}[log\, p_\theta(x_i|z)] + \mathbb{E}_{z\sim q_\phi(z|x_i)}[log\frac{q_\phi(z|x_i)}{p_\theta(z)}] + \mathbb{E}_{z\sim q_\phi(z|x_i)}[log\, p_\theta(x_i)] \\
& &(4)\\
&= -\mathbb{E}_{z\sim q_\phi(z|x_i)}[log\, p_\theta(x_i|z)] + D_{KL}[q_\phi(z|x_i)||p_\theta(z)] + log\, p_\theta(x_i) &(5)\\
&= -\mathcal{L}(x_i,\theta,\phi) + log\, p_\theta(x_i) &(6)\\
& &(7)
\end{aligned}$$

This gives $log\, p_\theta(x_i) = \mathcal{L}(x_i, \theta, \phi) + D_{KL}(q_\phi(z|x_i)||p_\theta(z|x_i))$. Since $log\, p_\theta(x_i)$ is independent of q_ϕ, and thus fixed, maximizing the ELBO loss minimizes the KL term. This tells us that maximizing the ELBO loss does exactly what we want: maximize the likelihood of generating real data and minimize the difference between the estimated and real posterior distributions of the latent space.

5. Team Member Contributions

Jun Tian: Implemented the code of Resnet, Cross Attention and denoising Unet model. Implemented some of the part2 and part3 code. Organized code structure and designed auto grading tests. Draw some schematic diagrams of the network structure. Assisted in designing some of the questions and answers for the writing assignment (coding part). Assist in polishing commentary and final report.

Cathie Lin: Wrote commentary for homework, and the review responses in the final report. Wrote the introduction/questions/solutions for the “Paper Background” section in the written portion of the homework.

Jihoon Yang: Implemented the text embedding from the latent diffusion paper into our codebase. Fixed miscellaneous issues piping embeddings into our attention layers.

Aayush Gupta: In the notebook, wrote some of the explanations (like the diffusion process), helped refactor some code, tested the code, and polished it for readability and clarity. Wrote the VAE Loss question, and helped polish the commentary.

6. Link to Code and Additional Materials

Coding Instruction: Image Synthesis with Diffusion Model(Instruction)
Coding Solution: Image Synthesis with Diffusion Model(Solution)
Project Github repository: https://github.com/jun-tian/CS182_Project_diffusion