CSE 291B Project Final Report: Hierarchical U-Net Vision Transformers with Residual Cross Attention for Latent Diffusion

Arman Ommid
Department of Computer Science
UC San Diego
aommid@ucsd.edu

Mayank Jain
Department of Computer Science
UC San Diego
maj014@ucsd.edu

The development of vision transformer (ViT) backbone architectures continue to advance in light of their proficiency in traditional vision tasks while progress with diffusion models have only recently turned to attension based denoising from popular U-Net based convolutional neural networks (CNN). We parallel progress in architectural research and continue recent trajectories in latent diffusion by designing a U-Net based hierarchical vision transformer for latent diffusion image generation with attention based residual connectivity (X-Swin). Current state-of-the-art attention based methods in latent diffusion suffer computationally inefficient full context attention that encourages coarse representation learning, lack local inductive bias and therefore lack of hierarchical processing, and lack UNet multi-scale feature learning that combines localized, shallow features with global, deep features. To address this, we design X-Swin as our denoising network in the latent diffusion pipeline and is characterized as an augmented Swin Transformer with versatile hierarchical layering and U-Net residual connections implemented using cross attention for guided and refined localized feature aggregation. We show our architecture design improves on existing state-of-the-art architectures on image generation for 64x64 images on Caltech101 with limited computation resources through quantitative and qualitative comparisons. Our proposed architecture improves on state-of-the-art transformer based denoising latent diffusion by $\sim 36.46\%$ for FID, $\sim 11.54\%$ for IS, $\sim 12.62\%$ for Precision, and $\sim 3.69\%$ for Recall on these conditions and further motivate future works to adopt our architecture at scale.
https://github.com/ArmanOmmid/XSwinDiffusion.git

1 Introduction

We wish to improve on the problem of image generation; that is to learn the distribution over images. More specifically, we wish to improve on image generation by designing a more advanced, attention based denoising backbones for latent diffusion. This is significant as latent diffusion has become establish as a state-of-the-art image generation technique while being efficient; but recent advanced backbones that use attention based vision architectures have inefficient computational scaling and lose on potentially powerful inductive biases for effective learning. To improve on these backbones would allow for more efficient and effective latent diffusion and therefore improvements in image generation. Towards this goal, we propose a hierarchical Swin based transformer with residual cross attention for more efficient attention computations and multi-scale hierarchical learning.

Problem Definition. Most generally, we want to solve the problem of image generation. This problem is about learning how to model a distribution over images and to do so with some sufficient degree of image diversity and image quality. Learning these complex distributions is difficult by the complexity of the data domain. Popular generative techniques like variational autoencoders (VAE) and normalizing flows (NM) are known to produce low fidelity data while generative adversarial networks (GAN) suffer from unstable training and diversity challenges like mode collapse or data copying. We approach the problem with the techniques based on denoising diffusion graphical models [16] as these approaches have been shown to be effective in image synthesis [9]. However,

Preprint. Under review.

the iterative nature of diffusion makes for expensive computation, which is why we have elected the latent diffusion approach for this problem.

Latent diffusion [35] encodes images from pixel space to a latent space where diffusion is performed in the latent space and then decoded back into the pixel space; this lowers the computational costs of performing denoising. It does so using a CNN based autoencoder that combines objectives from VAE [22], PatchGAN [19], and a perceptual loss [21]. In the latent space, the denoising steps are performed by denoising networks traditionally designed as CNN based U-Nets with concatenation or cross attention mechanism for conditioning using embeddings from an arbitrary choice of pretrained conditioning networks.

For our formulation of the problem, we implement a more advanced vision architectures for the denoising network to improve the semantic generative process of the latent diffusion process. Namely, we replace the CNN based diffusion U-Net for a Swin based transformer, include strategic placements of convolutional and global attention blocks, and then leverage a localized cross attention for more guided residual connections that preserve locality biases and remain computationally efficient. In this way, we preserving the U-Net's multi-scale architectural design. For conditioning, we use adaptive layer normalization modulators that condition on both time step and class embeddings.

Problem Significance. This problem is relevant to generative AI since state-of-the-art approaches for latent diffusion based image synthesis have generally used U-Net CNNs for their backbones; meanwhile there has since been much progress on improved state-of-the-art architectures for vision related tasks that extend beyond traditional U-Nets into the paradigm of attention based vision transformers. This leaves unrealized potential for improving on the semantic processing of latent diffusion by using state-of-the-art vision transformer architectures.

Only recently has latent diffusion begun to expand onto vision transformers [6] [12] [17] [31] [32]; however, these approaches don't preserve the multi-scale feature learning that motivate the U-Net architecture nor are they computationally efficient as they perform quadratic scaling attention on the full image context and as a results demands coarse patching. Meanwhile, there exist variations on vision transformers that address these concerns towards a more faithful attention based U-Net implementation [7] [11] [23].

Notably, SwinV2 [25] has been implemented for Imagen, but it does not take advantage of the computational efficiency of latent diffusion as it operates in the pixel space. Extending state-of-the-art architectures in vision to latent diffusion with UNet Swin based architectures has the potential to improve both the quality and computational efficiency of latent diffusion which contributes to the accessibility and performance of the image generation problem.

Technical Challenge. The challenges we expect for this project is mainly on the programming side. We need two major components for our codebase. Firstly, we need a latent diffusion training pipeline to train our model. Secondly, we need an implementation of our architecture. From our literature review, we found a minimal latent diffusion pipeline to start off with from the Diffusion Transformers paper [31]; this can also naturally serve as our baseline for comparison. For our model, we intended to use the SUNet [11] implementation to augment with cross attention skip connections and integrate into the diffusion pipeline. However, we instead started with SwinV2 Blocks [26] and built our architecture largely from scratch to ensure we could implement it with high technical precision. The challenge for these steps is mainly in understanding these codebases, learning how to modify them, and ensuring we can have a model agnostic pipeline to be able to train different models through our pipeline. Data wrangling also serves as an issue to deal with and our dataset of choice for benchmarking depends on the practicality of our compute resources which may not become apparent until we have a working pipeline and have estimates for how long experimentation might take. Lastly, there are challenges in terms of implementing an evaluation pipeline so that we can measure the FID, IS, and Recall/Precision scores of our models.

State-of-the-Art. State-of-the-art in transformer based latent diffusion extends from the Diffusion Transformer (DiT) [31] architecture that first introduced vision transformers to replace the traditional U-Net backbones for latent diffusion. Architectural choices in transformer based latent diffusion has since evolved to introduce long range residual connections to parallel those of U-Nets with U-ViT [6], performing latent masking for stronger contextual learning with Masked Diffusion

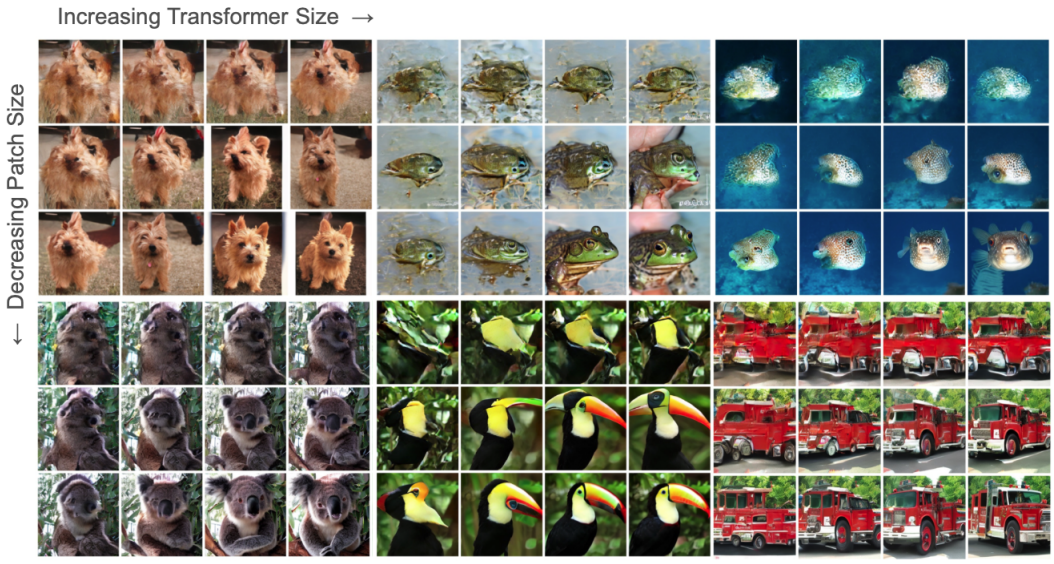

Figure 1: DiT Demonstration of effects of decreasing patch size at the cost of computational complexity due to the quadratic scaling of full context attention [31].

Transformers (MDT) [12], and use other researched techniques to improved performance like mixed U-ViT architectures that make selective layer choices as well as non-architectural choices [17] [32].

Focusing on the backbone architectures for latent diffusion, these state-of-the-art techniques forfeit the explicit multi-scale, local-global learning inductive bias induced by the original U-Net architecture. While convolution is naturally effective at low-level and local learning [18], attention provides a coarser, high level and global learning mechanism [14]. Since locally biased learning is not performed with these state-of-the-art U-ViT architectures in latent diffusion, these denoising networks do not perform analogous multiscale learning as with CNN U-Nets. Additionally, full context attention is computationally expensive and is responsible for encouraging coarse patching for the sake of efficiency. [31].

Meanwhile, state-of-the-art Swin architectures provide this stronger low level and local learning within the attention based paradigm in a hierarchical fashion towards flexible multi-scale learning while having localized context windows for efficient attention computations [26] [27]. Swin based U-Net architectures, meanwhile, can then effectively perform the analogous multi-scale local-global feature learning as in UNets [7]. These have demonstrated to be successful in denoising related tasks [11]. State-of-the-art techniques in traditional diffusion take advantage of these properties [25] while state-of-the-art denoising further improve on this architecture with selective, cross-attention based skip connections [23].

This motivates a latent diffusion implementation based on U-Net Swin backbones with cross attention skip connections for denoising (X-Swin).

Contributions

1. Improve on attention based latent diffusion image generation
2. Design and implement a hierarchical UNet Swin denoising backbone for latent diffusion
3. Augment the architecture with:
   (a) Localized residual cross attention
   (b) Outer convolutions
   (c) Global attention bottleneck
   (d) Conditioning modulation
4. Compare our performance with state-of-the-art attention based latent diffusion models
5. Perform ablation studies with localized residual cross attention

2 Related Work

2.1 Vision Transformers

The transformer architecture [39] has become increasingly dominant in state-of-the-art approaches in many major domains of deep learning and vision related tasks have been no exception since the advent of vision transformers (ViT) [10]. ViTs have since continued to demonstrate impressive scaling properties [41] and have maintained their presence in state-of-the-art computer vision.

However, the formulation of attention for vision has some notable distinctions in its inductive biases with respect to convolution. The strong local spatial inductive bias of convolution characterizes a high pass filter that learns refined, high frequency, lower level features (for example, textures) [18] [20] and naturally struggles with non local information interactions [20]. Meanwhile, attention characterizes stronger expressively with longer distance context interactions and dynamic feature aggregation with a weaker inductive bias [14] [20]; however, they produce single scale and low resolution representations which are not as effective at capturing fine-grained details [14]. This makes for a coarse, low frequency, higher level feature extractor and this is compounded by the necessity of patching due to the prohibitive nature of dense level attention.

Swin Transformers [26] [27] were designed to address these concerns by offering multi-scale feature extraction by limiting the attention window. This not only permitted for localized attention, but also permitted denser localized attention as limiting the attention scope significantly regularizes and reduces the quadratic scaling attention computations. This improves the fine grained representation learning abilities of ViT as it balances expressive context processing with sharper local processing. This design is showcased by Figure 2

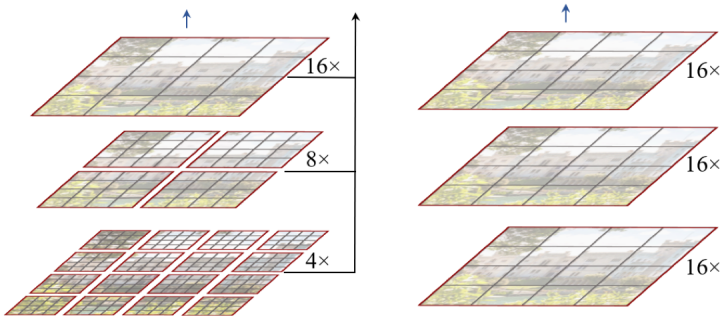

Figure 2: Swin Attention vs ViT Attention [27]

With multi-scale and localized feature learning, Swin can appropriately extended ViT to a meaningful U-Net design where skip connections represent combined local-global reasoning rather than just being long range residual connections with coarser local information. These U-Swin architectures have seen successful in segmentation tasks which require finer grained learning [7] and have demonstrated effectiveness in denoising vision tasks where these same properties are similarly very important [11] [23]. The design is showcased by Figure 3.

Lastly, very recent work explores more expressive connectivity within a U-Net Swin architecture by designing skip connections with cross attention over simple feature concatenation; allowing selectivity of global features towards local features [23]. The design is showcased by Figure 4.

As an isolated backbone network, our proposed architecture relates to these previous works by taking the U-Swin architecture and integrating these cross attention modules in an efficient, localized fashion faithful to the localize self-attention design of Swin and concatenating the outputs to the decoded features. Additionally, we also use outer fully convolutional blocks at the input and output layers as well as inner ViT blocks with global self-attention at the bottle neck to afford stronger hierarchical reasoning. The motivation and implementation of these modifications are further discussed in our methods section.

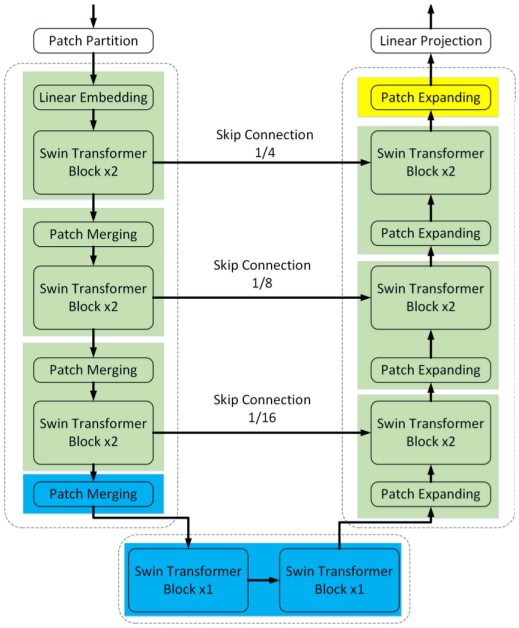

Figure 3: UNet Swin [7]

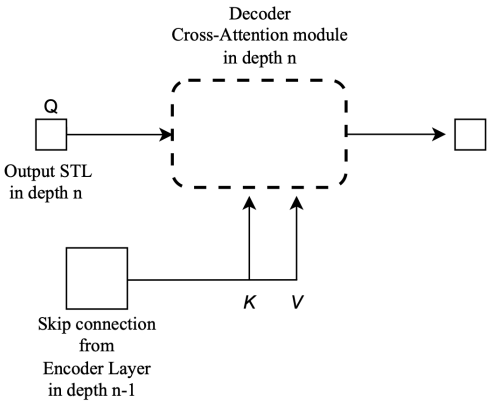

Figure 4: UNet Decoder Cross Attention [23]

2.2 Diffusion Models

Denoising diffusion probabilistic models (DDPM) [16], or Diffusion models have been particularly successful for image generation [28] [34] [37] succeeding generative adversarial networks [13] which were previously state-of-the-art [9]. These formulate generative modeling as a Markov chain of noisy latent variables towards pure noise and then learn the reverse process to recover the target data distribution from noise. A common technique is to provide additional conditioning using embeddings of text using models like CLIP [33] .

However, diffusion models are computationally expensive to run for high resolution generation since diffusion denoising is performed in the pixel space. This concern was addressed by Latent Diffusion Models (LDM) [35] which instead leverages a pretrained autoencoder to downsample high resolution images into a latent space and then perform the diffusion process in this lower dimensional space, thus saving computation. To ensure high fidelity compression and reconstruction, the authors take advantage of a variational autoencoder (VAE) [22] augmented with Perceptual Loss [21] and PatchGAN Discriminator Loss [19] to ensure both semantic and high resolution reconstruction. This is showcased by Figure 5.

Meanwhile, these major contributions and advancements in diffusion modeling still utilize convolutional U-Nets [36] as their backbone denoising network [36] despite aforementioned advancements in vision architecture namely following ViT; with the exception of sometimes using few cross attention layers for conditioning. With regards to latent diffusion, only very recently has transformer based backbones been incorporated into the latent diffusion process with contributions like DiT [31], which applies ViT for latent diffusion; U-ViT [6] which applies a U-Net like ViT for latent diffusion; and masked diffusion transformer (MDT) [12] which takes advantage of masked transformers in latent diffusion for more robust denoising. Other state-of-the-art approaches in latent diffusion use partial transformer architectures as well [17] [32].

Despite this, to our knowledge, Swin architectures, let alone Swin U-Net architectures, have not made an appearance for latent diffusion. For diffusion image synthesis, the closest we find is SwinV2-Imagen [25] and MT-DDPM [29] that do this for pixel level diffusion instead of latent diffusion. We also see a Swin based backbone for diffusion based weather forecasting [5]. Additionally, none of these leverage selective skip connections using cross attention.

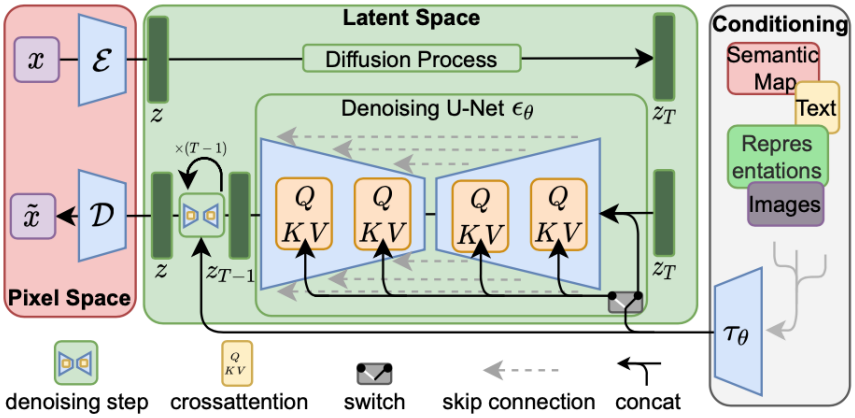

Figure 5: Latent Diffusion Model [35]

We hypothesize that our proposed architecture, a strongly hierarchical UNet transformer with refined skip connections, may prove valuable in latent diffusion where fine grained details in the latent space are especially semantically important and cross attention based local-global connectivity provides more guided multi-scale information processing. Furthermore, fine grain patching is afforded by the computationally more efficient localization of Swin context windows.

3 Methodology

Problem Setting. We want to learn and model the distribution over images $x \sim I$ with a function approximator such that we can sample our parameterized distribution and generate high quality images. We approach this problem with the latent formulation of diffusion where we instead perform diffusion on latent variables that control the semantic information of data in the pixel space. Latent diffusion [35] separates the problem of image generation into two stages: semantic compression and diffusion.

The semantic compression defines an encoder-decoder scheme for learning latent representations of images that enable reconstruction. It defines an encoder network $\mathcal{E}_{\phi_1}$ and a decoder network $\mathcal{D}_{\phi_2}$ such that $z_0 = \mathcal{E}_{\phi_1}(x)$ and $\tilde{x} = \mathcal{D}_{\phi_2}(z_0)$ where the semantic distance between x and $\tilde{x}$ is low (and to be defined later with the composite autoencoder loss [35]).

The diffusion stage defines the diffusion process over latent representations of various degrees of iterative corruptoin z_t and $t \in [0, T]$ with a fixed, forward, noising Markov chain q and a parameterized, reverse, denoising Markov chain p_θ. We want our reverse Markov chain to learn to predict $z_{t-1} = p_\theta(z_t, t)$ where $z_t = q(z_{t-1}, t)$.

Sampling from the model would look like $\mathcal{D}_{\phi_2}(p_\theta^{(T)}(z_T, t))$ where we apply the denoising step T times starting from a pure noise latent z_T and then decode back into the pixel space.

Idea Summary. Our idea can be summarized as performing latent diffusion with a U-Swin transformer backbone that uses cross attention skip connections. The non-backbone related architecture reflects that of traditional latent diffusion as originally formulated [35]. The novel contributions are reflected in the backbone changes for latent diffusion. Namely, this is using the cross attention based U-Swin transformer. This idea makes progress on the problem of image generation by applying more advanced, state-of-the-art vision backbones to a leading image generation design (latent diffusion models).

We use known strengths and weakness between the inductive biases of CNNs and ViTs [14] [18] [20] as well as known strengths of UNet architectures [36] to inform and motivate our idea of using a U-Swin architecture for latent diffusion. This is further motivated by contributions that use these kinds of multi-scale transformer architectures for improving on vision tasks where high frequency information has a stronger role in semantic variance [7] [26] [27] in addition to works that improve on

denoising related tasks [11] [25] [29]. Giving more selective multi-scale learning with cross attention in the UNet design is similarly motivated from a recent work for denoising with U-Swin transformers [23]. We hypothesize that this design will improve on current latent diffusion based approaches by providing true and selective multi-scale latent reasoning in a transformer based UNet design.

If our idea does not work out, we may try experimenting with a more simplified version of our architecture such as using traditional UNet concatonation or trying cross attention skips only; or perhaps attempt to do transformer based latent diffusion on a novel dataset and analyze its performance.

Description Our method can be decomposed into 3 main subjects of interest: The autoencoder, the diffusion framework, and the backbone architecture.

3.1 Autoencoder

First we define an encoding decoding scheme to reduce diffusion based generation into a learned latent space of images. Given an image $x \in \mathcal{R}^{H \times W \times 3}$ in an RBG image space, we have the following:

$$z_0 = \mathcal{E}_{\phi_1}(x)$$
$$\tilde{x} = \mathcal{D}_{\phi_2}(z_0) = \mathcal{D}_{\phi_2}(\mathcal{E}_{\phi_1}(x))$$

Where the latent representation z_0 is in some lower dimensional space $z_0 \in \mathcal{R}^{h \times w \times c}$ and $\tilde{x}$ is a reconstruction of x given the latent representation. The encoder $\mathcal{E}_{\phi_1}$ and decoder $\mathcal{D}_{\phi_2}$ are learned using the traditional variational evidence lower bound objective [22] in conjunction with a perceptual objective [21] to promote semantic reconstruction and an adversarial PatchGAN discriminator objective [19] to promote local realism in the output. The variational ELBO objective can be separated into a MSE reconstruction term and a KL-Divergence latent regularization term. Perceptual loss essentially performs an MSE loss in the latent space of a frozen, pretrained vision model (e.g. VGG16) denoted as $\mathcal{P}$. The PatchGAN discriminator objective performs uses binary cross-entropy to discriminate image patches as real or fake; we denote the discriminator as D_ψ.

For pretraining the autoencoder, we have the combined objective:

$$\begin{aligned}\mathcal{L}_{autoencoder} = \min_{\phi_1,\phi_2} \max_{\psi} [& \\ & \mathcal{L}_{elbo}(x, \mathcal{D}_{\phi_2}(\mathcal{E}_{\phi_1}(x)) & + \\ & \mathcal{L}_{perceptual}(\mathcal{P}(x), \mathcal{P}(\mathcal{D}_{\phi_2}(\mathcal{E}_{\phi_1}(x))) & + \\ & \mathcal{L}_{discriminator}(D_\psi(\mathcal{D}_{\phi_2}(\mathcal{E}_{\phi_1}(x)), y_{fake}) & + \\ & \mathcal{L}_{discriminator}(D_\psi(x), y_{real}) \\ ]\end{aligned}$$

For our implementation of the encoder and decoder, we use the autoencoder implementation `AutoencoderKL` from `diffusers` [40] which was pretrained on ImageNet [8] with the above technique.

3.2 Diffusion

The autoencoder training is separated from training the denoising network for diffusion. We use the latent diffusion design [35] for performing diffusion on the learned latent space. This defines a fixed forward Markov chain q that destroys the information of a signal towards noise and a parameterized reverse Markov chain p_θ that performs the inverse process to recover the original signal from noise. Because this is done on latent variables, we use z instead of x.

$$q(z_{1:T}|z_0) = \prod_{t=1}^{T} q(z_t|z_{t-1}) \qquad p_\theta(z_0|z_{1:T}) = \prod_{t=1}^{T} p_\theta(z_{t-1}|z_t)$$

The forward Markov chain applies noise according to a noise schedule with $\beta_t \in [0, 1]$. We can simplify this using the analytic closed form for integrating over Gaussians where $\alpha_t = \prod_\tau^t (1 - \beta_\tau)$.

$$q(z_t|z_{t-1}) = z_t \sim \mathcal{N}(\sqrt{1-\beta_t} z_{t-1}, \beta_t I)$$
$$q(z_t|z_{t-1}) = z_t \sim \mathcal{N}(\sqrt{\alpha_t} z_0, (1-\alpha_t) I)$$

The reverse Markov chain learns the denoising process to recover the latent before applying the noise.

$$p_\theta(z_t|z_{t-1}) = z_{t-1} \sim \mathcal{N}(\mu_\theta(z_t, t), \Sigma_\theta(z_t, t))$$

Diffusion can be done using the variational evidence lower bound (ELBO) objective. However, a simplified objective can be used that instead predicts the noise at each time step and does not involve a parameterization to predict the variance of the noise. [16]

$$\mathcal{L}_{simple} = \min_\theta \|\epsilon - \epsilon_\theta(\sqrt{\alpha_t} z_0 + \sqrt{1-\alpha_t}\epsilon, t)\|^2$$

Where ϵ_t is the total noise at t and ϵ_θ is our function approximator that now regresses the noise itself rather than regressing the reconstruction. ϵ_θ receives the noised latent z_t (expanded into its cumulative noise form on z_0) at t as conditioning (with the appropriate timestep embedding for t).

With the encoding scheme and diffusion scheme defined, the latent diffusion framework can be described with with Figure 5. Any conditioning information can be incorporated with either concatenation or cross-attention using a pretrained encoder τ_θ.

3.3 Backbone

The implementation of our architecture follows from UNet Swin designs [7] [11] with cross attention skip connections [23]. These are showcased in Figures 3 4.

The original Swin Transformer architecture [27] uses paritioned, smaller context windows with higher patching resolutions alongside merging layers to incorporate a stronger locality bias and improved learning resolution. Because of this, Swin provides an architectural prior for multi-scale feature learning which can be leveraged for a faithful re-design of UNet with Swin layers. Towards this end, we implement Swin patch-expanding layers that perform patch expanding to reconstruct higher resolutions and effectively reverse the effects of patch merging. To note, Swin also uses relative positional embeddings that trivialize their input parsing.

Regarding our implementation of our residual cross attention, we deviate from previous implementations by localizing the cross attention window and having our cross attention modules output refined local information to be combined with global information.

By augmenting our decoder residual cross attention to be localized within the respective Swin context windows of the current stage, we not only afford the performance gains of the Swin design, but this ensures the transformed embeddings only aggregate localized information. We then concatenate these with the original decoder embeddings to combine both local and global information for downstream Swin decoder blocks. This means our cross attention residual modules play a supportive, refinement role in processing local information but still allows the decoder blocks to fully consider both refined local and global features. To note, the first Swin block in each decoder stage is followed by a pointwise convolution to downsample the hidden representations to the appropriate dimensionality. This is showcased by Figure 6.

Next, we further augment our architecture to include outer convolution blocks before patching and after unpatching as well as inner global attention ViT blocks in the bottleneck. These choices further promote hierarchical learning and were originally inspired by early ablation tests on our architecture that observed patching artifacts on few shot segmentation tasks. Early and late convolutions provide even stronger localized learning prior to the resolution loss of patching and perform further pixelwise local refinement at the output. The ViT bottleneck provides globalize attention to ensure that global information is still received by all feature embeddings and remains efficient as it is performed in the lowest resolution stage. Positional encodings are manually reintroduced here since global attention requires absolute positional embeddings.

In isolation, our backbone is showcased by Figure 7.

3.4 Latent Denoising

To augment our backbone for latent diffusion, we need to incorporate conditioning information. We augment the architecture for latent diffusion by introducing modulation layers at the beginning of every layer for conditioning on both time step and class embeddings and thus enabling classifier free guidance. We implement our modulation layers as per DiT [31] using adaptive layer normalization

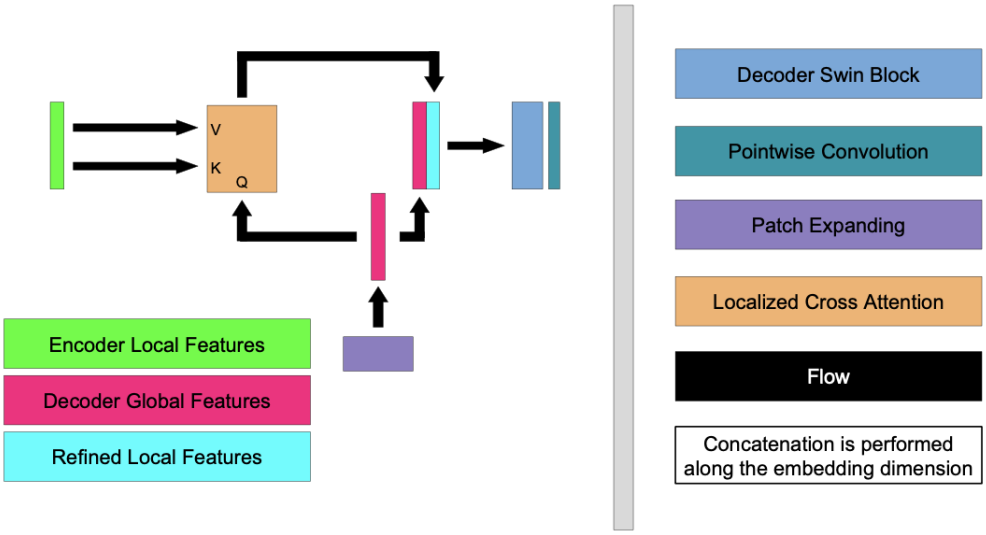

Figure 6: Our XSwin Cross Attnention. Following patch expanding, we use decoder (global) features to query encoder (local) features within the localized Swin context windows. This ensures the cross attention refinement still only aggregates local features so that multiscale reasoning can instead be fully done in the subsequent Swin decoder block. We follow this with one pointwise convolution layer to control dimensionality.

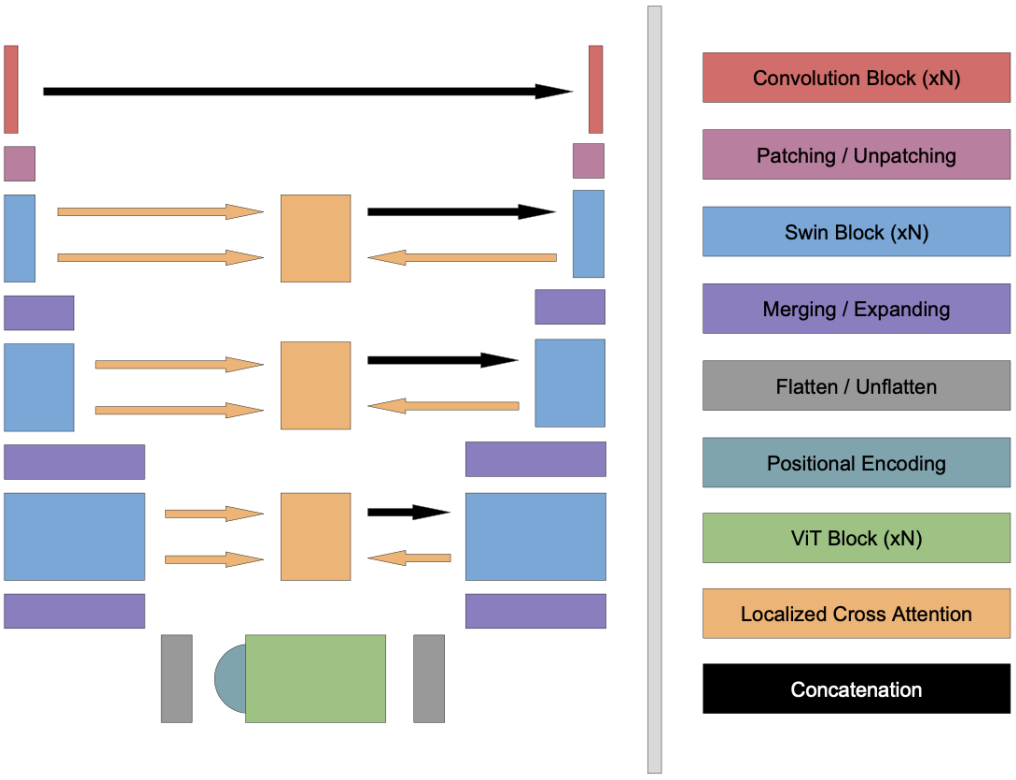

Figure 7: Our Isolated XSwin Backbone. We implement a UNet Swin architecture modified with localized cross attention that uses encoder features as keys/values and decoder features as queries to output refined encoder features and concatenate them with the original decoder features. Not shown is that we use pointwise convolution after the first decoder block in each stage to control dimensionality.

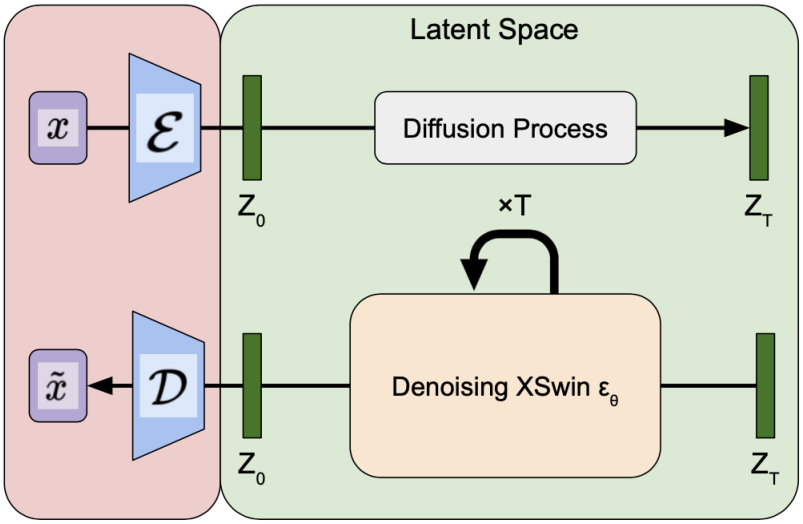

Figure 8: Our XSwin Latent Diffusion Pipeline. Not explicitly shown is is how we embed conditioning information, namely time step and class label conditioning information. We do this with time step and class embeddings and by augmenting *every* parameterized layer in our model from Figure 7 with an input modulation layer that modulates our features with the conditioning information. We implement this using an adaptive layer normalization module as per DiT [31].

with a modulator as it has been demonstrated perform better than cross attention or concatenation. To this end, we densely apply this conditioning mechanism as every parameterized layer of our architecture is internally preceded by a modulation layer to condition on time step and class information. In our codebase, one can observe both our normal XSwin architecture designed (and validated) on segmentation and our XSwinDiffusion architecture where every layer is re-implemented with conditioning modulation.

This is showcased by Figure 8.

Implementation With regards to our backbone architecture, we used PyTorch [30] and implemented it from scratch with few exceptions. We used the `torchvision` [30] implementation of the SwinV2 and PatchMergingV2 Blocks [26] and ViT Blocks [10]. Besides this, we implemented our backbone architecture from scratch without reference to any codebase. This includes model class implementation as well as modules like Patching, Unpatching, PatchExpandingV2, Localized Residual Cross Attention, and formal modules for the convolution blocks and the pointwise convolution layers.

With regards to latent diffusion, we implemented our pipeline based on their DiT latent diffusion pipeline [31]. From their code, we use their main diffusion training runner and sampler as well as their implementations for time step, class, and positional encodings, the latter taken from FAIR [2]. They, and by extension we, used `diffusers` [35] for the latent encoder and decoder and various components from OpenAI's diffusion codebases [1] [3] [4] for the gaussian diffusion. For integrating modulation, we formalized their modulation functions into generalizable layers and re-designed all our existing modules (including the ones imported from `torchvision`) with our custom Modulator layer. We also implemented a simple re-design of Sequential for use with conditioning information.

For the most part, we identified that the DiT [31] codebase was relatively minimal and modular and designed our architecture to be able to simply plug and play into their pipeline with regards to the inputs and outputs.

With regards to architecture, we experimented with using architectural blocks beyond Swin which led to the decision to include convolution blocks at the outer layers as a response to patching artifacts discovered in early segmentation testing as well as a global attention ViT layer at the bottleneck where global attention would be efficient and serve as the top feature hierarchy stage. We used batch normalization for all convolutional layers and layer normalization for all attention based

layers. We experimented with both dropout and attention specific dropout, but settled instead for using stochastic depth to regularize the model complexity. Some other hyperparameters include the patch size, embedding dimension, Swin stage depths, Swin window size, number of attention heads, attention MLP expansion multipliers, number of global ViT blocks at the bottleneck, whether or not to downsample before the bottleneck, class dropout percentage, and whether or not to use the localized residual cross attention as opposed to just concatenation for skip connections.

To ensure fair comparison with our baseline models, we tried to make sure hyperparameters like patch size, embedding dimension, stage depths, and number of attention heads were comparable to that of our baseline architectures; as well as making necessary adjustments to ensure similar parameter sizes. With regards to training hyperparameters, we chose those similar to the DiT experiments [31] and did not do much experimentation on this front as training diffusion was very expensive computationally and time wise, even with reduced resolution. Our main goal was to compare with existing state-of-the-art on their preferred hyperparameter configurations.

For our experiments, we downsampled our image sizes to 64x64, used batch sizes of 128, and ran our experiments for 1500 epochs with 150 diffusion steps. We found this was the most feasible combination to ensure we could do the amount of experiments we wanted for the time and compute resources we had available for us. All experiments used the same global seed (0).

4 Experiments

Datasets and Tools For our work, the choice of dataset was primarily influenced by the time requirement for training. Specifically, our dataset needed to be large enough so that the model could learn meaningful information and generate high-quality images. However, considering that state-of-the-art models typically require days of training on multi-GPU setups, we also had to be mindful of our limited compute resources. With these factors in mind, we chose the Caltech101 dataset [24] for our experiments. This dataset comprises 9,144 images across 101 object categories, with each category containing between 40 to 800 images. The sizes of these images vary but typically range from 200 x 300 pixels. In our experiments, we resized all images to 64 x 64 pixels to enhance computational speed. We use 80% of the data to train with and reserve 20% to to evaluate on metrics on; we do this split while preserving the original class proportionality in both splits so that the evaluation distributions cover all possible modes fairly.

In terms of tools, we have used GPU Jupyter Notebooks on the DSMLP platform as well as on Google Colab Pro. All code was done in Python with PyTorch as the deep learning framework.

Baselines We provide the following baselines to compare our proposed architecture with:

1. Scalable Diffusion Models with Transformers (DiT) [31]: We chose this method due to its transformer-based architecture, which aligns closely with our approach. As a state-of-the-art diffusion model, DiT represents a significant advancement in the field. Its accessible GitHub repository also aids in reproducibility and facilitates direct comparison.
2. All are Worth Words: A ViT Backbone for Diffusion Models (U-ViT) [6]: U-ViT extends DiT by simply introducing UNet like skip connections in spirit of the original UNet designs. This serves as the most similar variation of attention based latent diffusion with respect to our architecture in that in incorporates skip connections in a UNet style. However, UViT incorporates conditioning information differently that DiT through additional sequence tokens rather than modulation. We instead augment the original DiT architecture to have these skip connections (with embedded dimensional downsampling as per the UViT design) and compare the results as the main difference of interest is the UNet like skip connections.

These baselines have been chosen based on their architectural relevance and exceptional performance in image generation tasks. DiT's focus on introducing transformers into latent diffusion models and U-ViT's use of skip connections combining shallow and deep features provide comprehensive benchmarks for evaluating our proposed methods given the extent at which state-of-the-art attention based latent diffusion has developed. Both models have achieved remarkable results in class-conditional image generation and offer insights into the scalability and effectiveness of transformer-based architectures in diffusion models.

We also wish to perform an ablation study with our proposed architecture by observing the performance of our hierarchical model without the localized residual cross attention modules. To this end, we also run experiments with a USwin architecture, which is an exact replica of our XSwin architecture without the localized residual cross attention and instead using direct concatenation for skip connections like UViT. To note, while we will call this architecture USwin, it is not a purely USwin architecture since, as discussed before, we improve the feature hierarchy with convolutional and ViT layers in appropriate locations.

Overall, we compare 4 architectures:

1. **DiT** - Transformer Based Denoising Network
2. **UViT** - Transformer Based Denoising Network with UNet residual connectivity
3. **USwin** - Swin Transformer Based Denoising Network with Convolutional and ViT Blocks supporting the feature hierarchy and UNet residual connectivity.
4. **XSwin** - Swin Transformer Based Denoising Network with Convolutional and ViT Blocks supporting the feature hierarchy and UNet localized cross attention residual connectivity.

We run our experiments with the same global seed (0) with 64x64 images of our training split of the Caltech101 dataset over 1500 epochs with 150 diffusion steps with a batch size of 128. We ran this on Google Colab with V100 GPUs.

Evaluation Metrics To evaluate the performance of our generative models, we will employ both qualitative and quantitative metrics. Qualitatively, we will present samples of generated images. This approach allows for a visual assessment of the model's output, giving insights into the aesthetic and realistic aspects of the images.

Quantitatively, we will primarily use the Fréchet Inception Distance (FID) [15], which is a widely recognized standard in generative modeling. FID measures the distance between the feature vectors of real and generated images, as computed by an Inception-V3 model. It works by comparing the distribution of generated images to the distribution of real images in the feature space of an Inception-V3 model and attempts to capture both the fidelity and diversity of a generative model in its samples

We also report the Inception Score (IS) [38], which takes into account the following two factors: the clarity of the image (how easily it can be classified) and the diversity of the images (how varied the generated images are.) The Inception Score is calculated by exponentiating the KL divergence between these two distributions (the conditional label distribution for each image and the marginal distribution over all the images). A higher score indicates that the GAN produces high-quality, diverse images.

Lastly, we also include Precision and Recall metrics to evaluate how well the generated distributions match the real dataset. We evaluate Precision as the probability that a random image from the generated distribution falls within the support of the real distribution and Recall as the probability that a random image from real distribution falls within the support of generated distribution. These correspond to the average sample quality and the coverage of the sample distribution, respectively.

Quantitative Results In this section, we present the quantitative results obtained by comparing our proposed model, XSwin, our ablation of our proposed model, USwin, and the two baseline models DiT and UViT. The performance is evaluated using four metrics: the Fréchet Inception Distance (FID), the Inception Score, Precision, and Recall. Table 1 provides detailed overviews of these evaluations, respectively.

The FID scores, which measure the distance between the feature vectors of real and generated images, are reported. A lower FID score indicates better model performance, reflecting higher quality and diversity of the generated images. We also show the Inception Scores, which are a measure of both the clarity and diversity of the generated images. Higher scores indicate that the generated images are both highly discernible and diverse. Furthermore, we measure the FID and IS scores by taking the mean and standard deviations of these scores over 10 separate generative trials. Finaly, we include Precision and Recall metrics to further describe the average sample quality and the coverage of the sample distributions.

Model	FID Score	Inception Score	Precision	Recall	Parameters
DiT	158.30 ± 0.24	1.05 ± 0.002	0.0186	0.2049	**33M**
UViT	23.25 ± 0.098	1.30 ± 0.003	0.1013	0.6228	34M
USwin	17.92 ± 0.14	1.40 ± 0.003	0.1099	0.6282	34M
XSwin	**14.78 ± 0.11**	**1.46 ± 0.004**	**0.1141**	**0.6458**	35M

Table 1: Combined evaluation of the proposed model and baselines with FID Scores, Inception Scores, Precision, and Recall

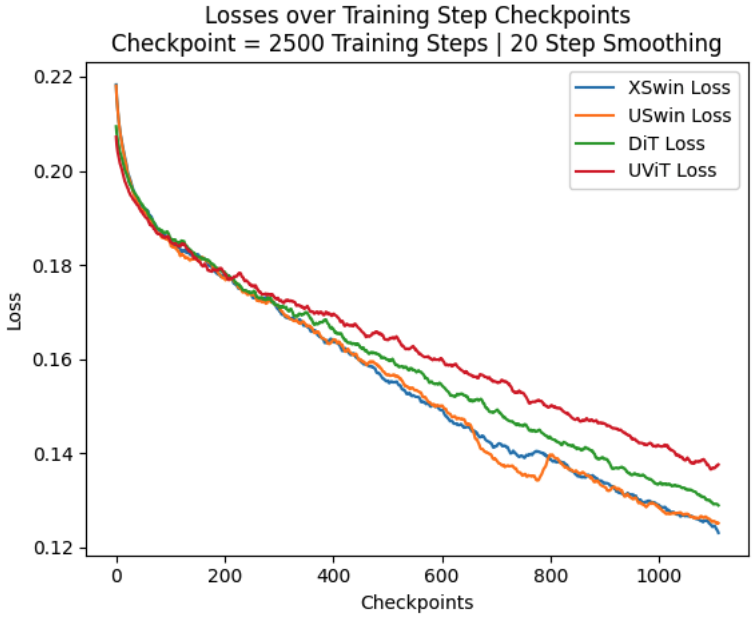

Figure 9: Training Losses over Training Checkpoints. Each checkpoint represents 2500 training steps. The original curves were highly noisy so we smoothed these results with a 20 step window so that they are more interpret-able. These curves represent training for exactly 1517 training epochs which results in 1100 checkpoints.

The results indicate that our XSwin model outperforms the baseline models in FID, Inception Score, Precision, and Recall, suggesting that it is capable of generating images with higher fidelity and greater diversity. The XSwin model achieves the lowest FID score, which is an indication of its superior performance in terms of image quality and variety. Additionally, the XSwin model's Inception Score is closest to that of the real dataset, further validating the effectiveness of the model.

Additionally, the training loss curves are shown for these models in Figure 9

Qualitative Results We provide a visual comparison of images generated by the proposed XSwin model and the three baseline models. The images are produced at 150 diffusion steps and are organized into a grid format for each model to facilitate direct comparison.

The generated images are qualitatively assessed based on the clarity of object recognition and the presence of artifacts. Generally, the images exhibit a reasonable level of clarity, with objects being vaguely identifiable in most cases. However, artifacts are present, which can detract from the overall quality.

For a more detailed analysis, we examined specific visual features across different classes generated by each model. The goal was to determine how effectively each model captures the nuances of the classes in terms of shape, texture, and color consistency. Some distinguished classes have noteable variance between the different models:

Figure 10: Grid of images generated from DiT-S/2 model for all Caltech101 classes

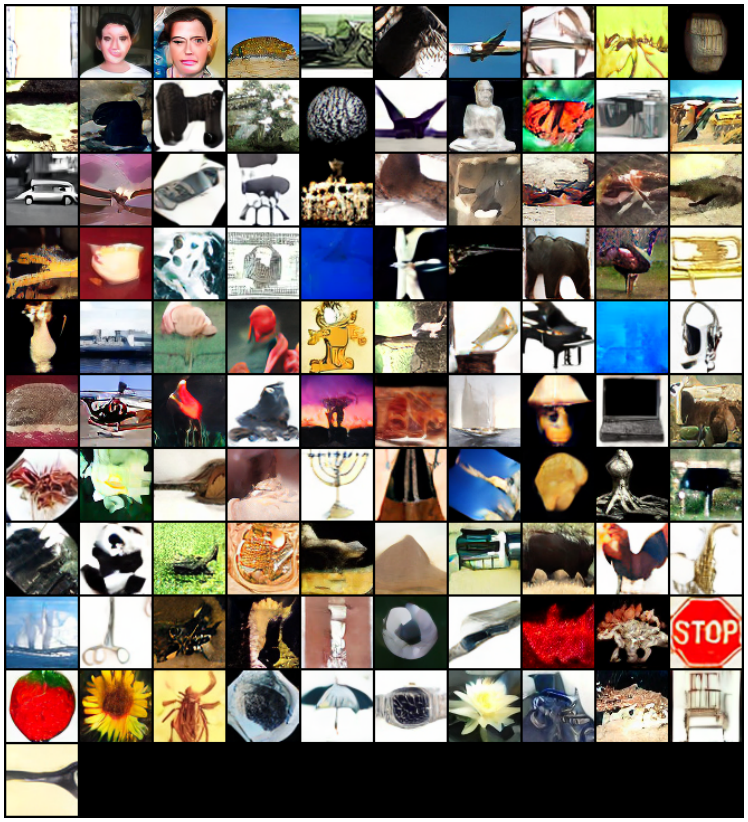

Figure 11: Grid of images generated from UViT-S/2 model for all Caltech101 classes

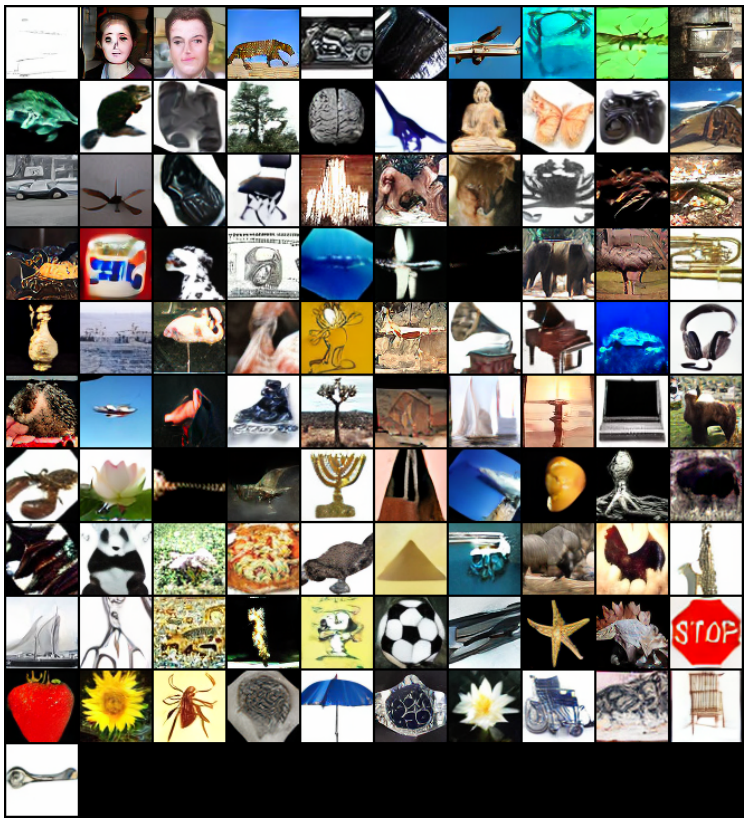

Figure 12: Grid of images generated from USwin model for all Caltech101 classes

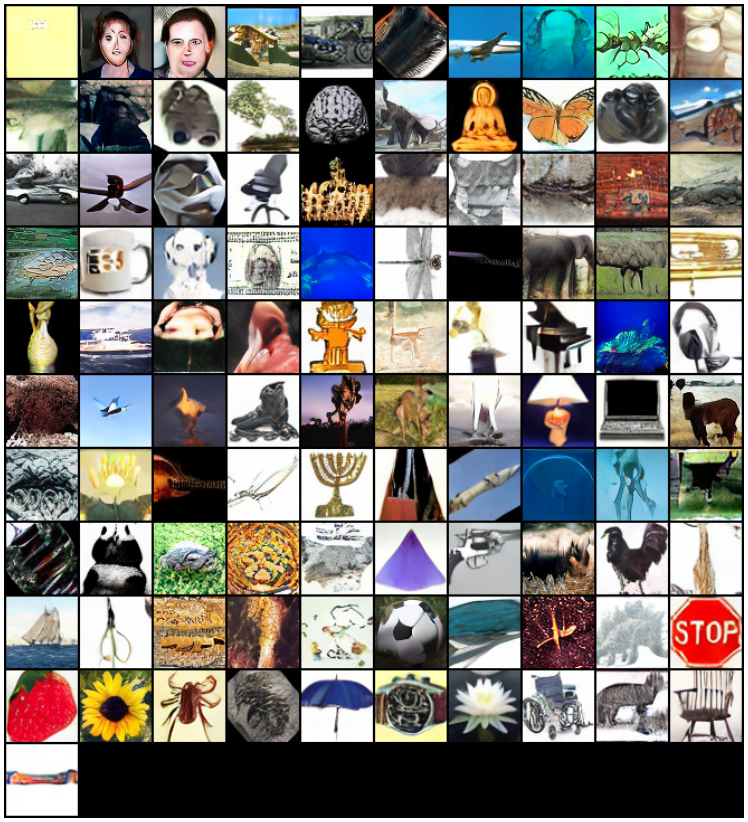

Figure 13: Grid of images generated from XSwin model for all Caltech101 classes

- **Stop Sign (10th from the left, 3rd from the bottom):** We observe that all the models with residual connectivity manage to correctly capture the lexical properties of the stop sign whereas DiT does not.
- **Butterfly (8th from the left, 2nd from the top):** We observe that while the non-hierarchical architectures (DiT, UViT) show more vibrant colors, it fails to capture the major structural details of a butterfly. Our proposed hierarchical architecture (USwin, XSwin) succeed in capturing the global structure of the butterfly while also capturing its high frequency details, and we further observe that XSwin's localized residual cross attention further improves these details.
- **Sunflower (2nd from the left, 2nd from the bottom):** We observe that the non-hierarchical architectures (DiT, UViT) have more blurring and smearing artifacts around the petals. In contrast, USwin suffers this less but incorrectly colors the flower center. Meanwhile, XSwin produces a very high quality sunflower.
- **Wheelchair (8th from the left, 2nd from the bottom):** We observe that the non-hierarchical architectures (DiT, UViT) fail to capture both global structure and high frequency details with high fidelity. In contrast, our proposed architectures (USwin, XSwin) captures both of these qualities significantly better while XSwin still offers better resolution on the high frequency details.
- **Pizza (4th from the left, 4th from the bottom):** We observe that the non-hierarchical architectures (DiT, UViT) either fail to produce fine-grained details (DiT) or fail to capture both global structure (UViT). USwin generates a pizza with moderate fidelity global structure with moderately grained details while XSwin generates a pizza with high fidelity global structure and fine grained details.
- **Instrument (10th from the left, 4th from the top):** We observe that the architectures without residual connectivity (DiT) fail to capture the global structure whereas those with residual connectivity (UViT, USwin, XSwin) succeed on this quality. Meanwhile, UViT does not succeed in capture high frequency details in comparison to our methods (USwin, XSWin).
- **Gun (7th from the left, 4th from the bottom):** While DiT and USWin manage to capture some of the central details of a handgun, only XSwin manages to full capture all the features (including the handle) with high quality.
- **Soccer Ball (5th from the left, 3rd from the bottom):** The images generated across the four models show that the XSwin model produces a soccer ball of notably higher resolution. The USwin model also performs commendably, accurately rendering the ball's roundness and distinctive panel pattern. In contrast, the UViT model's version lacks structure, making the soccer ball almost unrecognizable. The DiT model's soccer ball, while identifiable and with reasonable representation of roundness and pattern, lacks high-resolution quality and misses the context — the absence of a green background suggesting grass detracts from the contextual accuracy.
- **Ship (1st from the left, 3rd from the bottom):** The ship generated by the XSwin model exhibits high resolution and captures detailed contextual elements, such as the ship's presence in water — a detail not depicted by the other models. Although the UViT model attempts to delineate the water surrounding the ship with high resolution, the ship's structure is not well-defined. Consistent with the trend observed, the images from the Swin-based architectures demonstrate higher resolution when compared to those from the DiT and UViT models.

In summary, the qualitative analysis demonstrates that our XSwin model consistently produces images that are visually superior compared to those generated by the baseline models. This superiority is evident in the images' enhanced clarity, reduced artifact presence, and more accurate representation of the nuanced features of each class. The XSwin model's ability to maintain shape, texture, and color consistency across a diverse set of images stands out, reinforcing the quantitative metrics that also suggest its leading performance.

5 Conclusion and Discussion

We set out to learn a distribution over images of the Caltech101 dataset [24] using latent diffusion with an attention based architecture that still preserves the inductive biases of UNet. We achieve this with a Swin based hierarchical architecture that uses outer convolutional layers and a global attention ViT bottleneck to best promote hierarchical feature learning. This design enables multi-scale learning with a UNet design that leverages residual connectivity. We redesign simple skip connections with localized cross atttention to better select and refine local features to combine with global features in the decoder stages. Our overall designs avoids the full quadratic complexity of full context attention while providing feature learning at higher resolutions and with hierarchical, multi-scale structure in addition to more guided and refined residual connectivity.

From our experiments, we found that our proposed model improves on existing attention based architecture like DiT [31] and a UViT [6] design extending DiT. We also perform ablation studies without localized residual cross attention as well. Overall, we evaluated 4 models on Caltech101 with 64x64 images: DiT [31], UViT [6], USwin (our ablation experiment without localized residual cross attention), and XSwin (our full proposed model).

From the training experiments, we first observe the loss curves in Figre 9. We see that UViT actually had the weakest convergence while DiT still had weaker convergence with respect to our proposed architectures. XSwin represents our architecture with localized residual cross attention while USwin respresents our architecture without. In both cases, we see our proposed models fall into a local optima that they divergence and re-converge from; however this is much less pronounced when residual cross attention is used. We see that using the residual cross attention with our architecture very slightly improves losses but offers significantly more stable convergence. We believe that the weaker convergence of UViT despite its better performance on evaluation metrics is likely due DiT overfitting on the training data.

Quantitatively, we can refer to Table 1. We observe that the architectures that take advantage of connecting shallow and deep layers (UViT, USwin, XSwin) perform significantly better than architectures that don't (DiT). This difference is most pronounced with the metrics that are compare generated samples with the original, real dataset (FID, Precision, Recall). From this we conclude that, at the very least, having residual connectivity between shallow and deep layers in a UNet fashion play an important role in ensuring the generated dataset remains perceptually close to the original, real dataset. Meanwhile, we see that the hierarchical architectures (USwin and XSwin) further outperform non-hierarchical architectures (DiT and UViT) on all these metrics. Notably, the DiT scores significantly worse that the other architectures; we original believed this to have been a bug of sorts, but after re-running our DiT experiment, getting similar results, and then designing the UViT architecture as a direct modification of the DiT architecture to then get seemingly more reasonable scores, we come to the conclusion that the lack of residual connectivity by itself plays a very significant role in representation learning in latent diffusion models.

Qualitatively, we can refer to the Figures 10, 11, 12, 13. We observe how our proposed hierarchical architectures perform better at generating high resolution details and semantically capturing the qualities of the respective classes. We also see that localized residual cross attention further improves qualitative performance overall. We discuss a few distinguished examples in the qualitative results section.

Achievements To summarize our achievements, we sought out to improve on state-of-the-art image generation techniques that leveraged the efficiency of latent diffusion and more advanced vision architectures that were attention based. We show that we can, at once, improve on the inefficiency of full context attention and reintroduce hierarchical feature learning with Swin transformer modules supported by high resolution convolutions and low resolution global attention. We also show that using localized residual cross attention to guide and refine skip connections plays a role in improving performance as well through our ablation study with USwin. The significance of our contribution is that we provide a denoising architecture for latent diffusion that can process latent information with higher semantic resolution afforded by deliberate hierarchical design and the computational efficiencies of localized attention contexts. We show that, for small resolution datasets and with limit computational resources, our proposed architecture improves on state-of-the-art transformer based denoising latent diffusion by $\sim 36.46\%$ for FID, $\sim 11.54\%$ for IS, $\sim 12.62\%$ for Precision, and $\sim 3.69\%$ for Recall. As a result, we demonstrate the viability of our architecture and motivate future

works to experiment with our proposed architecture at scale with higher image resolutions and with larger compute environments to train longer.

Lessons Learned From this project, we learned how to effectively decide what kinds of code bases to build off of, the benefits of intermediate validation, what the current state of the art in attention based latent diffusion looks like, how to implement and augment advanced architectures from scratch, how diffusion conditioning can be implemented effectively, how to run diffusion experiments and evaluate results with modern metrics for unsupervised learning, deeper understandings of the interplay of inductive biases in architecture designs (both from our literature review and experiments), and how to effectively divide and conquer our project despite losing a team-mate and having a late start with a reformulated proposal. Much of the learning was also just in exercising our capacities as researchers, coders, and writers.

Given more time, with respect to implementation, we would have liked to have integrated some of our code in Google Colab as function modules in our codebase so that we could do metric evaluations by importing functions directly from our repository rather than using exposed code on Colab notebooks. Additionally, we would have liked to have had the time to implement the exact UViT architecture (moving conditioning to additional tokens rather than modulation), but in consolation the architecture we baselined in its place was much more comparable to both DiT and our architecture in terms of its conditioning mechanisms. In terms of selecting problems, with more time, we would have liked to explored and experimented with different architecture variations and on more datasets with potentially larger resolutions (also given more compute resources) as we feel the impact of our contributions were severely limited by the compute and time we had available; we would have liked to see performance and more competitive benchmarks. In terms of literature review, we think we did a very extensive job but believe that it would have been a nice bonus to explore more on interpret-ability regarding inductive biases. Regarding team formulation, we felt like we would have been more in touch with backup options in case someone dropped from our team; although we were informed very late about it. Given more time, we would have also liked to reformulate our project report into a formal paper submission. This is actually something we want to do after the course and tackle some of the other core points here as well.

For future students, the advice we would give is to identify accessible code bases with clear entry points to plug in your own implementations. You don't have a lot of time and ambitious projects require a lot of dependencies to work together in order to have a working pipeline. We reflect on what decisions made our project successful; one thing to note is that we spent very, very little time debugging the diffusion steps. This is because we had validated our architecture with segmentation before introducing diffusion modulators and conditioning and stuck very closely to the baseline interface so that we could simply plug and play our novel design with a functional pipeline. We were able to do this effectively because the DiT codebase had very accessible entrypoints to run diffusion. In fact, we more or less only needed one 240 line file that was uniquely theirs to be able to implement our pipeline; everything else we used from them was standardized and recycled from previous works; but they had compiled everything very succinctly.

References

[1] Adm guided diffusion code. https://github.com/openai/guided-diffusion/blob/main/guided_diffusion.

[2] Fair positional embeddings code. https://github.com/facebookresearch/mae/blob/main/util/pos_embed.py.

[3] Glide gaussian diffusion code. https://github.com/openai/glide-text2im/blob/main/glide_text2im/gaussian_diffusion.py.

[4] Iddpm gaussian diffusion code. https://github.com/openai/improved-diffusion/blob/main/improved_diffusion/gaussian_diffusion.py.

[5] Association for Artificial Intelligence 2023, Lei Chen, Fei Du, Yuan Hu, Fan Wang, and Zhibin Wang. Swinrdm: Integrate swinrnn with diffusion model towards high-resolution and high-quality weather forecasting, 2023.

[6] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models, 2023.

[7] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation, 2021.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[9] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[11] Chi-Mao Fan, Tsung-Jung Liu, and Kuan-Hsien Liu. SUNet: Swin transformer UNet for image denoising. In *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, may 2022.

[12] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer, 2023.

[13] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[14] Jiaqi Gu, Hyoukjun Kwon, Dilin Wang, Wei Ye, Meng Li, Yu-Hsin Chen, Liangzhen Lai, Vikas Chandra, and David Z. Pan. Multi-scale high-resolution vision transformer for semantic segmentation, 2021.

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018.

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[17] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images, 2023.

[18] Md Amirul Islam, Matthew Kowal, Patrick Esser, Sen Jia, Bjorn Ommer, Konstantinos G. Derpanis, and Neil Bruce. Shape or texture: Understanding discriminative features in cnns, 2021.

[19] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2018.

[20] Kishaan Jeeveswaran, Senthilkumar Kathiresan, Arnav Varma, Omar Magdy, Bahram Zonooz, and Elahe Arani. A comprehensive study of vision transformers on dense prediction tasks, 2022.

[21] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution, 2016.

[22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[23] David Kreuzer and Michael Munz. Transformer-based unet with multi-headed cross-attention skip connections to eliminate artifacts in scanned documents, 2023.

[24] Fei-Fei Li, Marco Andreeto, Marc'Aurelio Ranzato, and Pietro Perona. Caltech 101, 2022.

[25] Ruijun Li, Weihua Li, Yi Yang, Hanyu Wei, Jianhua Jiang, and Quan Bai. Swinv2-imagen: Hierarchical vision transformer diffusion models for text-to-image generation, 2022.

[26] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer v2: Scaling up capacity and resolution, 2022.

[27] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021.

[28] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models, 2022.

[29] Shaoyan Pan, Tonghe Wang, Richard L. J. Qiu, Marian Axente, Chih wei Chang, Junbo Peng, Ashish B Patel, Joseph Shelton, Sagar Patel, Justin R. Roper, and Xiaofeng Yang. 2d medical image synthesis using transformer-based denoising diffusion probabilistic model. *Physics in Medicine and Biology*, 68, 2023.

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[31] William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023.

[32] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023.

[33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[34] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022.

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[37] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022.

[38] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans, 2016.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[40] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. `https://github.com/huggingface/diffusers`, 2022.

[41] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers, 2022.