Final Report - SRIP

Novel View Synthesis Using Diffusion Models

Haarit Chavda (23110077)

0. Introduction

Novel view synthesis involves generating images of a scene from new viewpoints not present in the original dataset. This requires complex 3d reconstruction and specialised generative models. We are developing a diffusion-based approach for novel view synthesis that does not require retraining. Diffusion models are a very powerful tool for generating images and manipulation. Their latent space is highly structured, and we can leverage this structure to perform various tasks without paying the cost of retraining the model, thereby saving compute and time. We use this pretrained diffusion model to synthesize novel views without training a separate network. We aim to explore the latent space of this model and apply interpolation techniques to generate plausible novel views.

1. Literature Review and Ideation

1. **Exploring the latent space of diffusion models directly through singular value decomposition(*link*):**
   This paper investigates the latent space of diffusion models by performing SVD on the latent codes of the image. They found three core observations of the latent space.
   1) Small neighborhood, the subspaces constructed by left and right singular vectors (both are orthogonal vectors in descending order based on singular values) remain semantically similar across all time steps, which indicates arbitrary attributes destined by text prompts can be introduced into this small neighborhood.
   2) Attributes encoded in these singular vectors in the form of vector values and their entangled singular values, and the residential attributes can not be changed if no new singular vectors (presenting new attributes) are added.
   3) Mobility in order, assuming singular vectors are always ordered along with descending of their singular values at all time steps, singular vectors accounting for attributes have mobility in order across time steps.

2. **Addressing degeneracies in latent interpolation for diffusion models(*link*):**
   This paper addresses the degeneracy problem of interpolating between two inverted latent spaces. The observations are:
   1) Inverted Latents have a slight deterministic bias. When averaging multiple latent vectors, this bias is amplified by √N.

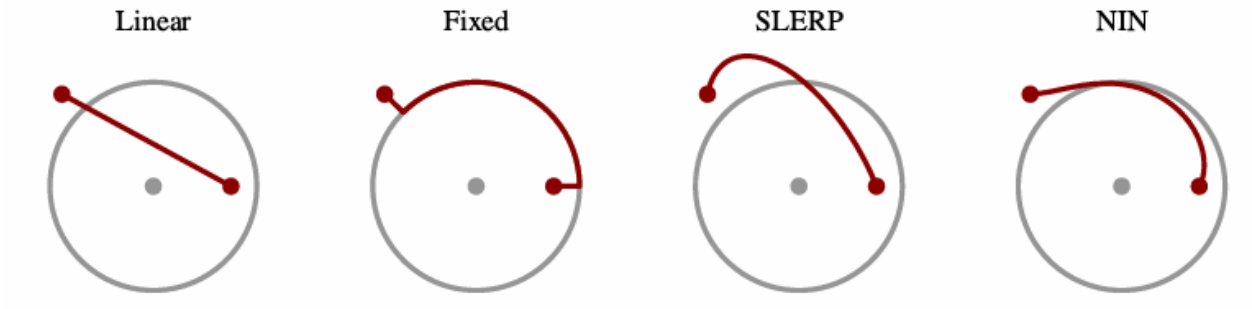
2) Unlike randomly sampled latents from N(0, I), inverted latents do not follow an ideal Gaussian distribution. Therefore, normalization does not work.

They propose a method to overcome these challenges. Decompose each latent into: $z_n = d_n + e_n$. d_n It is the channel-wise mean and $e_n = z_n - d_n$ the noise component. Interpolate the DN linearly and the noise part using NIN.

3. **Synthesizing Consistent Novel Views via 3D Epipolar Attention without Re-Training(*link*):**
   The objective of this paper is zero-shot novel view synthesis, given a source image without retraining. They provide a clever method of integrating the epipolar geometry for image generation by editing the attention maps. This is described in detail in later sections.

2. Interpolation Methods

Below is a visualization comparing interpolation trajectories of all methods. The gray circle represents the hypersphere of radius √d (where d is the latent dimension), denoting the region of maximum probability density as per the annulus theorem in high-dimensional geometry.



For generating novel views between images from two angles, we tried the following interpolation techniques:

1) **LERP(Linear Interpolation):**
   It is finding a value between two latent vectors on a straight line between them.
   Formula: $P(t) = (1 - t).A + t.B$

Transition from initial (top left) to final (bottom right) image using interpolation

2) Fixed Norm Interpolation:

When we do linear interpolation in higher dimensions, the resulting vector becomes short, i.e,.It will have less norm than the other two vectors. Diffusion models and other generative models have a latent space of higher dimension. Therefore, they expect the latents to lie on a hypersphere of radius √L, where L is the dimension of the latent space. Consequently, we will weight each interpolated vector by √L.

Formula:

$$zlerp = (1 - t)z_0 + tz_1$$

$$z' = \frac{\sqrt{L}}{||Zlerp||} . zlerp$$

Transition from initial (top left) to final (bottom right) image using interpolation

3) SLERP(Spherical Linear Interpolatoin):

Instead of drawing a straight line like LERP, it interpolates along the arc of the hypersphere between two vectors. It gives better interpolation as the generated vector will lie on the hypersphere of the desired radius.

Formula:

1. Normalize: $z_0' = \frac{z_0}{||z_0||} \quad z_1' = \frac{z_1}{||z_1||}$
2. Angle Between them: $w = arccos(\frac{z_0 \cdot z_1}{||z_0|| . ||z_1||})$
3. Interpolated Point: $slerp(t) = \frac{sin((1-t)w)}{sin(w)} . z_0 + \frac{sin(tw)}{sin(w)} . z_1$

4) NIN (Norm Interpolated Norm):

It is also a technique for interpolating between two vectors that has a unique property of maintaining a smooth norm between the vectors. The norm of the interpolated vectors matches the weighted average of input norms.

Formula:

1. Interpolated Vector: $z_1 = w_0 z_0 + w_1 z_1$
2. Interpolated Norm: $r = w_0 ||z_0|| + w_1 ||z_1||$
3. Rescale: $z' = r . \frac{z_1}{||z_1||}$

5) Channel Mean Interpolation

This is the method suggested in the paper. When we invert an image(for diffusion models), the latent produces some bias. To fix this, the paper proposes the following method

1. Decompose each latent into: $z_n = d_n + e_n$

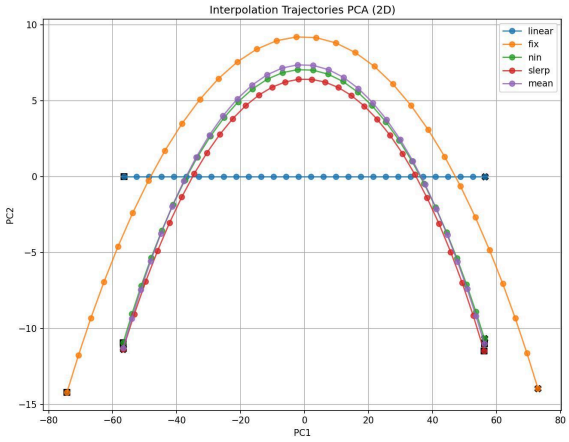
   d_n It is the channel-wise mean, and $e_n = z_n - d_n$ it is the noise component.
2. Interpolate separately: We interpolate dn linearly and the noise part using NIN

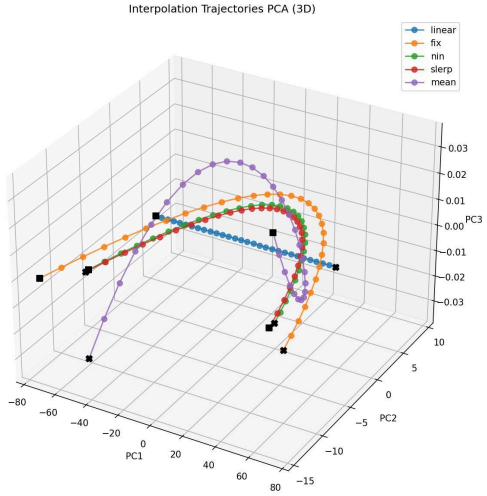
Results and observations

In linear interpolation, the intermediate images tend to be quite blurry and faded. Methods like SLERP, NiN, and channel-mean produce similar results, also resulting in blurry images. The fixed norm method generates images with much higher contrast, which is a result of normalization, but the images remain blurry.

We looked at the results for interpolation steps at 0 (starting image), 4, 8, 12, 16, 20, 24, and 30 steps using the abovementioned methods. Then, we used PCA (Principal Component Analysis) to look at how these interpolated vectors relate to the original ones, and we plotted them in both 2D and 3D to see whether they follow a particular trajectory.

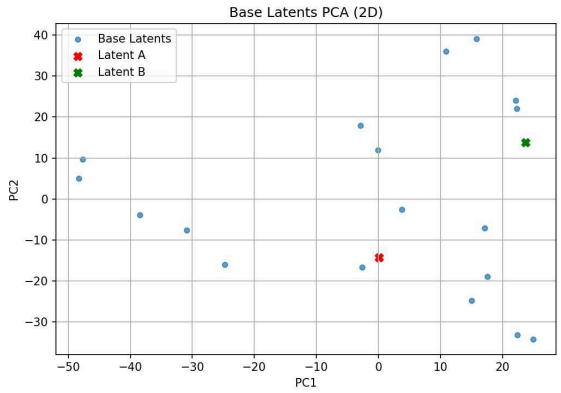
Interpolation Trajectories in 2D and 3D for LERP, SLERP, NI, ,N and Channel Mean methods

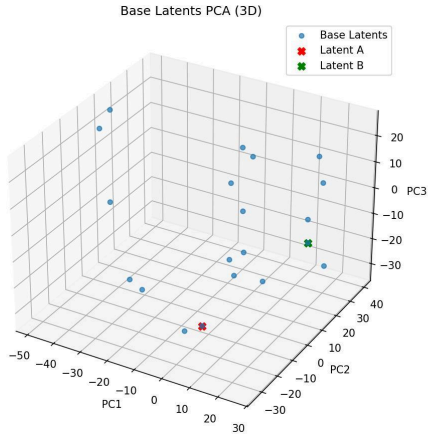




PCA RESULTS - BASE LATENTS (2D and 3D)

Latent A is the initiator, and Latent B is the final.



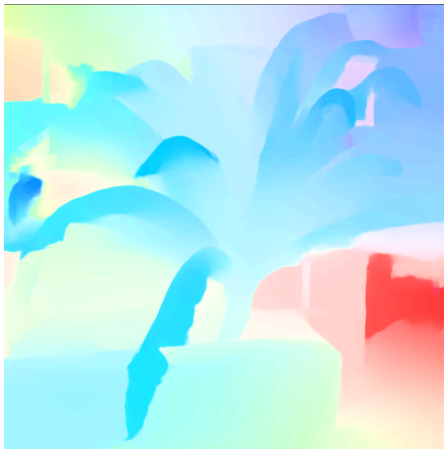


3. Blending Optical-Flow with Image

Given the initial and final images, we first compute the optical flow between them. We then treat the RGB representation of this optical flow as an image and invert it through the diffusion model to obtain its latent representation. This latent is added to the original image's latent, and the diffusion model decodes the combined representation to observe the resulting interpretation. The resulting image is a blend of flow and base image.

Image 1

Image 2



Optical Flow

Final Output

4. Applying gradient-based interpolation

In earlier experiments, we observed that direct interpolation in the complex latent space of diffusion models often leads to blurry transitions between views. This blurring effect is likely due to traversing regions of low probability density in the latent space.

To address this, we adopt a gradient descent (GD) approach to smoothly transition from the latent space of the initial image to that of the final image. Unlike fixed-path interpolation, this method optimizes the transition path while adhering to constraints that ensure fidelity to the diffusion model's latent space.

During gradient descent, we use the following optimization objectives:
Optimization Objectives for GD

1) **Noise Prediction** : $||e_{\theta}(z_c, t) - \varepsilon||^2$

   Enforces the latent to lie on the probabilistic plane of the diffusion model

2) **Target Proximity:** $||z_c - z_f||^2$

   Ensures that with each step of GD, the latent moves towards the target latent.

3) **Norm Constraint**: $| \ ||z_c||_2 - \sqrt{L} \ |^2$

   Ensures that the norm of the latent vector is close to $\sqrt{L}$ and lies on the hypersphere where the probability density is highest. (L is the dimension of the latent space)

Gradient descent did not yield the expected results. The blurring effect is supposed to vanish by this method. The images generated are also some sort of blend between the initial and the final images. This is because the latent space of the diffusion model is a complex combination of text and images. Below are the results for steps(0,100,200,300,500) for gradient-based optimization.

Initial Final

Integrating CLIP Loss

By integrating CLIP loss into this optimization, we aim to ensure that each intermediate image generated during GD is not only structurally plausible but also semantically aligned with the target image. Specifically, the CLIP loss measures and enforces the similarity between the generated image and the target (e.g., visual stimulus), helping to reduce semantic errors and improve consistency.

CLIP loss : $L_{clip}(\chi, G(\beta)) = -\varepsilon_{clip}(\chi) \cdot \varepsilon_{clip}(\chi_{\beta}(t))$

Despite integrating the diffusion CLIP loss to enhance semantic and viewpoint consistency, the resulting images remain blurry and do not show significant improvement. The generated outputs largely appear as blends of the initial and final images, rather than achieving clear, distinct transitions

Initial 50(gd steps) 150(gd steps) final

5. Image shifting

In this approach, we shift the image by a few pixels. To handle the resulting blank regions, we experiment with two different strategies:

1. Leaving the blank pixels as zeros
2. Filling them using the average value across rows or columns

The goal is to observe whether the diffusion model can complete the missing (blank) regions during inversion, effectively generating novel views corresponding to a shifted viewpoint.

Method 1

The model is complex enough to reconstruct the same image even after removing a significant portion. However, the texture of the fern and some subtle features in the reconstructed image differ from those in the initial image.

Initial image (uncompressed) - Final image

Method 2

This experiment's results are similar to those of the previous method. The reconstructed image is identical to the initial image.

Initial image (uncompressed) - Final image

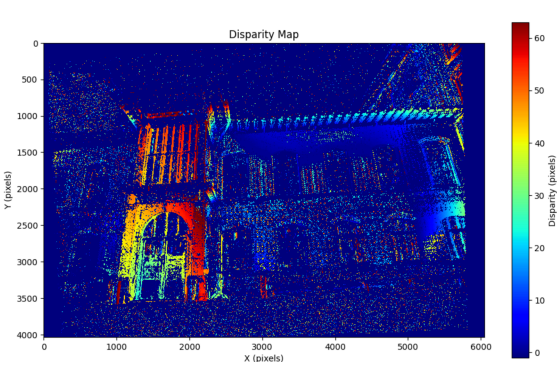
6. Implementation of 3d reconstruction using two images

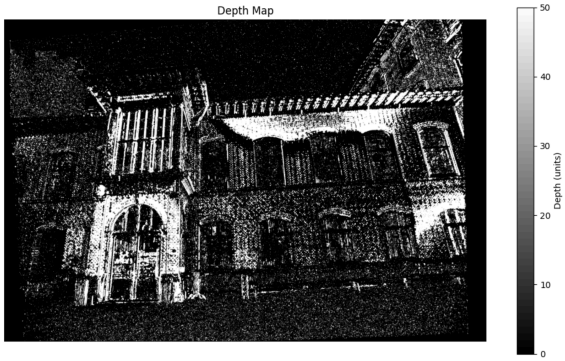
To better understand epipolar geometry, we implemented a pipeline that computes disparity and depth maps from a stereo image pair using the known intrinsic parameters of both cameras. This step-by-step reconstruction provides a practical understanding of how stereo vision and camera calibration generate 3D representations from 2D inputs.

1) Creating K(intrinsic matrix from the camera parameters)
2) Keypoint detection: Detect keypoints in both images using SIFT. Now use K-nearest neighbours to filter the keypoints. Filter outliers using Lowe's ratio test or mutual consistency checks.
3) Fundamental Matrix Estimation: OpenCV provides a function that calculates the fundamental matrix based on keypoints and intrinsic matrices from both images.
4) Stereo Rectification: Compute the rectification homographies for both images. By doing this, the epipolar lines are aligned horizontally, making it easier to compare the features.
5) Computing the disparity map: After rectification, compute the disparity map by finding the horizontal shift between pixels.
6) Depth Map Calculation: Convert disparity values to depth map using the formula $Depth(x, y) = \frac{f.B}{Disparity(x,y)}$ f, which is the focal length, and B, which is the baseline distance between the cameras.

Rectified Image 1

Rectified Image 2





7. Novel Views via 3D Epipolar Attention without Re-Training

Here we dig deeper into the paper, epipolar attention. The paper introduces a training-free framework that leverages epipolar geometry to boost the consistency in novel view synthesis. The core improvement is the epipolar attention module, which:

- Locates and retrieves features from the reference view along the epipolar line corresponding to each pixel in the target view.
- Aggregate these features to constrain the generation of the target view, ensuring that overlapping regions remain consistent in geometry and appearance.
- Requires no retraining or fine-tuning of the base diffusion model, making it plug-and-play for any pre-trained pose-conditional diffusion model

Implementation

1) **Epipolar Line Computation:**
   For each pixel in the target view, its corresponding point in the reference view must lie on a specific epipolar line, determined by the known relative camera pose. The epipolar line

is computed using the fundamental or essential matrix derived from the camera intrinsics and relative pose (R,t) as $l_i = R[t]_x K^{-1} p_i$, where p_i is the pixel in the target view, K is the intrinsic matrix. $[t]_x$ It is the skew-symmetric matrix of the translation vector t.

2) Feature Matching Along Epipolar Line

- For each target pixel, a set of points is sampled along its epipolar line in the reference view.
- The features at these sampled points are retrieved from the reference view using bilinear interpolation (to handle sub-pixel locations).
- The similarity between the target pixel's feature (query) and the sampled reference features (keys) is computed using the dot product (as in standard attention), followed by a softmax to obtain attention weights.
- The similarity between the target pixel's feature (query) and the sampled reference features (keys) is computed using the dot product (as in standard attention), followed by a softmax to obtain attention weights.

3) Reference Feature Injection and Parameter Duplication

The aggregated reference features are injected into the target view's generation process. The final output feature at each attention block is a blend of the standard self-attention output and the epipolar attention output, controlled by a fusion weight α.

$$F' = \alpha F'_{src} + (1 - \alpha)F$$

F'_{src} is the epipolar attention output, and F is the standard self-attention output

An overview of the method proposed in the paper

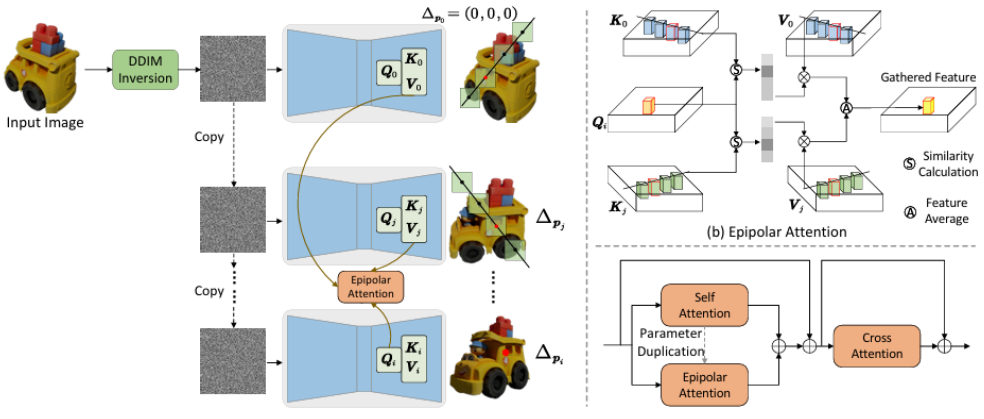


Figure 3. **Overview of our method.** (a) We first perform DDIM inversion on the input image to obtain the initial noise, which is shared during the multi-view image generation process. Throughout the generation of each view, our epipolar attention block efficiently locates and retrieves corresponding information from both the input image and other target views. (b) The architecture of our 3D epipolar attention module. (c) Location of our inserted epipolar attention block.

Results

The outcomes of our experiments did not fully align with our expectations. While the generated views exhibit variations in texture and some fine details, they do not represent true novel viewpoints of the scene. Instead, the changes are limited to subtle feature modifications rather than significant perspective shifts.

A key challenge we encountered was the lack of an available codebase for the referenced paper, which was only published in February 2025. As a result, we had to implement the epipolar attention block from scratch, which may have introduced additional complexity and potential discrepancies from the original method.

Another important factor contributing to the limited success of our approach is the difference in base models. The original paper utilizes the Zero123 model, which inherently encodes spatial information crucial for novel view synthesis. In contrast, our experiments were conducted using the Stable Diffusion model, which does not possess this spatial encoding, making it less suited for generating accurate novel views.

These are the 4 generated views by our approach. The texture of the fern and some finer features change, but it does not give us a novel view. ($\alpha = 0.7$)

View 1 View 2 View 3 View 4

For $\alpha = 0.9$ only epipolar attention

The generated images are noticeably blurry. This is expected, as relying solely on epipolar attention restricts the model to features along the epipolar line, limiting the richness and clarity of the output.

Blurry images

For $\alpha = 0.1$ more weight on self-attention

The images are sharper and clearer compared to using only the epipolar attention block. Increasing the influence of self-attention helps preserve more of the original image structure, though it still does not fully achieve the desired novel view synthesis.

Sharper Images

8. References

1) Synthesizing Consistent Novel Views via 3D Epipolar Attention without Re-Training. (2025). In arXiv [Journal-article]. https://arxiv.org/abs/2502.18219v1

2) Addressing degeneracies in latent interpolation for diffusion models. (2025). In arXiv: Vol. 2505.07481v1 [Journal-article]. https://arxiv.org/abs/2505.07481v1

3) Wang, L., Zhejiang University, Gao, B., University of Oxford, Li, Y., University of Bedfordshire, Zhejiang University, Bournemouth University, Clifton, D. A., University of Oxford, Xiao, J., & Zhejiang University. (n.d.). Exploring the latent space of diffusion models directly through singular value decomposition. https://arxiv.org/abs/2502.02225